



APOYO DE TOMA DE DECISIÓN A
COMUNIDADES DE CONSTRUCCIÓN DE
CONOCIMIENTO

Jonathan Martin

Director: Dr. Diego Torres

Codirector: Dr. Alejandro Fernández

Presentado en cumplimiento de los requisitos para la obtención del
grado de

Doctor en Ciencias Informáticas

FACULTAD DE INFORMÁTICA

Universidad Nacional de La Plata

Noviembre, 2020

Agradecimientos

Quiero agradecerle a mi familia. A mis padres por su infinita paciencia, confianza plena y apoyo incondicional. A mi madre por escuchar mis crípticas explicaciones sobre los problemas técnicos de cada enfoque evaluados en este trabajo. A mi padre por cada mañana camino a la facultad sea de día, de noche, llueva o truene. A mi hermana y mi sobrina por su compañía a pesar de las distancias. Y a mi abuelo que ya no está, pero siempre me recordó cuanto orgullo sentía por mí.

También quiero agradecerles a mis amigos. A Ramiro, que fue el quien me acerco al LIFIA por primera vez y con quien compartí, comparto y espero continuar compartiendo muchas materia, trabajo, proyectos y charlas de todo juntos. A Lucas, por todos esos mates y charlas matutinas que demuestran que una diferencia de oficinas no es impedimento para nada. Y a todos los que si bien no veía o no hablábamos diariamente siempre estaban ahí para charlar o despejarnos juntos ya sea de la carrera o de la vida.

Al financiamiento otorgado por la Universidad Nacional de La Plata para la realización de este trabajo, el cual me permitió la dedicación plena a mis estudios e investigación.

Finalmente, y ni por cerca menos importante, quiero agradecerle a todo el LIFIA. A Diego y a Alejandro, que sin su dirección y apoyo este trabajo no sería posible. A la gente de becarios que alegraban la oficina día a día. Especialmente a Natalia por su buena onda todas las mañanas, y su paciencia con toda la charla que le daba. Podría escribir las siguientes páginas contando y agradeciendo a cada uno de los miembros de este maravilloso laboratorio, contando anécdotas y lecciones importantes que me dejaron. Pero no dejaría lugar para este trabajo de tesis, así que solo les digo gracias por su apoyo y calidez.

Muchas Gracias.

Resumen

La construcción del conocimiento puede verse como resultado del proceso activo del aprendizaje. Siguiendo este concepto en la actualidad podemos encontrar varias comunidades que se forman a partir de grupos de personas con intereses comunes para compartir y generar conocimiento. De estas comunidades, particularmente a las que brindan soporte computacional a las distintas etapas de la construcción de conocimiento las denominaremos Comunidades de Construcción de Conocimiento (CCC).

La calidad del contenido en las CCC depende de la actividad y participación de sus miembros. Existen diferentes enfoques para alentar la participación de una CCC, por ejemplo, centrarse en atraer a los recién llegados o brindarles más atención u oportunidades de participación que a los miembros más antiguos de la comunidad. Un enfoque de tendencia es el uso de la ludificación.

La ludificación es la aplicación de cualquier elemento de juegos o videojuegos en entornos no lúdicos. A través de estas estrategias, es factible mejorar la motivación para participar en diferentes escenarios de una CCC, así como guiar el tipo de participación que se ajuste mejor a los principios de la CCC.

Una CCC podría enriquecerse mediante una ludificación que recompense la edición, la participación en discusiones con otros miembros, y diseminación de las producciones. Sin embargo, definir y probar un diseño ludificado en una CCC es un proceso largo que requiere iteraciones de configuración y evaluación que pueden resultar perjudiciales para la comunidad.

Esto se debe a que el resultado de la aplicación de la ludificación no se puede predecir por completo. Una ludificación puede ser disruptiva para la comunidad si está mal configurada, por ejemplo, induciendo una competencia improductiva o desmotivando la participación.

Esta tesis apoya la toma de decisión recomendando configuraciones, a fin de guiar al encargado de ludificar una comunidad. Tomando en consideración el contexto de incertidumbre inherente a la forma de trabajo colaborativo en las CCC.

Para ello se desarrolla una metodología, que mediante el uso de aprendizaje por refuerzo y simulación basada en datos históricos configura ludificaciones para CCCs. Esta metodología a su vez, se presenta aplicada en el framework de RL desarrollado en este trabajo de tesis, llamado Historical Reinforcement Learning.

Historical RL brinda soporte para la generación de entornos de aprendizaje por refuerzo que permitan entrenar agentes de RL en la tarea de configurar una ludificación para una determinada CCC. Dentro de sus características principales esta la compatibilidad con diversas librerías de agentes mediante el estándar de OpenAI Gym. También facilita la integración de diversas comunidades y gestiona el flujo de datos históricos con los que se alimenta a la simulación. Finalmente introduce herramientas para el manejo de acciones por parte de los agentes y la entrega de recompensas a los mismos.

En este trabajo se presentan también diversos casos de estudio donde se evalúa el alcance de la metodología, así como las capacidades del framework. Estos estudios demuestran una mejora respecto a la capacidad de configuración de la ludificación en un entorno no disruptivo para la comunidad.

Índice general

1. Introducción	11
1.1. Problemática de la configuración de una estrategia de ludificación	12
1.2. Hipótesis y enfoque de solución	13
1.3. Resultados obtenidos	14
1.4. Publicaciones	15
1.5. Organización de la tesis	16
2. Marco conceptual	19
2.1. Sistemas de soporte a la toma de decisiones	19
2.2. Comunidades de construcción de conocimiento	22
2.2.1. Soporte por computadoras	23
2.2.2. Wikipedia y Stack overflow	24
2.3. Ludificación	32
2.3.1. MDA Framework	33
2.3.2. Ludificación en CCC	37
2.4. Reinforcement Learning	38
2.4.1. Algoritmos de RL	38
2.4.2. Entornos de RL	41
2.4.3. RL para configurar una ludificación	42
3. Análisis de Patrones en CCC	46
3.1. Procesamiento del historial de revisiones	46
3.1.1. Extracción de Wikipedia	47
3.1.2. Detección de actividades	49
3.2. Abstracción de eventos en Wikipedia	51
3.3. Resultados	52
3.3.1. Herramientas	53
3.3.2. Métricas	53
3.3.3. 1er caso: Análisis de Grupos de Revisiones	54
3.3.4. 2do caso: Análisis de Evolución de Revisiones	55
3.3.5. 3er caso: Patrones de adición y eliminación	56
3.3.6. 4to caso: Granularidad fina de cambios	58

3.3.7. 5to caso: Evolución de revisiones detalladas por grupos	60
3.4. Resumen del capítulo	62
4. Incorporación de ludificación en CCC	63
4.1. Enfoque de MDA en Wikipedia	63
4.2. Simulación de Metagame	65
4.2.1. La comunidad de Agricultura en Wikipedia	65
4.2.2. Definición de la problemática	66
4.2.3. Metodología	68
4.2.4. Evaluación	71
4.2.5. Resultados	72
4.3. Resumen del capítulo	77
5. Enfoque de solución	79
5.1. Simulación de la ludificación	79
5.2. Aprendizaje por refuerzo como enfoque	80
5.3. Resumen del capítulo	81
6. Framework de reinforcement learning para automatización de configuración en CCC	82
6.1. Presentación tecnologías	82
6.1.1. OpenAI Gym	83
6.1.2. Deep learning	83
6.2. Historical Framework	85
6.2.1. Historical Enviroment	87
6.2.2. Manejo de acciones	88
6.2.3. Abstracción del simulador	89
6.2.4. Procesamiento de datos históricos	90
6.3. Resumen del capítulo	91
7. Casos de estudio	93
7.1. Wikipedia	94
7.1.1. Integración del framework	94
7.1.2. Materiales y métodos	97
7.1.3. Resultados	103
7.1.4. Análisis de resultados	110
7.1.5. Conclusiones	114
7.2. Stack Overflow	115
7.2.1. Ludificación de Stack Overflow	115
7.2.2. Metodología de evaluación	118
7.2.3. Resultados	120
7.2.4. Análisis de resultados	130
7.2.5. Conclusiones	132
7.3. Integración de nuevos agentes	132

7.3.1. Selección de agentes de RL	133
7.3.2. Metodología de evaluación	134
7.3.3. Resultados	136
7.3.4. Análisis de resultados	141
7.3.5. Conclusiones	142
7.4. Resumen del capítulo	143
8. Conclusiones y trabajo futuro	144
8.1. Resumen de contribuciones	144
8.1.1. Un estudio de factibilidad del uso de datos históricos .	144
8.1.2. Una estrategia para la extracción y conversión de da- tos históricos	145
8.1.3. Una estrategia de simulación para estrategias de ludi- ficación mediante datos históricos	145
8.1.4. Un enfoque para la obtención de configuraciones ópti- mas mediante aprendizaje por refuerzo	146
8.1.5. Casos de uso del enfoque	146
8.2. Limitaciones	147
8.3. Trabajo futuro	148
Acrónimos	150
Glosario	151

Índice de figuras

2.1.	Artículo de Stack Overflow en Wikipedia en su versión inglesa.	25
2.2.	Edición artículo de Stack Overflow en Wikipedia en su versión inglesa.	26
2.3.	Artículo de Stack Overflow en Wikipedia en su versión inglesa, versión del 26 de Mayo del 2020.	27
2.4.	Historial de revisiones del artículo de Stack Overflow en Wikipedia en su versión inglesa.	28
2.6.	Página de discusión del artículo de Stack Overflow de Wikipedia en su versión inglesa.	29
2.5.	Pregunta realizada en Stack Overflow con su respuesta más votada.	30
2.7.	Sección de mensajería privada de Stack Overflow.	31
2.8.	Proceso de Reinforcement learning.	39
3.1.	Procesamiento del historial de revisiones.	47
3.2.	Extracto del artículo del Papa en texto plano.	49
3.3.	Extracto del artículo del Papa en Wikipedia.	50
3.4.	K-means clusters.	55
3.5.	Rangos de valores en Eclat y Apriori para cada cluster.	60
3.6.	Rangos de valores en Eclat y Apriori para cada cluster.	60
4.1.	Wikigamification con Metagame.	69
4.2.	Tipo de jugadores por nivel de actividad.	73
4.3.	Número de actividades por tipo de jugador.	73
4.4.	Distribución de eventos por proyecto.	74
4.5.	Las acciones de Metagame en el tiempo.	75
4.6.	Jugadores de Metagame por proyecto y por rango.	76
4.7.	Metagame. Porcentaje de jugadores por rangos.	77
4.8.	Entrega de insignias en el tiempo.	78
6.1.	Historical Framework	86
6.2.	Relación entre las clases Env y Historical Enviroment	87
6.3.	Jerarquía para el manejo de acciones	88

6.4. Elementos del simulador.	90
6.5. Manejadores de datos	91
7.1. Artículo destacado con distintivo de una estrella de bronce. . .	100
7.2. Distribución del tiempo de vida de los artículos destacados. . .	101
7.3. Tiempo como artículos destacados.	102
7.4. Artículos destacados, tiempo de vida y tiempo como destacados.	102
7.5. Cantidad de acciones requeridas para obtener un nuevo nivel de insignia en Wikicup.	105
7.6. Escala de dificultad para WikiCup.	105
7.7. Insignias entregadas a lo largo del tiempo para WikiCup. . .	106
7.8. Cantidad de acciones requeridas para obtener un nuevo nivel de insignia en el grupo de artículos jóvenes.	109
7.9. Cantidad de acciones requeridas para obtener un nuevo nivel de insignia en el grupo de artículos antiguos.	110
7.10. Escalado de dificultad en el grupo de artículos jóvenes.	111
7.11. Escalado de dificultad en el grupo de artículos antiguos. . . .	112
7.12. Insignias entregadas en el tiempo para los artículos jóvenes. .	113
7.13. Insignias entregadas en el tiempo para los artículos antiguos.	113
7.14. Diagrama entidad relación simplificado de Stack Overflow. . .	117
7.15. Primer semestre. Nuevas acciones necesarias para obtener ca- da insignia.	121
7.16. Primer semestre. Total de acciones necesarias para obtener cada insignia.	121
7.17. Primer semestre. Entrega de insignias a los usuarios a lo largo del tiempo.	121
7.18. Segundo semestre - Configuración del primer semestre. En- trega de insignias a los usuarios a lo largo del tiempo.	123
7.19. Segundo semestre - Configuración Nueva. Nuevas acciones ne- cesarias para obtener cada insignia.	126
7.20. Segundo semestre - Configuración Nueva. Total de acciones necesarias para obtener cada insignia.	126
7.21. Segundo semestre - Configuración Nueva. Entrega de insignias a los usuarios a lo largo del tiempo.	126
7.22. Segundo semestre - Configuración por reentrenar 1 episodio. Nuevas acciones necesarias para obtener cada insignia.	128
7.23. Segundo semestre - Configuración por reentrenar 10 episodios. Nuevas acciones necesarias para obtener cada insignia.	128
7.24. Segundo semestre - Configuración por reentrenar 20 episodios. Nuevas acciones necesarias para obtener cada insignia.	128
7.25. Segundo semestre - Configuración por reentrenar 1 episodio. Total de acciones necesarias para obtener cada insignia.	129
7.26. Segundo semestre - Configuración por reentrenar 10 episodios. Total de acciones necesarias para obtener cada insignia.	129

7.27. Segundo semestre - Configuración por reentrenar 20 episodios. Total de acciones necesarias para obtener cada insignia.	129
7.28. Segundo semestre - Configuración por reentrenar 1 episodio. Entrega de insignias a los usuarios a lo largo del tiempo.	130
7.29. Segundo semestre - Configuración por reentrenar 10 episodios. Entrega de insignias a los usuarios a lo largo del tiempo.	130
7.30. Segundo semestre - Configuración por reentrenar 20 episodios. Entrega de insignias a los usuarios a lo largo del tiempo.	130
7.31. Resultados del entrenamiento agrupado por agentes.	137
7.32. Resultados del entrenamiento agrupado por tipo de estado y tipo de recompensa.	137
7.33. Resultados de la evaluación en Wikicup2018 agrupado por agentes.	139
7.34. Resultados de la evaluación en Wikicup2018 agrupado por tipo de estado y tipo de recompensa.	139
7.35. Resultados de la evaluación en Wikicup2019 agrupado por agentes.	140
7.36. Resultados de la evaluación en Wikicup2019 agrupado por tipo de estado y tipo de recompensa.	140

Índice de tablas

2.1. Niveles de experiencia en Metagame (Rango del jugador). . .	36
3.1. Wiki-markups analizados.	50
3.2. Clusters recomendados.	54
3.3. Caso 3. Resultados de Eclat.	57
3.4. Caso 3. Resultados de Apriori.	57
3.5. Caso 4. Resultado de Eclat.	58
3.6. Caso 4. Resultado de Apriori.	59
3.7. Caso 5. Resultados de Apriori para Ediciones Comunes	61
3.8. Caso 5. Resultados de Eclat para Ediciones Comunes.	61
4.1. Agriculture by type en Wikipedia.	66
4.2. Eventos de Wikipedia en el contexto de Metagame.	66
4.3. Puntos para ganar insignias en Metagame.	68
7.1. Configuración de variables de la función input(x) para Wiki- Cup.	104
7.2. Porcentaje de jugadores por rango en Wikicup.	106
7.3. Rango y puntaje de jugadores en WikiCup.	108
7.4. Configuración de variables de la función input(x) para el gru- po de artículos destacados jóvenes.	108
7.5. Configuración de variables de la función input(x) para el gru- po de artículos destacados antiguos.	108
7.6. Rango de jugadores en artículos destacados.	111
7.7. Comparativa de Rangos.	114
7.8. Eventos de Stack Overflow en el contexto de la ludificación. .	118
7.9. Configuración para comunidad de python, primer semestre del 2018.	120
7.10. Rango de jugadores en comunidad de python, primer semestre del 2018.	122
7.11. Rango de jugadores en comunidad de python, segundo semes- tre del 2018 con la configuración del primer semestre.	124

7.12. Nueva configuración para comunidad de python, por el agente pre-entrenado, segundo semestre del 2018.	125
7.13. Rango de jugadores en comunidad de python, segundo semestre del 2018 con la configuración nueva del agente pre-entrenado.	125
7.14. Nueva configuración para comunidad de python, por el agente reentrenado en 1 episodio, segundo semestre del 2018.	127
7.15. Nueva configuración para comunidad de python, por el agente reentrenado en 10 episodios, segundo semestre del 2018.	127
7.16. Nueva configuración para comunidad de python, por el agente reentrenado en 20 episodios, segundo semestre del 2018.	127
7.17. Rango de jugadores en comunidad de python, segundo semestre del 2018. Por periodo de reentrenamiento.	131

Capítulo 1

Introducción

La construcción del conocimiento puede verse como resultado del proceso activo del aprendizaje[1]. Siguiendo este concepto en la actualidad podemos encontrar varias comunidades con soporte computacional que se forman a partir de grupos de personas con intereses comunes para compartir y generar conocimiento, como pueden ser foros especializados, o grupos como los brindados por servicios como Groups Yahoo[2], o comunidades de la familia MediaWiki como Wikipedia[3]. A estas comunidades las denominaremos Comunidades de Construcción de Conocimiento (CCC), que son, según Stahl [4], las que dan soporte computacional a las distintas etapas de la construcción de conocimiento.

La calidad del contenido en las CCC depende en gran medida de la actividad y participación de los miembros que la componen[5]. Existen diferentes enfoques para alentar la participación de una CCC, por ejemplo, centrarse en atraer a los recién llegados o brindarles más atención o oportunidades de participación que a los miembros más antiguos de la comunidad. Un enfoque en auge para, entre otros objetivos, la motivación de la participación es el uso de la ludificación[6, 7].

La ludificación es la aplicación de cualquier elemento de juegos o videojuegos en entornos no lúdicos evitando a su vez modificar su estructura central [8, 9, 10]. A través de la misma, es factible mejorar la motivación para participar en diferentes escenarios de una CCC, así como guiar el tipo de participación que se ajuste mejor a los principios de la CCC: colaboración, generación de calidad y participación. La ludificación es una de las estrategias que se explora para construir y mantener una comunidad productiva y activa de editores [11, 8]. Una CCC podría enriquecerse mediante un juego que recompense la edición, la participación en discusiones con otros miembros, y la diseminación de las producciones.

Otro aspecto clave para este trabajo respecto a las estrategias de ludificación, es la posibilidad del reuso de las mismas en distintas comunidades. Para ello se requeriría que una estrategia de ludificación en lugar de ser construida de cero, pudiese ser extendida o configurada para adaptarse a comunidades distintas de la comunidad en la que se creó. Esta extensión o configuración a su vez debería permitir centrar su estrategia de ludificación en aspectos de interés de la nueva comunidad.

Una estrategia de ludificación en un entorno colaborativo debería considerar fortalecer los aspectos clave como el valor emergente que la comunidad busca en cada participación y colaboración. Un aspecto a reforzar es motivar una mayor y mejor participación de los usuarios. La motivación es un elemento que constituye un eje a partir del cual se diseña un proceso de ludificación. Este eje, a su vez, depende del resultado que desee obtener con este diseño.

1.1. Problemática de la configuración de una estrategia de ludificación

Si bien la ludificación puede traer muchos beneficios también conlleva ciertas dificultades. Si por ejemplo, se desea utilizar una estrategia de ludificación existente, la problemática de ludificar una CCC implicaría:

- a) Identificar eventos o acciones de los participantes de la CCC que actuarán como acciones en la estrategia de ludificación.
- b) Ajustar la transformación de los eventos de la comunidad a acciones en la estrategia de ludificación de forma que se correlacionen creando una experiencia agradable. Lo cual puede implicar desde correlacionar un único evento en la comunidad con múltiples acciones en la estrategia de ludificación, a transformar dichos eventos a formatos que puedan ser procesados por en la estrategia de ludificación.
- c) Encontrar la mejor configuración de los parámetros o ajustes internos de la estrategia de ludificación para que la experiencia se adapte a los objetivos de la comunidad mientras permite a los jugadores avanzar por la misma.

Las dos primeras tareas requieren iteración y experimentación. Por ejemplo, suponiendo que se tienen acciones de distinto tipo ($A, B, C...$) y se recompensa a los usuarios por realizar cierta cantidad de un tipo de acciones o una combinación de ambas. Si solo se identifican eventos de un tipo A y no de los un tipos B o C , esto podría generar que no se pueda obtener ciertas recompensas que estén vinculadas a esos tipos, lo cual a su vez afectaría la dinámica del juego y la experiencia resultante podría no ser satisfactoria.

En cuanto a la tercera tarea, aborda la problemática de la configuración de la estrategia de ludificación propiamente dicha que se relaciona con el área de investigación de la configuración de sistemas[12]. Esta problemática, en conjunto con el hecho de que el resultado de la aplicación de una estrategia de ludificación no se puede predecir por completo [13, 7, 14], conduce a que aplicar ludificación pueda ser disruptivo para la comunidad, generando una competencia improductiva o desmotivando la participación [10, 15, 16] si la misma esta mal configurada. Por ejemplo, una estrategia de ludificación que promueve la competencia individual en una comunidad donde la colaboración debería ser el valor más relevante.

Además, aunque la estrategia de ludificación puede ser ajustada, el ajuste de la configuración de la estrategia de ludificación en producción (con la CCC real y usuarios reales) es arriesgado. Los usuarios pueden percibir los cambios en la estrategia de ludificación como errores del sistema o como cambios injustos. Para evitar esto se requeriría lanzar una versión estable de la estrategia de ludificación en la CCC, para ello es necesario considerar otras alternativas de ajuste basadas en simulaciones [17].

Sin embargo, encontrar la configuración de la estrategia ludificación que sea efectiva para una determinada CCC es complejo. Para encontrar una ludificación óptima, se requiere una estrategia de ludificación con una configuración que sea efectiva para la comunidad objetivo, dicha configuración debe adaptarse a la comunidad. Por lo tanto, la ludificación óptima, proporciona métodos o principios para ayudar a las personas a lograr sus objetivos y procrastinar menos. Lieder et. al. [18] proponen una adaptación de la estrategia de ludificación mediante un marco matemático utilizando pseudo-recompensas óptimas para guiar a los usuarios en las actividades que realice, sin embargo, este enfoque requiere conocimiento perfecto del entorno de decisión o la posibilidad de aproximarlos.

1.2. Hipótesis y enfoque de solución

En este trabajo, se busca apoyar la toma de decisión a fin de lograr una “ludificación óptima” desde la base de una estrategia de ludificación preexistente y tomando en consideración el contexto de incertidumbre inherente a la forma de trabajo colaborativo en las CCC. Por lo tanto, la hipótesis de este trabajo es aplicar agentes de aprendizaje por refuerzo en un entorno que simule de una estrategia de ludificación, en base a datos históricos, se puede apoyar la selección de variables de configuración óptimas para integrar la estrategia de ludificación en una CCC manteniendo su comportamiento intrínseco. De esta hipótesis se desprenden las siguientes afirmaciones a validar, las cuales a su vez guiarán el enfoque de este trabajo de tesis:

- Existen patrones de comportamiento de los miembros de las CCC los

cuales pueden encontrarse en sus datos históricos de actividad.

- A partir de los datos históricos del comportamiento de la comunidad también se podrá captar y mantener comportamientos relacionados a una estrategia de ludificación distinta a la cual se pretende configurar. Realizando una transferencia de conceptos de ludificación de una estrategia a otra.
- El enfoque de solución introducido en este trabajo es flexible a la integración de nuevas estrategias y aplicación de futuros avances en el área de RL.

1.3. Resultados obtenidos

En este trabajo de tesis se presenta una metodología para brindar apoyo a la toma de decisiones en el contexto de inserción de una estrategia de ludificación en una Comunidades de Construcción de Conocimiento. Como parte de esta metodología se presentan aplicaciones simuladas de una estrategia de ludificación en diversas comunidades. Demostrando que existen patrones dentro de los datos históricos de la comunidad que permiten la simulación del comportamiento de la misma. La primera de estas aplicaciones, la cual es configurada manualmente, demuestra las dificultades inherentes a la configuración de una estrategia de ludificación. También se presenta un análisis de los datos utilizados por estrategia de ludificación, junto con sus métodos de extracción y transformación para la simulación.

También se desarrolla un framework para facilitar la implementación de la metodología propuesta. Por lo tanto, este framework permite la optimización automática de configuraciones para estrategias de ludificación en CCC. Este framework permitió estudiar el desempeño de la metodología planteada, brindando flexibilidad y capacidad de extensión para la aplicación de la metodología en diferentes CCC. A partir de las aplicaciones del mismo se permite determinar configuraciones optimizadas de una ludificación para una comunidad existente manteniendo el comportamiento intrínseco de la misma.

Se presenta una evaluación de este framework en la CCC de Wikipedia. Particularmente se estudian los artículos de mayor calidad de Wikipedia y también la estrategia de ludificación preexistente llamada Wikicup. Durante dicha evaluación se comprueba la generación exitosa de configuraciones por parte del agente entrenado en las diferentes sub-comunidades. También se introduce el concepto de función de recompensa para el enfoque de RL, que nos permite observar las capacidades de adaptación del enfoque según los requerimientos o los objetivos planteados con cada sub-comunidad. Particularmente el caso de la Wikicup nos permite observar que es factible trasladar

conceptos de una estrategia de ludificación a otra exitosamente con la función de recompensa correcta.

Para validar la capacidad de generalización a otras CCC, se presenta una evaluación del framework en la CCC de Stack Overflow. Aquí se demuestra que con pocos cambios el framework permite cambiar fácilmente la CCC objetivo. También se evalúa la efectividad de la configuración encontrada para periodos posteriores de tiempo en la misma comunidad. Luego se analiza la posibilidad de optimizar dicha configuración y las dificultades que se presentan por cambios en la actividad por parte de la comunidad. Esta evaluación demuestra que las configuraciones obtenidas son efectivas en periodos de tiempo posteriores de la misma comunidad, aunque la misma pueda optimizarse aún más.

Por último, se valida la capacidad de extensión del enfoque y el framework en lo referente a nuevas estrategias o enfoques el área de RL. Para ello se realiza una evaluación para integrar diferentes tipos de agentes de RL junto con la aplicación de diferentes métricas y tratamiento de los datos de entrada recibidos por los agentes. Esta evaluación nos demuestra la facilidad de integración de nuevos agentes independientemente de la tecnología que los soporte. También nos permite comprender mejor el impacto de las variables de entrada y de las métricas relacionadas a las funciones de recompensa.

Con estas evaluaciones contenidas en los distintos casos de uso se logra por lo tanto validar el correcto cumplimiento de la hipótesis de este trabajo. Brindando la metodología, el framework que la soporta y casos de uso pertinentes para su posterior reproducción y extinción en futuras investigaciones, así como su aplicación en contextos no simulados.

1.4. Publicaciones

Durante el proceso de investigación se publicaron los siguientes artículos directamente relacionadas al tema de este trabajo de tesis. En todos los casos, se trató de publicaciones de alcance internacional:

- MARTIN, Jonathan; TORRES, Diego; FERNANDEZ, Alejandro. Patterns of markup use in Wikipedia. En 2017 36th International Conference of the Chilean Computer Science Society (SCCC). IEEE, 2017. p. 1-9. [19]
- MARTIN, Jonathan, et al. Using Citizen Science Gamification in Agriculture Collaborative Knowledge Production. En Proceedings of the XIX International Conference on Human Computer Interaction. 2018. p. 1-8. [13]
- MARTIN, Jonathan; TORRES, Diego; FERNANDEZ, Alejandro. Op-

timizing a Gamified Design Through Reinforcement Learning-a Case Study in Stack Overflow. En Conference on Cloud Computing, Big Data & Emerging Topics. Springer, Cham, 2021. p. 89-103. [20]

Adicionalmente, durante este período surgieron temas de investigación relacionados con temas relacionados a los presentados en este trabajo (como las CCC) que permitieron obtener conocimiento de diferentes dominios y desarrollar una visión mucho más amplia sobre la solución a implementar en el trabajo de tesis. Estos trabajos dieron origen a las siguientes publicaciones:

- ANTONELLI, Leandro; CAMILLERI, Guy; CHALLIOL Cecilia; FERNANDEZ Alejandro; HOZIKIAN, Mariangeles; GIANDINI, Roxana; GRIGERA, Julian; ALEJANDRA BEATRIZ, Alejandra; MARTIN, Jonathan; TORRES, Diego; ZARATE, Pascale. An extension to scenarios to deal with business cases for the decision-making processes in the agribusiness domain. En Agriculture Value Chain-Challenges and Trends in Academia and Industry. Springer, Cham, 2021. p. 35-50. [21]
- GARRIDO, Alejandra; ANTONELLI, Leandro; MARTIN, Jonathan; ALEMANY DÍAZ, Maria Del Mar; MULA, Josefa. Using LEL and scenarios to derive mathematical programming models. Application in a fresh tomato packing problem. Computers and Electronics in Agriculture, 2020, vol. 170, p. 105242. [22]
- ALEMANY DÍAZ, Maria Del Mar; ESTESO, Ana; ORTIZ, Angel; HERNÁNDEZ, Jorge; FERNÁNDEZ, Alejandro; GARRIDO, Alejandra; MARTIN, Jonathan; LIU, Shaoefeng; ZHAO, Guoquin; GUYON, Cecile; IANACCONE, Rina; HERNANDEZ, Jorge; KACPRZYK, Janusz. A Conceptual Framework for Crop-Based Agri-food Supply Chain Characterization Under Uncertainty. En Agriculture Value Chain Challenges and Trends in Academia and Industry. Springer, Cham, 2021. p. 19-33. [23]

1.5. Organización de la tesis

Esta tesis se organiza de la siguiente manera:

- El Capítulo 2 introduce el marco conceptual sobre el que se trabajó. Dentro de este Capítulo se introducen nociones de sistemas de soporte a la toma de decisión (DSS por sus siglas en ingles). Se introduce conceptos de CCC abordando ejemplos de Wikipedia y StackOverflow. También hace referencia a la ludificación de estas comunidades. Finalmente se introducen conceptos de Reinforcement Learning y de Reinforcement Learning aplicado al área de ludificación. Estos conceptos ayudaran a lo largo de la tesis para comprender los fundamentos sobre los que se construye este trabajo.

- Para validar la relevancia de los datos históricos para la simulación así como para presentar las herramientas de extracción que se utilizaran para los casos de uso relacionados con Wikipedia, el Capítulo 3 presenta un análisis del historial de revisiones que se usa como entrada de datos de Wikipedia. Este Capítulo nos permite comprobar la existencia de patrones de comportamiento en los datos utilizados para el enfoque de este trabajo.
- El Capítulo 4 presenta la introducción de una estrategia de ludificación de forma simulada en Wikipedia. Esto nos permitirá validar la posibilidad de integración de una estrategia de ludificación preexistente en una CCC. Con este fin, dentro de este Capítulo se detallarán los pasos necesarios para realizar dicha integración. También se presentará el framework teórico de llamado Models, Dynamics and Aesthetics (MDA), con el cual se estructurarán los componentes de nuestra estrategia de ludificación en Wikipedia y sobre el cual podrá analizarse los resultados de la integración. Luego la simulación de la ludificación se presenta y analiza sobre una sub-comunidad de Wikipedia. De esta simulación se analizan los resultados de la ludificación y se exponen las dificultades y problemáticas que surgen de esta integración manual. Esto nos brinda una perspectiva más clara de la problemática que se aborda en esta tesis.
- El Capítulo 5 se centra en introducir el enfoque de solución basado en Reinforcement Learning. Para explicar este enfoque se divide el Capítulo en una sección sobre la importancia de los datos históricos, una sobre la simulación de las estrategias de ludificación y finalmente como estos se relacionan con RL.
- El Capítulo 6 presenta el framework de Reinforcement Learning para automatización de configuraciones de ludificaciones en CCC desarrollado durante esta tesis. Dentro de este Capítulo se introducen las tecnologías utilizadas para su desarrollo como también las herramientas que la misma extiende. Se presentarán tecnologías utilizadas también posteriormente en los casos de uso por parte de los agentes de Reinforcement Learning. Este Capítulo facilita la comprensión de las características del framework que permitirá la evaluación de la metodología propuesta en este trabajo.
- El Capítulo 7 presenta casos de estudio sobre aplicación del framework detallando las extensiones necesarias para cada caso particular. Estos casos de estudio permitirán validar los distintos elementos de la hipótesis evaluando el framework y la metodología en distintas comunidades u objetivos. Dentro de este Capítulo se presenta un caso de estudio del framework en Wikipedia y otro en Stack Overflow. Sobre ambos se presentan los resultados respecto a las configuraciones

recomendadas y la evaluación histórica de la ludificación en cuestión. Finalmente se presenta un caso de estudio que realiza una comparación de agentes y los cambios en el desempeño de los mismos en base a las configuraciones del entorno simulado.

- El Capítulo 8 presenta finalmente las conclusiones generales de la tesis, así como sus limitaciones y sus posibilidades de trabajo futuro.

Capítulo 2

Marco conceptual

Este capítulo provee una revisión de diferentes conceptos relacionados a la presente propuesta en sus distintos puntos. Primero se introducen los sistemas de soporte a la toma de decisión o Decision Support System (DSS) los cuales se desean aplicar para estrategias de ludificación en Comunidades de Construcción de Conocimiento (CCC). Por lo cual se introduce a estas comunidades presentando sus características y principales desafíos. Luego se presentan los conceptos básicos sobre ludificación y sus aplicaciones. Finalmente se hablará de aprendizaje automático centrado en aprendizaje por refuerzo, como herramienta para el soporte a la toma de decisión relacionado con la configuración de una estrategia de ludificación en CCC. Introduciendo además trabajos relacionados a la aplicación de aprendizaje por refuerzo en ludificación que pueden compararse con nuestra propuesta, y cuáles son las principales diferencias o ventajas respecto de los mismos.

2.1. Sistemas de soporte a la toma de decisiones

El apoyo a la toma de decisiones es un área de estudio que surge según Fick et al.[24] de dos áreas de investigación que son los estudios teóricos de la toma de decisión y el trabajo sobre sistemas informáticos interactivos, a partir de los cuales surge la noción de sistemas de apoyo de decisión. Las decisiones que pueden verse apoyadas por estos sistemas varían ampliamente según el contexto en el que se apliquen. Estas decisiones pueden ser decisiones no estructuradas, como las que se encuentran en niveles administrativos que consisten en decisiones para las cuales no existen procedimientos previos o se carece de los métodos para resolver la decisión, o decisiones semiestructuradas donde se posee noción o parte de la respuesta, pero aún se requiere intervención humana en la toma de la decisión[25]. Por lo cual, podemos considerar que estos sistemas tienen como objetivo ayudar a tomar buenas decisiones, donde por “buenas decisiones” se refiere a las decisiones que to-

mamos cuando poseemos la información relevante y apropiada para decidir entre nuestras alternativas[26].

En esta tesis se tomará como definición de Sistemas de Apoyo a la toma de decisiones o Decision Support System (DSS), a un sistema informático que sustenta el proceso de toma de decisiones. Durante este proceso se pueden presentar la utilización de datos y modelos para la generación, la estimación, la evaluación o la comparación sistemática de diversas opciones, ayudando a los responsables de la toma de decisión a reunir información y tomar decisiones[27, 26, 28].

Dichos sistemas fueron adoptados en gran medida por su utilidad en el ambiente de *Business Intelligence* por permitir analizar múltiples variables de negocio, y presentarlas a través de por ejemplo tableros de control, para apoyar la toma de decisión de los directivos. Por otro lado, gracias a la flexibilidad entre las decisiones a las que pueden brindar soporte, los DSS han demostrado ser verdaderamente multidisciplinarios como por ejemplo en el campo de la medicina donde brindan ayuda para el seguimiento del paciente mediante la sugerencia de análisis recomendados en base a los síntomas indicados [29, 30], o en la agricultura apoyando decisiones referentes al riego o tipo de plantación preferente en base a datos históricos o climatológicos [31, 32].

Aunque los DSS se emplean en contexto de tomas de decisiones no estructuradas o semiestructuradas, algunas decisiones pueden ser más estimables por lo cual se podría hablar también de un sistema de recomendación. Un ejemplo en el que se plantean los elementos necesarios para un sistema de recomendación en CCC se presentan en el trabajo de Torres et al. [33], en el mismo se muestra un enfoque que permite recomendar a los editores cómo completar la información en Wikipedia a partir de información existente en DBpedia respetando las convenciones propias de las comunidades preexistentes.

Otro ejemplo de sistema de recomendación como DSS se da dentro del área de recomendación de configuraciones [34]. Zhang et. al[35] presenta un sistema de recomendaciones para apoyar el proceso de toma de decisión durante la selección de configuraciones facilitando la integración de servicios en la nube a partir de las aplicaciones específicas del usuario. Este tipo de sistemas de recomendación para el apoyo de toma de decisión si bien no son aplicados a CCC o estrategias de ludificación, permiten observar un enfoque similar al que se presenta en esta tesis en lo que respecta a la recomendación de configuraciones.

Como se mencionó previamente, para tomar buenas decisiones se requiere que la información sea apropiada, relevante y esté organizada en un modelo de forma tal de ayudar en la elección de la decisión correcta [36]. Los DSS

permiten obtener esta información incluso en entornos donde se producen grandes volúmenes de información en forma constante, generando grandes volúmenes de datos. Este contexto de grandes volúmenes de datos puede observarse tanto en redes sociales como Facebook [37] o CCC como Wikipedia. Este volumen de información dificulta su análisis de forma manual por la persona encargada de tomar una decisión.

Hay diferentes fuentes de las que puede obtenerse la información requerida para un DSS. Estas fuentes pueden ser de origen externo o interno a nuestro contexto. El caso de esta tesis el contexto son los elementos de conocimiento creados por las CCC, esto se debe a que es la actividad sobre estos elementos de la comunidad lo que se utilizara como estímulos o acciones de la ludificación que se desea integrar a la comunidad sobre la cual se requiere apoyo para la selección de configuraciones.

Existen fuentes externas al contexto de decisión que pueden enriquecer la información generada para el apoyo en la toma de decisión. Estas fuentes tienen como características que se forman fuera del contexto donde se toma la decisión. Esto impacta por que dicha información no necesariamente se encontrara en un formato que permita ser consumida o utilizada fácilmente por el DSS, requiriendo en muchos casos una adaptación de la información a fin de que la misma resulte de utilidad.

Como fuente interna a los contextos de decisión, se puede considerar a toda la información generada y almacenada por el propio sistema. Si bien la confianza en la información generada internamente suele ser mayor, la misma requiere ser analizada, sobre todo en contextos sociales como puede ser el de una CCC. Una herramienta que comparte, en parte, este fin es el análisis del provenance o la proveniencia. La proveniencia permite obtener información de procedencia sobre un determinado elemento. La información sobre la procedencia de un elemento es información sobre la historia del elemento, a partir de su creación, incluida la información sobre sus orígenes[38, 39]. El análisis de la proveniencia puede realizarse con información de las bases de datos internas del sistema, sobre el que se realiza la toma de decisión, a fin de obtener información sobre la evolución en el sistema sobre la pieza de información analizada. También pueden utilizarse copias de seguridad que registren todas las actividades realizadas a modo de logs o datos históricos, por ejemplo, la familia de las MediaWikis posee un historial de las ediciones que se hacen a cada artículo. Este tipo de información permiten generar métricas de confianza sobre esa información[40], a fin de identificar cuando un dato es de calidad o valioso.

Otra capacidad relevante de los DSS para esta tesis, es la de enfrentar entornos dinámicos o con incertidumbre. Un entorno dinámico es aquel en el cual las variables de decisión cambian a gran velocidad requiriendo por lo tanto una mayor velocidad de respuesta por parte del sistema para brindar

información oportuna al decisor. Por otro lado, un entorno de decisión con incertidumbre se caracteriza por el desconocimiento parcial o total de las variables que impactan en la toma de decisión o también por la incapacidad de comprender como una decisión impactara a futuro[41]

Como ejemplo de las técnicas aplicadas en DSS para lidiar con este tipo de entornos podemos observar en Alemany et. al. [42] el uso de programación lineal combinada con conjuntos difusos para crear modelos configurables para su DSS como forma flexible de lidiar con la incertidumbre. También puede aplicarse técnicas de caracterización o perfilado, conocido como por su nombre en inglés como *profiling*, sobre los datos a fin de lograr una mejor comprensión del entorno de toma de decisión para lidiar con la incertidumbre. Continuando con este último lineamiento, en esta tesis se presentan aplicaciones de *profiling* para estudiar las características y la viabilidad de los datos. En este contexto se aplicará principalmente minado de patrones sobre los datos de proveniencia obtenidos en el contexto de una comunidad de construcción de conocimiento, analizando así la generación de información y el comportamiento de los usuarios.

Junto con la información obtenida de la validez y relevancia de los datos, en esta tesis se eligió aplicar un entrenamiento mediante aprendizaje por refuerzo, también conocido como *Reinforcement Learning (RL)*. *Reinforcement Learning* es una técnica de aprendizaje automático, sobre la que se detallara posteriormente en este mismo capítulo, y se aplicara como medio para brindar el soporte a la toma de decisión en esta tesis.

2.2. Comunidades de construcción de conocimiento

Las Comunidades de Construcción de Conocimiento según Scardamalia et. al. [43] pueden verse como un conjunto de personas trabajando sobre objetos de conocimiento mediante la interacción social con el objetivo de mejorarlos. Siendo este proceso de forma iterativa y con la tecnología brindando apoyo desde una infraestructura facilitadora. Las comunidades que apoyan esta actividad utilizando un enfoque basado en la Web y son a las que nos referimos como Comunidades de Construcción de Conocimiento (CCC) [44]. Moskaliuk et. al. [45] presenta a Wikipedia como una de estas comunidades, centrándose en el proceso de construcción de conocimiento el cual tiene como objetivo nuevamente la producción y la mejora continua de piezas de conocimiento. Estas piezas de conocimiento también poseen la característica de ser una creación colectiva de conocimiento público.

En esta tesis por lo tanto se considerará a las Comunidades de Construcción de Conocimiento como aquellas que brindan un soporte computacional

al proceso de construcción de conocimiento. Dicho proceso por lo tanto consiste en la colaboración e interacción de múltiples individuos, cubriendo diferentes roles, durante múltiples iteraciones con el objetivo de producir y mejorar piezas de conocimiento.

2.2.1. Soporte por computadoras

Actualmente el soporte por computadora se está dando principalmente de forma online, permitiendo un acceso sencillo para todos los miembros de la Comunidades de Construcción de Conocimiento. Los miembros de la comunidad se encuentran comúnmente modelados dentro del sistema de la comunidad con facilidades para la gestión de usuarios. Esta gestión de usuarios debe de igual forma permitir la gestión de roles dentro de la comunidad y la posibilidad de variar entre ellos dado que un usuario puede cumplir múltiples roles durante diversas etapas del proceso de construcción de conocimiento. Por ejemplo, un mismo usuario en una etapa puede estar agregando contenido en una pieza de conocimiento, mientras que en una etapa posterior puede estar actuando a modo de revisor del contenido de dicha pieza o coordinando los esfuerzos de otros usuarios para mejorar dicha pieza de conocimiento.

También es necesario, por lo tanto, que se caracterice o se modele la pieza de conocimiento dentro del sistema, esto puede darse de muchas formas, por ejemplo, una estructura muy básica podría ser la de un documento con un título y un contenido que pueda ser redactado por los usuarios. Este elemento tiene que ser fácilmente editable y puede variar su formato según la comunidad como veremos más adelante. Es común observar que el sistema permita la interacción de varios usuarios sobre estas piezas de conocimiento y que se brinde la posibilidad de categorizarlas para facilitar la búsqueda entre dichas piezas.

Un aspecto relevante que refuerza el soporte por computadora es la capacidad de coordinar a los diversos usuarios que actúan sobre una misma pieza de conocimiento. Esta coordinación puede verse durante la interacción sobre las piezas de conocimiento, así como también en la comunicación entre los usuarios para la coordinación de esfuerzos. Dicho soporte puede ser de forma sincrónica, permitiendo la edición conjunta en forma simultánea de una misma pieza de conocimiento, así como también la comunicación sincrónica por ejemplo mediante salas o espacios de chat. O puede llevarse a cabo de forma asincrónica, por ejemplo, con ediciones turnadas o por manejo de versiones de las piezas de conocimiento, y en lo referente a la comunicación puede ser mediante mensajería tradicional como un mailing o foros de discusión[43].

Transversal a todos estos elementos, el soporte por computadoras permite registrar todo lo que sucede dentro de la comunidad, desde las inter-

acciones de cada usuario de forma individual, como su participación sobre las piezas de conocimiento, como actividades comunitarias de coordinación o participación grupal. Este registro puede darse mediante el resguardo de la información tanto en base de datos tradicionales como en logs de actividad, o en base de datos semánticas con el uso de ontologías. Por medio de esta característica que brinda el soporte por computadora es posible generar proveniencia de información.

El estudio de la proveniencia es aplicado para comprender la evolución de las piezas de conocimiento o el comportamiento de los usuarios a lo largo del tiempo. Con esta información, es posible, reconstruir el proceso de escritura de principio a fin de una pieza de conocimiento [46]. Otra posibilidad es detectar y clasificar el comportamiento de los usuarios[47], permitiendo detectar roles implícitos en la comunidad o detectar acciones destructivas, vandálicas o simplemente contrarias a los objetivos de la comunidad. También es posible mediante el mismo estudiar los procesos que pueden ser indicativos de calidad en las piezas de conocimiento de la comunidad[48]. Finalmente, también puede permitir la detección de patrones, por ejemplo, sobre uso de las herramientas proporcionadas para realizar el proceso de creación de conocimiento puede observarse como las mismas se utilizan a fin de predecir qué elementos se utilizan en conjunto o con que finalidad.

Teniendo en consideración estas características brindadas a partir del soporte por computadoras, se seleccionaron Comunidades de Construcción de Conocimiento que brindaran soporte en todos estos aspectos y las mismas se introducen a continuación.

2.2.2. Wikipedia y Stack overflow

Wikipedia y Stack Overflow son comunidades bien conocidas con una gran cantidad de usuarios[49], siendo Wikipedia en su versión inglesa la que presenta más usuarios, con 39.355.070 usuarios registrados¹ y Stack Overflow con 12.615.110 usuarios registrados².

Ambas comunidades son Comunidades de Construcción de Conocimiento que promueven la construcción del conocimiento en distintos formatos. Wikipedia por un lado enfoca la construcción de conocimiento en formato enciclopédico donde los usuarios actúan como editores y revisores de la misma. Por otro lado, Stack Overflow cubre el proceso de construcción de conocimiento desde una perspectiva de preguntas y respuestas, donde cada usuario puede variar entre los roles de consultante y quien brinda respuestas, a fin de cubrir los requerimientos de la comunidad. Ambas comunidades

¹Información obtenida el 25/06/20 en <https://en.wikipedia.org/wiki/Special:Statistics>

²Información obtenida el 25/06/20 en <https://data.stackexchange.com/stackoverflow/query/1255610>

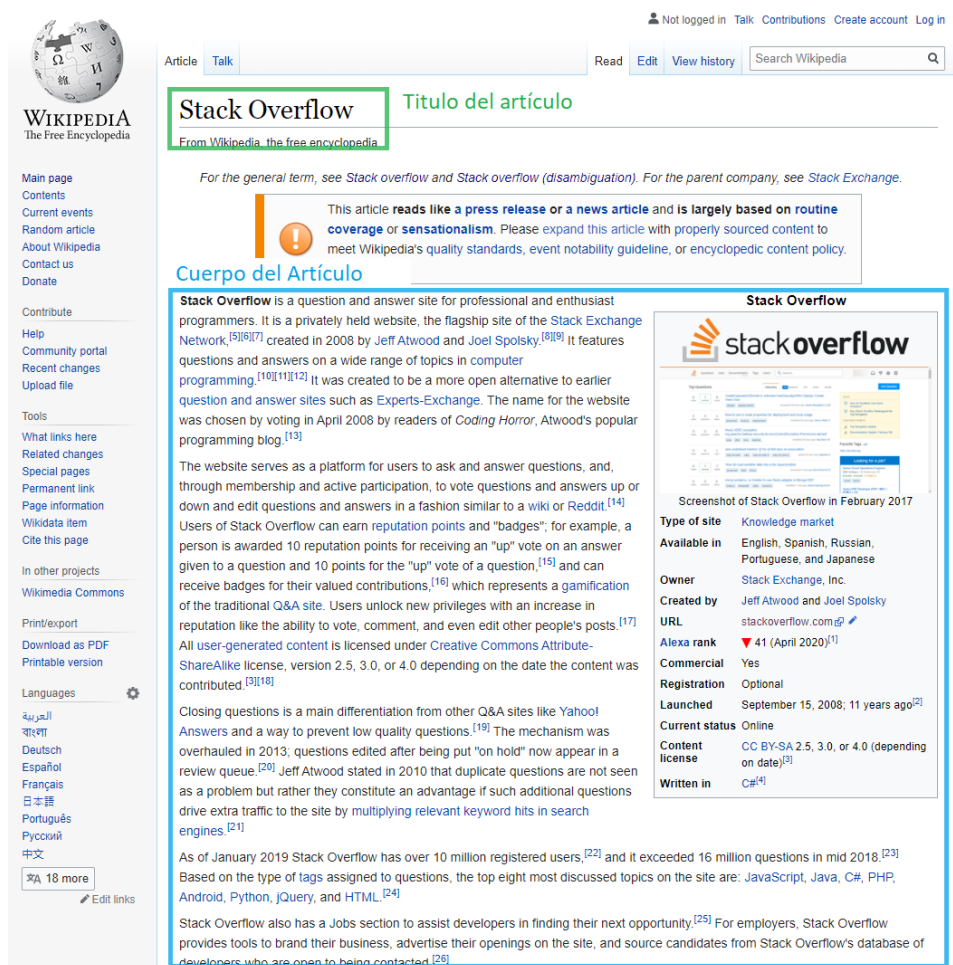


Figura 2.1: Artículo de Stack Overflow en Wikipedia en su versión inglesa.

brindan soporte por computadora permitiendo un acceso web a la misma, modelizando además de estos roles de usuarios roles que permiten la moderación o gestión de las comunidades.

Otro aspecto que modelan acorde al soporte por computadora son los elementos considerados en este trabajo como piezas de conocimiento los cuales representan la unidad atómica en la cual se agrupa un conocimiento determinado. Wikipedia en su formato enciclopédico estructura las piezas de conocimiento por artículos dedicados a un tópico de conocimiento y a su vez en categorías que contienen varios artículos de tópicos similares, un ejemplo de un artículo de Wikipedia puede observarse en la Figura 2.1.

Como puede observarse en la Figura 2.2, estos artículos pueden ser creados y modificados libremente sin importar si el usuario se encuentra regis-

Article [Talk](#) [Read](#) [Edit](#) [View history](#)

Editing Stack Overflow

Se puede editar sin estar conectados

You are not logged in. Your IP address will be publicly visible if you make any edits. If you [log in](#) or [create an account](#), your edits will be attributed to a user name, among other benefits.

Content that violates any copyrights will be deleted. Encyclopedic content must be verifiable. Work submitted to Wikipedia can be edited, used, and redistributed—by anyone—subject to certain terms and conditions.

Editor de artículos

```

{{about|the general term|Stack overflow|and|Stack overflow (disambiguation)|the parent company|Stack Exchange}}
{{short description|Website hosting questions and answers on a wide range of topics in computer programming}}
{{Cleanup-PR|date=May 2020}}
{{Use dmy dates|date=March 2012}}
{{Infobox website
| name = Stack Overflow
| logo = Stack Overflow logo.svg
| screenshot = Stack Overflow homepage, Feb 2017.png
| caption = Screenshot of Stack Overflow in February 2017
| url = {{Official URL}}
| alexa = {{Decrease}} 41 {{as of|2020|04|14|alt=April 2020}}<ref>{{cite web |title=stackoverflow.com Competitive Analysis, Marketing Mix and Traffic |url=https://www.alexa.com/siteinfo/stackoverflow.com |website=Alexa.com |accessdate=14 April 2020}}</ref>
| commercial = Yes
| programming_language = [[C Sharp (programming language)|C#]]<ref name="roadchap" />
| content_license = [[Creative Commons license|CC BY-SA]] 2.5, 3.0, or 4.0 (depending on date)<ref name="license split 3.0 4.0"/>
| type = [[Knowledge market]]
| language = English, Spanish, Russian, Portuguese, and Japanese
| registration = Optional
| owner = [[Stack Exchange]], Inc.
| author = [[Jeff Atwood]] and [[Joel Spolsky]]
| launch_date = {{start date and age|2008|9|15}}<ref name="launches">{{cite web |accessdate=2014-07-07 |url=http://www.joelonsoftware.com/items/2008/09/15.html |title=Stack Overflow Launches |last=Spolsky |first=Joel |date=2008-09-15 |publisher=Joel on Software }}</ref>

```

Insert Cite your sources: <ref></ref>

[Edit summary](#) (Briefly describe your changes)

By publishing changes, you agree to the [Terms of Use](#), and you irrevocably agree to release your contribution under the [CC BY-SA 3.0 License](#) and the [GFDL](#). You agree that a hyperlink or URL is sufficient attribution under the Creative Commons license.

Figura 2.2: Edición artículo de Stack Overflow en Wikipedia en su versión inglesa.

trado en el sistema o no, generando lo que se conoce como revisiones de un artículo. Además este editor presenta funcionalidades para la edición de texto plano, agregado de estilos al texto y se lo puede enriquecer utilizando elementos multimedia como imágenes. Finalmente permite observar una vista previa del artículo con los cambios y publicarlos junto con un comentario informativo de los cambios realizados.

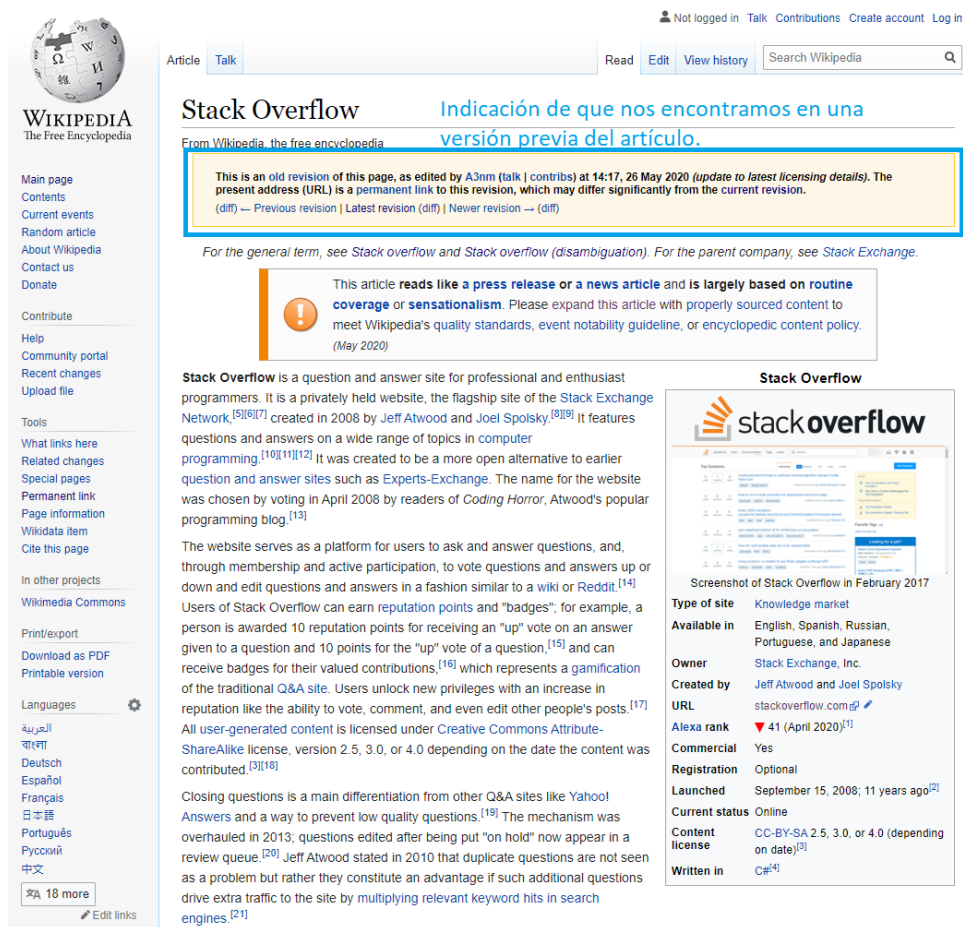


Figura 2.3: Artículo de Stack Overflow en Wikipedia en su versión inglesa, versión del 26 de Mayo del 2020.

Una revisión de un artículo, como la que puede observarse en la Figura 2.3, representa una copia completa del artículo con todo su contenido luego de sufrir una edición.

Estas copias a su vez son almacenadas en un historial de revisiones como se aprecia en la Figura 2.4. Stack Overflow modela sus piezas de conocimiento en el formato de preguntas/respuestas, esto puede verse en la Figura

Article Talk Read Edit View history

Stack Overflow: Revision history Help

[View logs for this page \(view filter log\)](#)

Filter revisions

External tools: [Find addition/removal](#) (Alternate) · [Find edits by user](#) · [Page statistics](#) · [Pageviews](#) · [Fix dead links](#)

For any version listed below, click on its date to view it. For more help, see [Help:Page history](#) and [Help:Edit summary](#).
 (cur) = difference from current version, (prev) = difference from preceding version,
m = minor edit, → = section edit, ← = automatic edit summary
 (newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

Compare selected revisions Versión actual del Artículo

- (cur | prev) 22:15, 21 June 2020 AnomieBOT (talk | contribs) **m** . . (20,624 bytes) (0) . . *(Dating maintenance tags: {{Citation needed}})* (undo)
- (cur | prev) 20:14, 21 June 2020 Mostafa 36a2 (talk | contribs) **m** . . (20,624 bytes) (+34) . . *(→ Technology)* (undo)
- (cur | prev) 14:18, 26 May 2020 A3nm (talk | contribs) **m** . . (20,590 bytes) (0) . . *(typo)* (undo) (Tag: 2017 wikitext editor)
- (cur | prev) Fecha de edición A3nm (talk | contribs) . . (20,590 bytes) (+63) . . *(update to latest licensing details)* (undo) (Tag: 20)
- (cur | prev) 13:36, 25 May 2020 Unforgettableid (talk | contribs) **m** . . (20,527 bytes) (+27) . . *(Comentarios relacionados con la edición)*
- (cur | prev) 13:35, 25 May 2020 Unforgettableid (talk | contribs) **m** . . (20,556 bytes) (+29) . . *(Wikipedia:Handling trivia s.v. "If an item is too large to bold and remove it".)*
- (cur | prev) 03:04, 25 May 2020 DGG (talk | contribs) . . (21,260 bytes) (+29) . . *(Added {{Cleanup-PR}} tag (TW))* (undo)
- (cur | prev) 22:42, 22 May 2020 Esprit15d (talk | contribs) . . (21,231 bytes) (+424) . . *(→ Reception: {{subst:2020 Webby Awards|Webby People's Voice Award for Community|Web}})* (undo)
- (cur | prev) 01:29, 27 April 2020 98.200.15.39 (talk) . . (20,807 bytes) (-225) . . *(No citation on the first sentence, but really just wanted to alert you to my talk post. I'm not sure if I did it right or if you'd be aware. Not really trying otherwise to edit this main page again.)* (undo)

Figura 2.4: Historial de revisiones del artículo de Stack Overflow en Wikipedia en su versión inglesa.

2.5 donde se ve una pregunta que representa el tópico o elemento de conocimiento mientras que la respuesta complementa el contenido con material de valor para el consultante. También se pueden observar elementos de gestión de votos o comentarios de los cuales se tratara con mas detalle a continuación.

Ambas plataformas permiten la coordinación de usuario por distintos medios. El caso de Wikipedia, la comunicación se da mediante páginas de discusión propias de cada artículo, donde su funcionamiento es idéntico al de los artículos con sus revisiones, pero los usuarios suelen seguir una organización de escritura similar a la de un foro con publicaciones y respuestas, un ejemplo de esto está en la Figura 2.6 donde puede verse la escritura indentada con la firma de distintos usuarios al finalizar cada respuesta.



Figura 2.6: Página de discusión del artículo de Stack Overflow de Wikipedia en su versión inglesa.

En Stack Overflow la comunicación y coordinación de usuarios se da mediante las propias preguntas y respuestas con la posibilidad además de agregar comentarios a las mismas o dar retroalimentación mediante calificaciones positivas o negativas, todas estas características pueden observarse en el ejemplo de la Figura 2.5 donde tanto la pregunta como la respuesta tienen múltiples votos positivos y además en el caso de la respuesta podemos observar los comentarios adicionales que recibió. Además, la Figura 2.7 muestra un sistema de comunicación privada entre usuarios para coordinar temáticas más específicas como infracciones a las normas de la comunidad o contacto entre usuarios.

Como se mencionó anteriormente gracias al soporte por computadora el cual mantiene registro de las actividades que se realizan en la comunidad se

The screenshot shows a Stack Overflow page for the question "Calling an external command from Python". The question has 4912 votes and 62 answers. The top answer, by Corey Goldberg, has 4721 votes and is marked as the best answer. Annotations include a purple box around the vote counts, a blue box around the question title and details, a green box around the top answer, and a red box around a comment on the answer. A red label "Comentarios de la respuesta" points to the comment box.

Pregunta

Calling an external command from Python

Asked 11 years, 9 months ago Active 2 months ago Viewed 3.4m times

Suma de votos positivos y negativos de la pregunta: 4912

add a comment Posibilidad de comentar la pregunta

62 Answers

Respuesta

Look at the `subprocess` module in the standard library:

Suma de votos positivos y negativos de la respuesta: 4721

The advantage of `subprocess` vs. `system` is that it is more flexible (you can get the `stdout`, `stderr`, the "real" status code, better error handling, etc...).

The [official documentation](#) recommends the `subprocess` module over the alternative `os.system()`:

Comentarios de la respuesta

40 @KevinWheeler You should NOT use `shell=True`, for this purpose Python comes with `os.path.expandvars`. In your case you can write: `os.path.expandvars("$PATH")`. @SethMMorton please reconsider your comment -> [Why not to use shell=True](#) - Murlmel Nov 11 '15 at 20:24

5 If you want to create a list out of a command with parameters, a list which can be used with `subprocess` when `shell=False`, then use `shlex.split` for an easy way to do this [docs.python.org/2/library/shlex.html#shlex.split](#) - Daniel F Sep 20 '18 at 18:05

show 9 more comments

Figura 2.5: Pregunta realizada en Stack Overflow con su respuesta más votada.

obtiene la posibilidad de generar proveniencia de la información. La particularidad en este tópico por parte de estas comunidades es que brindan acceso abierto a esta proveniencia de forma pública permitiendo análisis externos a los propios de la comunidad. Ya sea mediante el historial de revisiones con el lenguaje de marcado “wikitext” en Wikipedia o con las copias de la base de datos de Stack Overflow accesibles mediante BigQuery, el análisis de la información registrada permite realizar múltiples estudios algunos de los cuales se mencionan a continuación.

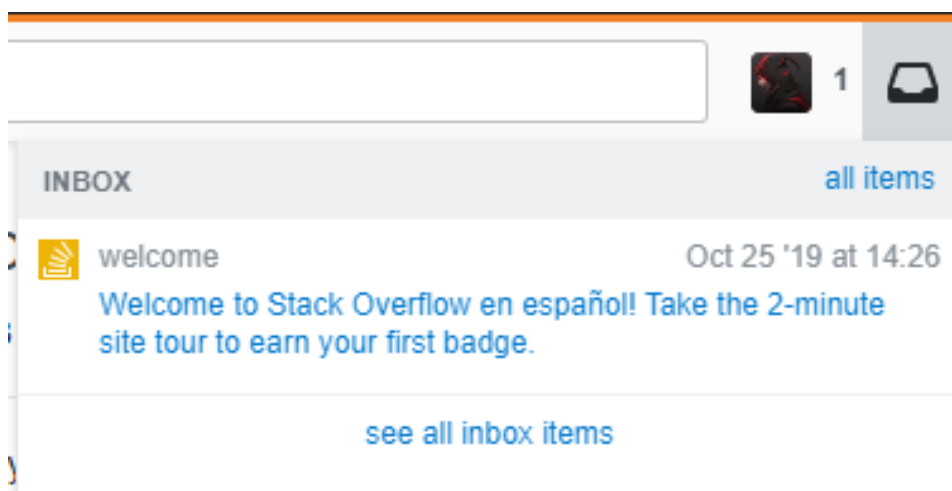


Figura 2.7: Sección de mensajería privada de Stack Overflow.

En el contexto de Wikipedia con esta información, es posible, por ejemplo, reconstruir el proceso de escritura desde el principio hasta el final [46]. También en relación con Stack Overflow se encuentran múltiples estudios relacionados con la detección y clasificación del comportamiento del usuario. Esto permite la categorización de roles implícitos en las comunidades [47, 50, 51, 52, 53], o predecir la participación o niveles de participación de los usuarios a lo largo del tiempo y el impacto del nivel de participación en la calidad del contenido [54, 55, 48, 56]. Relacionado con el comportamiento de los usuarios también se analiza la detección de actos vandálicos o nocivos para las comunidades [57, 58, 59].

En el contexto de medir la participación de los usuarios y como esta impacta en la calidad del contenido es donde aparece una problemática común a las Comunidades de Construcción de Conocimiento, como alentar la participación de la comunidad para generar más contenido y mejorar el existente. Esta es la problemática enfocada en esta tesis, la cual se busca afrontar mediante la ludificación de dichas comunidades.

2.3. Ludificación

Los sistemas de información se han ido adaptando para satisfacer cada vez más las motivaciones de los usuarios[60, 61]. Si bien existe la separación de sistemas según si atienden necesidades utilitarias como si atienden necesidades hedónicas[62], cada vez más, los sistemas se diseñan para atender a ambas necesidades, es decir, como sistemas mixtos. Los sistemas mixtos son los que tienen como objetivo motivar a los usuarios hacia diferentes comportamientos beneficiosos a niveles individuales y colectivos [6, 7].

El objetivo de estos sistemas continúa siendo utilitario y está relacionado con la productividad. Sin embargo, los medios y el diseño que promueven la productividad son de naturaleza hedónica[63, 64]. Como los sistemas utilitarios tradicionales apuntan a la productividad a través de la eficiencia, y los sistemas hedónicos tradicionales apuntan a la creación de experiencias divertidas, estos sistemas de información pueden caracterizarse por apuntar a la “productividad a través de la diversión”.

A estos sistemas se los conoce como Sistemas de Información Motivacional, e integran de los sistemas de información utilitarios y hedónicos que la aceptación este impulsada principalmente por la utilidad, como en los sistemas utilitarios, y que dicha utilidad este determinada por el disfrute del uso. Dentro de los Sistemas de Información Motivacional, el enfoque que aprovecha las características hedónicas que surgen de los juegos digitales se conoce comúnmente como ludificación[65].

La ludificación es comúnmente definida como “el uso de elementos de diseño de juegos en contextos no relacionados con el juego”[8]. Aunque, podemos encontrar definiciones más extensas centradas al objetivo de la ludificación como la de Kapp et. al.[66]: “la ludificación es el uso de la mecánica, la estética y el pensamiento del juego para involucrar a la gente, motivar la acción, promover el aprendizaje y resolver problemas”.

Además ludificación adquiere diferentes características en base al contexto donde se aplica[67]. Por lo cual, al elegir elementos de la ludificación, es necesario conocer previamente las necesidades de los usuarios, ya sea en un contexto individual o en una comunidad colaborativa[68].

Estos elementos lúdicos permiten la creación de distintos escenarios, que pueden orientarse a la competición o a la colaboración[69]. Mientras que las ludificaciones de tipo competitivo buscan alentar la participación y el perfeccionamiento individual, las ludificaciones colaborativas alientan la coordinación grupal a fin de cumplir objetivos comunes. Según el entorno en el que se desee realizar aplicar una estrategia de ludificación un tipo u otro tendrá mayor impacto. El caso de las CCC presenta un contexto adecuado para ludificaciones del tipo colaborativas.

Una problemática común a las ludificaciones es encontrar la mejor configuración de los parámetros internos de la estrategia de ludificación, con los cuales podríamos definir variaciones en la dificultad mediante la modificación de valores que definan por ejemplo en el contexto de una ludificación con niveles, la cantidad de niveles o puntos necesarios entre los niveles con el fin de que la experiencia sea escalable y se ajuste a los objetivos del contexto. Esta problemática se relaciona con la configuración de la estrategia de ludificación propiamente dicha, esto a su vez se relaciona directamente con el área de investigación de la configuración de sistemas[12]. Junto con esta problemática se presenta el hecho mencionado anteriormente de que el resultado de la aplicación de una estrategia de ludificación no se puede predecir por completo.

Además, aunque la estrategia de ludificación puede ser ajustada, el ajuste de su configuración durante su funcionamiento es arriesgado. Los usuarios pueden percibir los cambios en la mecánica de la estrategia de ludificación como errores del sistema o como cambios injustos. Para evitar esto se requiere lanzar una versión estable de la estrategia de ludificación, con este fin es necesario considerar otras alternativas de ajuste basadas en simulaciones [17]. En esta tesis se descartó la simulación sintética del comportamiento de los usuarios ya que por un lado había demasiadas variables involucradas y por el otro se consideró que existía otra opción más fiel para representar el comportamiento de los usuarios en una estrategia de ludificación particular. Como alternativa, se adoptó el marco conceptual Orta & Ruiz [70] y se exploró ejecutar la estrategia de ludificación en datos históricos de CCC existentes.

Sin embargo, encontrar la configuración que sea óptima para una determinada CCC es complejo. Para encontrar una ludificación óptima, una estrategia de ludificación que se ajuste correctamente a los usuarios y cumpla con los objetivos deseados con los que fue integrada al sistema, la configuración de la estrategia de ludificación debe adaptarse a la comunidad. Lieder et. al. [18] proponen la adaptación de una estrategia de ludificación mediante un marco matemático utilizando pseudo-recompensas óptimas para guiar a los usuarios en la actividad por lo que se requiere conocimiento perfecto del entorno de decisión o la posibilidad de aproximarlos. En cambio, en esta tesis se utilizarán técnicas de *Reinforcement Learning* a fin de obtener una ludificación óptima para una CCC. Como conclusión, la ludificación óptima, puede proporcionar por lo tanto una forma o principios para ayudar a las personas, o la comunidad, a lograr sus objetivos y procrastinar menos.

2.3.1. MDA Framework

Dado que el propósito de ludificación es diferente al de la creación de juegos[71], la aplicación de *frameworks* que guíen el proceso de ludificación

es requerido. Si bien existen diversos *frameworks* de ludificación mayoritariamente se basan en el *framework* de MDA[72, 73, 74, 75, 76, 77].

MDA es un *framework* genérico utilizado tanto en el diseño de juegos como de entornos de ludificación[78]. Este *framework* busca cubrir la brecha entre el diseño/desarrollo de los juegos, y la investigación técnica de los juegos, a partir de un enfoque estructural formal para definir y comprender los juegos. Con este fin, propone una división del sistema ludificado en tres ejes:

- Mecánicas (Mechanics), que describe los componentes particulares del juego, como las reglas, las acciones y los mecanismos de control. Se distinguen tres tipos de mecánica: mecánica de configuración (Setup mechanics), que determina la configuración de la experiencia del entorno; Mecánica de reglas (Rule mechanics), que configuran el objetivo de la experiencia; y la mecánica de progresión (Progression mechanics) que se refiere a las herramientas que se agregan para influir en la experiencia del juego [79].
- Dinámicas(Dynamics), que describe el desarrollo a lo largo del tiempo de las mecánicas a través de las respuestas de los usuarios y otros factores que pueden intervenir en el sistema. Describe comportamientos e interacciones. Los entornos de trabajo en equipo o grupos de individuos con objetivos comunes pueden generar dinámicas como la cooperación [79].
- Estéticas (Aesthetics), que son las respuestas emocionales en los usuarios cuando interactúan con el sistema cuando siguen la mecánica y generan dinámicas. [78] distingue ocho categorías, pero sin limitarlas a expandirse. Estas categorías son sensación, fantasía, narrativa, desafío, compañerismo, descubrimiento, expresión y sumisión. También se podría considerar disfrute o compromiso.

Metagame

Un ejemplo de aplicación de MDA en ludificación es Metagame[80]. Metagame es un juego transversal basado en recompensas de insignias con impacto en diferentes proyectos. En estos proyectos, diferentes personas (llamadas jugadores) participan realizando tareas específicas que se registrarán en Metagame y, dependiendo del tipo de acción, los jugadores reciben insignias como recompensa. La participación en Metagame es indirecta en un modo no intrusivo.

La diversidad de insignias que se pueden obtener está directamente relacionada con el tipo proyecto que integra Metagame. Como las principales acciones y propósitos son diversos entre proyectos, cada proyecto tiene que

definir sus insignias utilizando distintas *unidades de trabajo* que serán otorgadas por Metagame. Por ejemplo, en un proyecto de clasificación de imágenes, la insignia podría ser obtenida por un participante después de clasificar cinco imágenes; mientras que, en un proyecto de localización de mapas, la insignia podría ser obtenida después de que un participante marque un punto en el mapa.

Este ejemplo ilustra diferentes niveles de esfuerzo para lograr una tarea inspirada en un típico proyecto de ciencia ciudadana: clasificar una imagen requiere menos esfuerzo que caminar para detectar un problema en la calle. La dificultad y la magnitud para obtener una insignia en cada caso tienen impactos directos en Metagame.

Metagame diferencia entre tres tipos de acción: Contribución, Refuerzo y Difusión.

Contribución: Estas son las principales acciones que un jugador realiza. Estas acciones reflejan el objetivo principal del proyecto. Por ejemplo, en un proyecto de clasificación de imágenes, la acción de clasificar una imagen sería una contribución.

Refuerzo: Estas son acciones secundarias o de refuerzo que un jugador puede realizar. Representa una opinión o crítica de una acción anterior como una contribución. Por ejemplo, un jugador puede estar de acuerdo o en desacuerdo con un criterio de clasificación.

Difusión: Se trata de acciones sociales de propagación y difusión de otras actividades. Compartir un resultado en Facebook o Twitter es un ejemplo de difusión.

Rangos de jugadores en Metagame

Cada vez que un jugador recibe una insignia, está avanzando a un nuevo nivel del juego. Los jugadores en Metagame recorren una narración sobre un mundo de ciudadanos científicos en la cual, al principio, los jugadores se inician como visitantes. Luego a medida que cumplen los objetivos establecidos por Metagame, alcanzarán mejores rangos como “Ciudadano Científico” o “Ciudadano Científico Visionario”. La Tabla 2.1 muestra la lista completa de rangos en Metagame. La primera columna contiene la calificación del jugador y la segunda columna detalla el tipo y la cantidad de insignias necesarias para obtener el rango.

El diseño de los rangos en Metagame no es lineal, en cambio plantea el avance de los jugadores de forma exploratoria permitiendo que los mismos se centren en las actividades de mayor interés para ellos, y finalmente llegando a una meseta en la que se busca retener al jugador mediante el requisito de actividades mensuales. El diseño de los niveles se basa en una narración

Rango del jugador	Insignias necesarias
Visitante	Ninguna
Explorador	<i>Yo estuve aquí / Bienvenido de nuevo</i>
Ciudadano Científico (CC)	<i>Unidad de trabajo / De ida y vuelta / Grita en voz alta</i>
CC Prolífico	10 insignias
- CC Prolífico Especialista	x de 10 insignias en el mismo proyecto
- CC Prolífico Generalista	y de 10 insignias en diferentes proyectos
- CC Prolífico Networker	j de 10 insignias de refuerzo
-CC Prolífico Difusor	k de 10 insignias de diseminación
Ciudadano Científico Comprometido	1 una medalla por mes

Tabla 2.1: Niveles de experiencia en Metagame (Rango del jugador).

donde cada rango tiene un nombre que representa un momento en la vida de un científico. Los rangos usados en Metagame son:

- **Visitante:** Es el primer rango, con el cual todos los jugadores inician cuando comiencen en Metagame. Para pasar al siguiente rango, el jugador tiene que obtener las dos primeras insignias (“Yo estuve aquí” y “Bienvenido de nuevo”).
- **Explorador:** Metagame espera que los jugadores de este rango aprendan la mecánica del juego, por lo tanto, para pasar al siguiente rango se requiere que el jugador obtenga una insignia de cada tipo principal: Contribución, Refuerzo y Difusión. Una vez que se obtienen estas insignias, el jugador pasará al siguiente rango.
- **Ciudadano Científico:** El jugador deja de ser un novato, y pasa a ser un jugador regular en la plataforma. El Ciudadano Científico es libre de obtener cualquier insignia, y para crear un hábito, en Metagame se les reta a obtener diez nuevas insignias, para avanzar al siguiente rango, “Ciudadano Científico Prolífico”.
- **Ciudadano Científico Prolífico:** Estos jugadores son habituales en Metagame. Pueden ser especializados según el tipo de acciones que realizan.
 - **Generalista:** Son los Ciudadano Científico Prolíficos que han recibido insignias de más de un proyecto en Metagame. Este rango motiva a los jugadores a participar en varios proyectos.
 - **Especialista:** Con este perfil el jugador estará motivado a obtener una mayoría de insignias de tipo Contribución.
 - **Diseminador:** Con este perfil el jugador estará motivado a realizar mayoritariamente acciones de difusión.

- **Networker:** Con este perfil el jugador estará motivado a realizar mayoritariamente acciones de refuerzo.
- **Ciudadano Científico Comprometido:** Este es el nivel máximo que un jugador puede alcanzar. Estos son jugadores fieles en Meta-game. Sin embargo, este es el único rango en el que el jugador debe permanecer activo, o de lo contrario, su rango será revocado y volverá a Prolífico Ciudadano Científico.

2.3.2. Ludificación en CCC

En las CCC, las estrategias de ludificación se centran en aumentar la motivación y el compromiso, ya sea tanto a corto como a largo plazo[81]. Para diseñar estas estrategias en CCC, se debe tener en cuenta: a) el tipo de *framework* que se utiliza para el diseño; b) el contexto, lo que significa si se refiere a una comunidad online o a una pequeña comunidad privada y; c) la relación de los usuarios en relación con la comunidad[82].

La ludificación ya se está utilizando en múltiples dominios como la recolección de contenidos generados por el usuario, información social móvil y la detección ciudadana[83]. También en ciencia ciudadana, la cual es un proceso de colaboración donde los voluntarios trabajan con científicos profesionales para estudiar los problemas del mundo real[84].

Más cercano a las CCC, el trabajo de Marques [85] tuvo como objetivo obtener un sistema de gestión y construcción de conocimiento, en el cual menciona que será alimentado por expertos y que se implementará utilizando un entorno Wiki. Además, para mejorar la participación de los usuarios en la Wiki se incluyeron elementos de ludificación. Esos elementos de ludificación fueron los pluggings de Wikipedia llamados Xquiz³, para crear cuestionarios y publicar sus resultados, se confeccionó una tabla de acciones que define una puntuación por acción, con lo que después se calculaba un ranking de los usuarios y sus puntos.

Relacionado con la ludificación de Wikipedia, el trabajo de Narayan et. al.[86] presenta un tutorial interactivo, llamado **La aventura de Wikipedia** o **The Wikipedia Adventure (TWA)** que ofrece una introducción estructurada y ludificada a Wikipedia. Su objetivo era integrar a los nuevos usuarios en Wikipedia con sus complejas normas. Esta estrategia de ludificación aplicada en un tutorial se esperaba que mejorara la integración del nuevo usuario. Además, TWA se aplicó directamente a los proyectos de Wikipedia sobre la comunidad de usuarios, en nuestro caso la estrategia de ludificación se desarrolló como simulación sobre datos históricos.

³<http://extensions.xwiki.org/xwiki/bin/view/Extension/XQuiz/> Accedido el 25 de abril de 2018

Por último, otros trabajos relacionados con la ludificación en el proceso de construcción de conocimiento pero no implementados sobre Wikis es desarrollado por Senabre[87] y se llama **Gamification Quest:*** o **GQ***. Se centra en la presentación de una mecánica y un sistema de juego que utiliza las notas y entregas del estudiante, recogidas por la aplicación GQ. De manera de motivarlo a mejorar su rendimiento académico para progresar en el juego.

2.4. Reinforcement Learning

El algoritmo Reinforcement Learning es una técnica de aprendizaje automático que difiere de los algoritmos de aprendizaje supervisado porque no requiere los pares correctos de entrada/salida en el punto de partida, y sólo se utiliza un objetivo claro para establecer la política que maximice la satisfacción, de forma acumulada y a largo plazo para un entorno desconocido con todas sus características[88]. RL es el método por el cual se aprende como mapear situaciones a las acciones a fin de maximizar el resultado o una recompensa subyacente[89]. Este método afronta problemas que se centran en la interacción entre un agente activo que toma decisiones y su entorno, dentro del cual el agente trata de lograr un objetivo a pesar de la incertidumbre sobre su entorno. Mas allá de sus dos componentes base, el agente y el entorno, hay ciertos elementos que definen el proceso de interacción de los mismos que son los siguientes:

- Una recompensa, la cual comúnmente trata de maximizarse o minimizarse en torno a un objetivo.
- Un estado, el cual refleja la situación actual del entorno, así como toda información que pueda ser relevante para el agente.
- Un conjunto de acciones, mediante el cual el agente puede interactuar con el entorno generando ciertos efectos en el mismo.

Estos elementos interactúan en el proceso de aprendizaje de refuerzo comúnmente de la forma que se muestra en la Figura 2.8. Hay un entorno que define qué acciones podrían realizarse en cada momento, de las cuales el agente selecciona una como A_t , y luego el entorno devolverá tanto el estado en cada momento como S_t , como la recompensa R_t basada en el objetivo.

2.4.1. Algoritmos de RL

Para determinar la siguiente acción que debe realizar el agente suele emplearse funciones que mapean el estado del entorno con valores de importancia para seleccionar la acción preferente. La función que mapea acción-valor determina un peso para cada acción posible cuando el agente se encuentre en un estado particular, y con este peso se podría determinar la siguiente acción

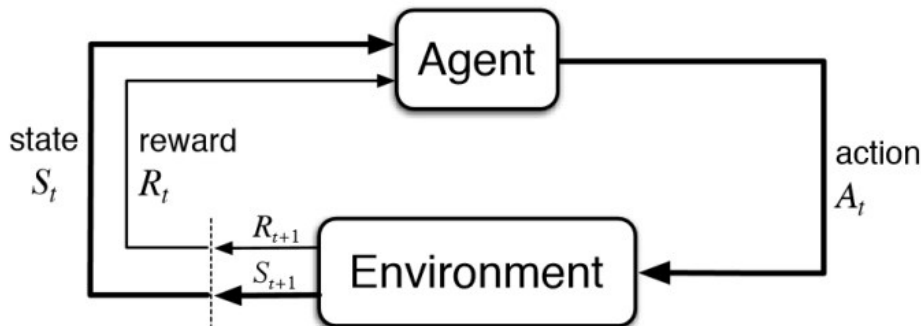


Figura 2.8: Proceso de Reinforcement learning.

óptima. A continuación, se presentan algunos de los algoritmos empleados por los agentes RL.

Q-Learning

Q-learning es una forma de aprendizaje de refuerzo sin modelos[90], la cual puede ser vista como un método de programación dinámica asincrónica (DP). Proporciona a los agentes la capacidad de aprender a actuar de forma óptima en los dominios Markovianos experimentando las consecuencias de las acciones, sin necesidad de que se construyan mapas de los dominios. El aprendizaje procede de manera similar al método de Sutton de diferencias temporales (DT)[91]: un agente intenta una acción en un estado particular, y evalúa sus consecuencias en términos de la recompensa o penalidad inmediata que recibe y su estimación del valor del estado al que se lleva. Probando repetidamente todas las acciones en todos los estados, aprende evaluando la recompensa a largo plazo.

Q-learning utiliza una aproximación de función para estimar la función de valor de la acción a en el estado s , $Q(s, a, \Theta) \simeq Q^*(s, a)$ basada en la ecuación de Bellman como una actualización iterativa (Función 2.1)

$$Q^*(s, a) = \sum_{s'} [r + \gamma \max_{a'} Q^*(s', a') | s, a] \quad (2.1)$$

Deep Q Network

Aunque Q-learning es un algoritmo muy poderoso, su principal debilidad es la falta de generalidad. Si se ve el Q-learning como la actualización de los números en una matriz bidimensional ($A * S$), se asemeja, de hecho, a la programación dinámica. Esto indica que para los estados que el agente de

Q-learning no ha visto antes, no tiene ni idea de qué acción tomar. En otras palabras, el agente de Q-learning no tiene la capacidad de estimar el valor de los estados no vistos. Para tratar este problema, Deep Q Network (DQN) descarta la matriz bidimensional remplazándola con una Red Neural.

DQN aprovecha una Red Neural para estimar la función del valor Q. La entrada de la red es el estado actual del entorno, mientras que la salida es el correspondiente valor Q para cada una de las acciones.

Otras dos técnicas son también esenciales para el entrenamiento de las DQN:

Repetición de la experiencia: Dado que las muestras de entrenamiento en la configuración típica de RL están altamente correlacionadas, y son menos eficientes en cuanto a los datos, esto llevará a una mayor convergencia para la red. Una forma de resolver el problema de la distribución de muestras es adoptar la repetición de la experiencia. Esencialmente, se almacenan las transiciones de la muestra, que luego se seleccionarán al azar de la “reserva de transición” para actualizar los conocimientos.

Red de objetivos separada: La Red de Objetivo Q tiene la misma estructura que la que estima el valor. Cada paso C , la red de destino se reajusta a la otra. Por lo tanto, la fluctuación se vuelve menos severa, resultando en entrenamientos más estables.

Antes de la DQN, se demostró que los algoritmos de RL eran inestables o incluso divergentes cuando se aproximaban al valor de las acciones con una función no lineal como las redes neuronales[92]. DQN hizo varias contribuciones importantes:

1. Estabilizar el entrenamiento de la aproximación de la función de valor de acción con redes neuronales profundas (CNN) utilizando la repetición de la experiencia [93] y la red de objetivos.
2. Diseñar un enfoque de RL de extremo a extremo, con sólo los píxeles y la puntuación del juego como entradas, de modo que sólo se requiere un conocimiento mínimo del dominio.
3. Entrenar una red flexible con el mismo algoritmo, arquitectura de red e hiper-parámetros para desempeñarse bien en muchas tareas diferentes, es decir, por ejemplo 49 juegos Atari [94], y superar los algoritmos anteriores y desempeñarse de manera comparable a un profesional humano.

El agente utilizado en este trabajo se basó en los algoritmos DQN[95] que son capaces de combinar el aprendizaje de refuerzo con una red neuronal. Algunas de las implementaciones o extensiones se listan a continuación:

- Van Hasselt et al.[96] propusieron el Doble DQN o Double DQN (D-DQN) para abordar el problema de la sobre estimación en el Q-learning.
- Schaul et al.[97] propusieron priorizar la repetición de la experiencia, para que las transiciones importantes de la experiencia puedan ser repetidas con mayor frecuencia, para aprender más eficientemente.
- Wang et al.[98] propusieron la arquitectura de redes competitivas (dueling network) para estimar la función de valor de estado $V(s)$ y la función de ventaja asociada $A(s, a)$, y luego combinarlas para estimar la función de valor de acción $Q(s, a)$, para converger más rápidamente que el Q-learning. En DQN, a una capa CNN le sigue una capa totalmente conectada (FC). En la arquitectura de duelo, una capa CNN va seguida de dos corrientes de capas FC, una para estimar la función de valor y otra para estimar la función de ventaja por separado; luego las dos corrientes se combinan para estimar la función de valor de acción. La arquitectura de duelo implementada con D-DQN y la repetición de la experiencia priorizada mejoró el trabajo anterior, DQN y D-DQN con repetición de la experiencia priorizada.
- Ansel et al.[99] propusieron reducir la variabilidad e inestabilidad en un promedio de las estimaciones de los valores Q anteriores.
- He et al.[100] propusieron acelerar la DQN mediante el ajuste de la optimización, un enfoque de optimización restringido, para propagar la recompensa más rápidamente, y mejorar la precisión sobre la DQN.
- O'Donoghue et al.[101] propusieron una política de gradiente y Q-learning (PGQ).
- Oh et al.[102] propusieron la predicción de vídeo espacio-temporal condicionada a acciones y marcos de vídeo previos con redes neuronales profundas en los juegos de Atari.
- Osband et al.[103] diseñaron una mejor estrategia de exploración para mejorar el DQN.
- Hester et al.[104] propusieron aprender de la demostración con nuevas funciones de pérdida.

2.4.2. Entornos de RL

La contraparte del agente es el entorno de aprendizaje, el cual actúa como una caja negra para el agente. Este entorno desconocido por el agente que actúa como fuente de incertidumbre, también define el área de conocimiento del agente.

El conocimiento del agente depende completamente de lo que el entorno

exponga en forma de estado. Este estado representa la situación interna del entorno y puede variar desde el conjunto de píxeles que observaría un jugador de un videojuego hasta variables definidas limitadas en tuplas de las cuales el agente no conoce necesariamente todas las variaciones.

Junto con el estado, el entorno define una fuente más de conocimiento para el agente que consiste en la recompensa. La misma tiene implícita información tanto sobre qué estado es favorable, así como que acciones son positivas en contexto de los mismos. Permitiendo que el agente al interpretar correctamente la recompensa pueda aprender a tomar las acciones correctas en el estado oportuno.

Estas posibles acciones de parte del agente también se encuentran limitadas por el entorno. Dependerá del entorno definir un espacio de acción para que el agente pueda adecuarse y conocer las acciones que puede realizar para posteriormente comprender el impacto de las mismas.

Finalmente, si bien los entornos pueden variar desde entornos de texto a representaciones con entornos gráficos, con múltiples tipos de acciones y recompensas es necesario poder comparar a los agentes de forma desacoplada de los entornos. Para esto se requiere poder generalizar la estructura y el funcionamiento del entorno. Gym de OpenAI ofrece una propuesta de estandarización de los entornos definiendo interfaces y estructuras sobre las cuales se pueden construir entornos compatibles con múltiples implementaciones de agentes facilitando así la comparación y selección del agente indicado para la problemática de un entorno particular[105].

2.4.3. RL para configurar una ludificación

El trabajo de Lieder et. al [106] busca la manera de obtener una ludificación óptima mediante un marco matemático. En este trabajo introducen un enfoque de ludificación para problemas de decisión secuenciales, calculan pseudo-recompensas óptimas para guiar a los usuarios en la actividad por lo que se requiere un conocimiento perfecto del entorno de decisión o la posibilidad de aproximarlos. En este trabajo, tratamos de lograr una “ludificación óptima”, pero a partir de una ludificación existente y sin la limitación de haber sido aplicada en el contexto de un problema de decisión secuencial, ya que no es la forma de trabajo colaborativo en las Comunidades de Construcción de Conocimiento.

Como se trata de una ludificación existente, la forma de lograr una ludificación óptima es configurando la ludificación para ajustarla a nuestro propósito. Estos enfoques abordan el tema de la problemática de la configuración de sistemas que típicamente implica un aprendizaje a partir del análisis de ejecuciones reales o datos históricos, para luego modelar aspectos específicos de los sistemas, y finalmente adaptarse a las condiciones reales en

base a los requerimientos [16]. Una forma común de ajustar las configuraciones es manualmente por ingenieros de rendimiento, requiriendo varias horas de trabajo y análisis [107]. Se propusieron otros métodos que consumen menos tiempo y son más precisos para encontrar configuraciones. Algunos de ellos se enumeran a continuación categorizados en base a los enfoques de sus métodos:

- Basado en Reglas (Rule-based): Un enfoque heurístico de caja gris en el que los expertos del dominio trabajan con los usuarios para establecer un conjunto de reglas que se utilizan para recomendar configuraciones adecuadas. Un enfoque podría ser la minería de datos multi-relacional, o Multirelational Data Mining (MRDM), que es útil para descubrir patrones, pero requiere la recopilación de los metadatos de la base de datos del sistema a configurar que describan la mejor estrategia de análisis y transformación de la base de datos en formatos MRDM. Con este método, es fácil descubrir un gran número de patrones en una base de datos donde la mayoría de estos patrones son realmente obvios, redundantes e irrelevantes o poco interesantes para el usuario[108]. Bilal et al.[109] presentan un enfoque en el que los usuarios proporcionan una clasificación de los parámetros de acuerdo con un nivel de prioridad y especifican si un aumento del valor del parámetro tiene un impacto positivo general en la latencia y el rendimiento. Este enfoque favorece la rápida búsqueda de una configuración adecuada a expensas de ser óptimo.
- Basados en modelos (Model-based): Un enfoque que se ocupa de realizar experimentos en un subconjunto del total de configuraciones para observar su rendimiento. Los resultados se utilizan durante el entrenamiento de un modelo estadístico para encontrar buenas configuraciones. Fischer et al. [110] y Trotter et al. [111] presentan un algoritmo de autoajuste utilizando la Optimización Bayesiana o Bayesian Optimization (BO)[112] para lograr un alto rendimiento. Jamshidi et al. [113] del mismo modo proponen un enfoque utilizando BO, sin embargo, optimiza la latencia y aprovecha los Procesos Gaussianos [114] para estimar continuamente la media y el intervalo de confianza de una variable en configuraciones aún por explorar.
- Basados en búsquedas (Search-based): Para este enfoque, se selecciona una configuración inicial después de la cual los experimentos se llevan a cabo de forma secuencial. Cada iteración utiliza los resultados de la anterior para ajustar un modelo estadístico que se utiliza para seleccionar la siguiente configuración. Los algoritmos de búsqueda evolutiva se adoptan típicamente para el ajuste automático de parámetros. Trotter et al. [111] proponen un método que utiliza algoritmos genéticos o Genetic Algorithms (GA) para optimizar el rendimiento. Además, en un

artículo posterior Trotter et al. [115] utilizan GA para optimizar el rendimiento usando clasificadores SVM para refinar aún más la búsqueda del espacio de configuración. Bilal et al. [109] proponen un algoritmo escalonado basado en el Latin Hypercube Sampling[116] teniendo en cuenta tanto la métrica de latencia como la de rendimiento.

- Basados en aprendizaje (Learning-based): Un enfoque en el que se utilizan técnicas de aprendizaje en línea(online), como el aprendizaje por refuerzo o Reinforcement Learning (RL), para encontrar la configuración óptima reaccionando a una retroalimentación, es decir, a las métricas, de un *framework* de procesamiento con flujo distribuido en tiempo de ejecución es presentado en [117]. Este enfoque puede combinarse con técnicas de aprendizaje fuera de línea(offline) para acelerar la convergencia [12].

Normalmente, los anteriores trabajos se centran en un número limitado de configuraciones dentro del total disponible, y para realizar esta selección suelen requerirse conocimientos de cuáles son las variables más significativas en el impacto de la configuración. Además, esta limitación suele extenderse también al momento de tener en cuenta las dependencias o el impacto generado de la correlación entre las configuraciones. Si bien estos trabajos demuestran ser efectivos, en la práctica existen numerosos puntos de configuración con muchas dependencias entre ellos. Además, no en todos los contextos se posee conocimiento exacto de como cada configuración impactara en el rendimiento de un sistema como para realizar una selección de las que posean mayor impacto o relevancia.

El enfoque presentado en este trabajo, en cambio, permite la utilización de todos los parámetros de configuración como entrada para un RL, el cual se ocupará de establecer las relaciones y las prioridades entre los distintos parámetros más allá de la incertidumbre introducida por el desconocimiento del impacto real de los mismos. Además, este enfoque se ajusta mejor a nuestro contexto en el que no hay pares correctos de entrada/salida que definan una configuración correcta para una ludificación, pero podría definirse una política de satisfacción para una CCC que actuaría como nuestro entorno desconocido.

Hay trabajos que aplicaron el aprendizaje automático al contexto de ludificación para mejorar el resultado de la estrategia de ludificación. Si bien, las aplicaciones habituales del aprendizaje automático son los juegos[118], mediante la aplicación de diferentes enfoques como aprendizaje supervisado o de refuerzo, hay un incremento en la aplicación del aprendizaje automático para mejorar las ludificaciones. Fadhil et. al. [119] agregan una capa de ludificación a una capa de aprendizaje automático conversacional que promueve el uso de la última para potenciar el conocimiento de los niños sobre la dieta y el estilo de vida saludables.

Esta estrategia de ludificar el contexto para alentar la participación también podría aplicarse al contexto colaborativo como en Knutas et. al [120]. En este contexto, generan una nueva estrategia de ludificación para un entorno colaborativo respaldado por computadora (CSCE por sus siglas en inglés) con la característica particular para obtener una experiencia de usuario generada automáticamente para cada usuario. Wikipedia como CCC también es una CSCE, pero en este trabajo no hay una nueva estrategia de ludificación, en cambio se utiliza una estrategia de ludificación existente la cual se ajusta para la comunidad, lo que trae la otra diferencia principal de este trabajo, que es la adaptación a nivel comunitario y no a nivel de usuario, por lo cual, se modifica la experiencia para todos los usuarios y no la experiencia particular de cada uno.

Capítulo 3

Análisis de Patrones en CCC

En este capítulo se introduce un análisis de los datos históricos las CCC, tomando como ejemplo a Wikipedia, con el fin de validar la existencia de patrones de comportamiento de dentro de los datos históricos de las CCC. Además, sirve para introducir los elementos que sirven de entrada a los procesos de ludificación de las CCC. Para ello, primero se presenta la metodología de extracción, con las tecnologías utilizadas como referencia. Luego se presenta la fuente de información utilizada para el análisis y los elementos a extraer de la misma. Finalmente, se presenta un análisis sobre dichos elementos aplicando un enfoque de aprendizaje no supervisado y sus conclusiones.

3.1. Procesamiento del historial de revisiones

Aunque Wikipedia cuenta con diferentes herramientas para contribuir, toda contribución se almacena finalmente en el historial de revisiones como una versión completa del artículo, por lo cual la información que se obtiene del mismo es el estado final de la revisión y no de los cambios que se realizaron.

La primera etapa para el análisis del historial de revisiones, la cual se presenta en la Sección 3.1.1, consiste en obtener todos los artículos considerados relevantes con su historial de revisiones. La siguiente etapa consiste en generar una lista cronológica de todas las diferencias de los historiales de revisiones. Dichas diferencias luego son analizadas para detectar acciones de edición en Wikipedia como se explica en la Sección 3.1.2. Además, estas acciones posteriormente se podrán mapear a acciones de los jugadores en las simulaciones de la estrategia de ludificación. La Figura 3.1 muestra este proceso, comenzando por Wikipedia que contiene los distintos tipos de artículos que se extraerán de la misma mediante el uso de su API. Esta

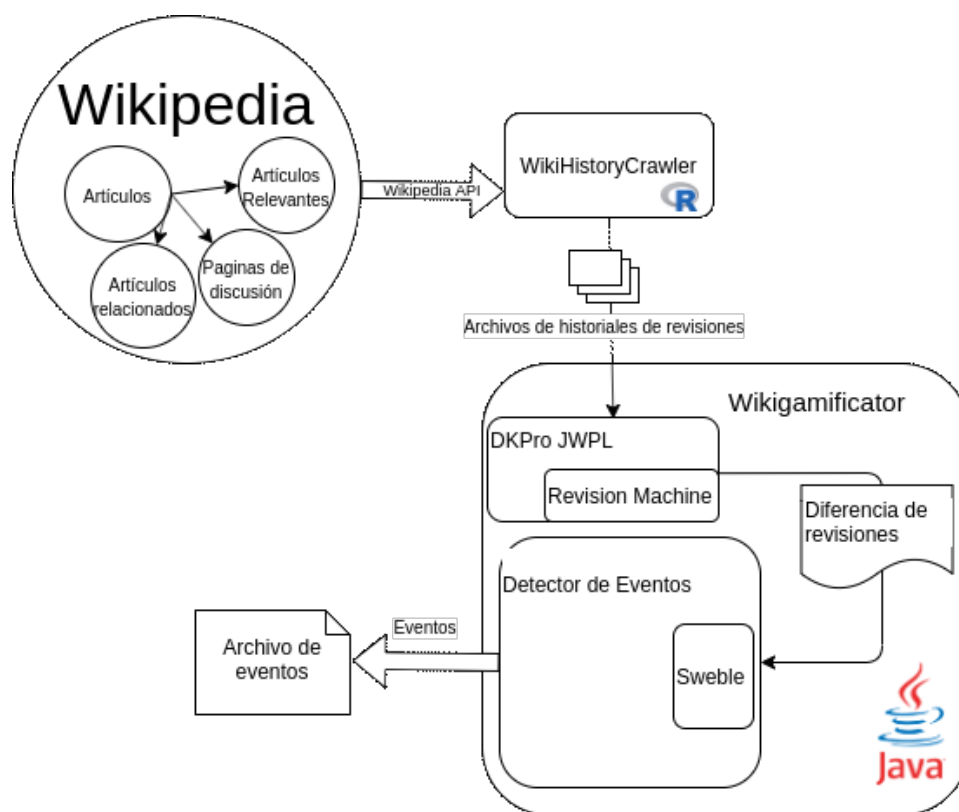


Figura 3.1: Procesamiento del historial de revisiones.

interacción será realizada utilizando el script llamado `WikiHistoryCrawler` el cual obtendrá todos los archivos de revisiones de los artículos que se le indiquen. Estos archivos serán tomados por el `WikiGamificator` el cual generara finalmente un archivo con los eventos sobre los que se realizara al búsqueda de patrones. Las subsecciones siguientes describen este flujo de datos y las herramientas involucradas¹ en mayor detalle.

3.1.1. Extracción de Wikipedia

Cada artículo de Wikipedia tiene un historial de revisión que incluye todas las revisiones anteriores ordenadas desde la más reciente hasta la más antigua. Cuando un editor modifica un artículo y lo guarda, se crea una nueva revisión y se coloca en la primera posición del historial de revisiones.

El historial de revisiones del artículo está disponible en varios formatos, este trabajo utiliza el formato XML. En el Listado 3.1.1 se observa un frag-

¹Las herramientas utilizadas pueden encontrarse en: <https://github.com/jonx18/WikiCrawler>

mento del historial de revisiones, en él se encuentran el título del artículo (title), la identificación del artículo (id), el espacio de nombres (ns) para describir el artículo el cual adopta diferentes valores según si es un artículo común, una categoría, una página de discusión u otros, y las revisiones (revision). Cada revisión en el historial tiene un identificador (id), una marca de tiempo (timestamp), la persona que la hizo (contributor), un comentario (comment), y por último, el contenido completo del artículo en el momento en que se creó la revisión actual (text).

```

<page>
  <title>Pope</title>
  <ns>0</ns>
  <id>23056</id>
  <revision>
    <id>2806055</id>
    ...
  </revision>
  <revision>
    <id>2806196</id>
    <parentid>2806055</parentid>
    <timestamp>2004-03-17T16:46:39Z</timestamp>
    <contributor>
      <username>Barbara Shack</username>
      <id>40231</id>
    </contributor>
    <text>{{About|the leader of the
      Catholic Church|the popes of
      other churches, and other uses}}...
    </text>
  </revision>
  <revision>
    ...
  </revision>
</page>

```

Listado 5.1.1: Ejemplo de artículo en archivo XML.

La tarea de extracción, obtiene de Wikipedia todo el historial de revisiones de los artículos que se le indique considerándolos como artículos relevantes de extracción. Para procesar las páginas relevantes, se desarrolló un script en el lenguaje R que interactúa con la API de Wikipedia² para descargar los datos de la Web. El mismo puede observarse en la Figura 3.1

²https://www.mediawiki.org/wiki/API:Main_page

```

==History==
{{Main article|History of the papacy}},
===Title and etymology===
The word "'pope'" derives from [[Greek language|Greek]]
πάππας meaning "father". In the early centuries of
[[Christianity]], this title was applied, especially in the
east, to all bishops<ref name=ODCC:Pope></ref> and other
senior clergy, and later became reserved in the west to the
Bishop of Rome, a reservation made official only in the 11th
century.<ref></ref><ref></ref><ref></ref><ref></ref><ref></ref></ref>
The earliest record of the use of this title was in
regard to the by then deceased [[Patriarch of Alexandria]],
[[Pope Heraclas of Alexandria]] (232-248).<ref></ref> The
earliest recorded use of the title "pope" in English dates
to the mid-10th century, when it was used in reference to
[[Pope Vitalian]], in an Old English translation of
[[Bedē]]'s ''[[Historia ecclesiastica gentis
Anglorum]]''<ref></ref>

```

Figura 3.2: Extracto del artículo del Papa en texto plano.

como WikiHistoryCrawler. Este script permite la extracción de los historiales de revisiones de los artículos relevantes, así como también permite buscarlos por categorías o subcategorías. También permite obtener páginas de discusión y otros artículos con un enlace a estos artículos relevantes. Con el listado de artículos a descargar generado, el script obtiene el historial de revisiones y lo guarda en un archivo XML para cada artículo.

3.1.2. Detección de actividades

El contenido de la revisión descrita dentro del XML con la etiqueta `text` es un texto enriquecido con elementos de marcado (markups) de Wikipedia o wiki-markups. Por ejemplo, la Figura 3.2 muestra la versión en texto plano con wiki-markups para el artículo del Papa de la Wikipedia en lenguaje inglés ³. En la Figura, se puede ver el texto `==History==` que representa un encabezado de nivel 2 con el título *History* (la Figura 3.3 muestra la vista del usuario). El wiki-marukup que define este encabezado es conocido como `heading2`.

A partir del historial de revisiones obtenido en la tarea anterior, el `Wikigamificator` genera la lista cronológica con todas las acciones de edición en Wikipedia. Primero, utilizando la biblioteca DKPro JWPL [121]

³<https://en.wikipedia.org/wiki/Pope> accedido el 28 de diciembre de 2016

History

Main article: History of the papacy

Title and etymology

The word *pope* derives from Greek πάππας meaning "father". In the early centuries of Christianity, this title was applied, especially in the east, to all bishops^[18] and other senior clergy, and later became reserved in the west to the Bishop of Rome, a reservation made official only in the 11th century.^{[19][20][21][22][23]} The earliest record of the use of this title was in regard to the by then deceased Patriarch of Alexandria, Pope Heraclas of Alexandria (232–248).^[24] The earliest recorded use of the title "pope" in English dates to the mid-10th century, when it was used in reference to Pope Vitalian in an Old English translation of Bede's *Historia ecclesiastica gentis Anglorum*.^[25]

Figura 3.3: Extracto del artículo del Papa en Wikipedia.

Nombre de Referencia	Wiki-markup	Apertura	Cierre	Descripción
nowiki	Nowiki	<nowiki>	</nowiki>	Previene la aplicación de wiki-markups.
big	Big text	<big>	</big>	Incrementa el tamaño del texto
small	Small text	<small>	</small>	Decrementa el tamaño del texto.
sup	Superscripts	[]	Crea un superíndice.
sub	Subscripts	_		Crea un subíndice.
s	Strike-through	<s>	</s>	Tacha el texto contenido.
blockquote	Blockquote	<blockquote>	</blockquote>	Crea un bloque de cita.
includeonly	Includeonly	<includeonly>	</includeonly>	Es utilizado para la definición de plantillas.
reference	Reference	<ref>	</ref>	Crea una referencia.
heading2	Heading 2	==	==	Crea un título tipo 2.
heading3	Heading 3	===	===	Crea un título tipo 3.
heading4	Heading 4	====	====	Crea un título tipo 4.
heading5	Heading 5	=====	=====	Crea un título tipo 5.
italic	Italic text	Dos apóstrofes ^{''}	Dos apóstrofes ^{''}	Texto en cursiva.
blod	Bold text	Tres apóstrofes ^{'''}	Tres apóstrofes ^{'''}	Texto en negrita.
italicblod	Italic and bold text	Cinco apóstrofes ^{'''''}	Cinco apóstrofes ^{'''''}	Texto en negrita y cursiva.
external	External links			Conecta con sitios web fuera de Wikipedia.
internal	Interwiki link			Conecta a páginas propias de Wikipedia.
numberedelement	Ordered Element	#	Not need	Crea elementos ordenados.
bulletedelement	Unordered Element	*	Not need	Crea elementos sin orden.
redirect	Redirects	#REDIRECT [[]]	Redirecciona a un artículo con otro nombres.
indent2	Indent text 2	Two colon ::	Not need	Crea sangría de tipo 2.
indent1	Indent text 1	Colon :	Not need	Crea sangría de tipo 1.
infobox	InfoBox	{{Infobox	}}	Crea un InfoBox.
wikitable	WikiTable	{{class=wikitable	}}	Crea una tabla.
cite	Cite	{{cite	}}	Crea una cita.

Tabla 3.1: Wiki-markups analizados.

que permiten la carga y el procesamiento de forma eficiente de historiales de revisión de Wikipedia, se detectan los cambios entre revisiones para saber qué cambios realizó un editor. Luego esos cambios se analizaron con la librería Sweble [122] la cual permitió la estructuración jerárquica del texto del artículo dividiéndolo en nodos delimitados por el uso del wiki-markup. A partir del conjunto de diferencias y la estructuración del contenido se pudieron identificar los cambios de los wiki-markups. Este proceso de identificación de cambios en el wiki-markup se muestra en la Figura 3.1 como **Event Detector**.

La Tabla 3.1 presenta una lista completa de los wiki-markups analizados en este capítulo. La primera columna detalla el nombre de referencia, la segunda columna detalla el nombre real del markup, la tercera y cuarta columna detallan la etiqueta de inicio y final de cada wiki-markup respectivamente, y su descripción aparece en la última columna. Llamamos a la lista resultante **Conjunto de eventos**. Este proceso conocido como **Event Detector** es el que posteriormente se adaptara para que las acciones de los editores se mapeen a acciones de los jugadores para la simulación de la estrategia de ludificación.

3.2. Abstracción de eventos en Wikipedia

El estudio de la proveniencia como se mencionó anteriormente, puede realizarse mediante el análisis del historial de revisiones para comprender la evolución y el comportamiento de un elemento a lo largo del tiempo. En el contexto de Wikipedia, la evolución del contenido de cada artículo, como se mencionó anteriormente, se describe en su historial de revisiones. Con esta información, es posible, por ejemplo, reconstruir el proceso de escritura desde el principio hasta el final [46]. También es posible detectar y clasificar el comportamiento del usuario [47], estudiar la evolución de la calidad de un artículo [48] o detectar los patrones de uso de las herramientas proporcionadas para realizar el proceso de creación de conocimiento, como páginas de discusión, tablero de comunicación o elementos de formato.

Dado que el contenido de las revisiones se presenta en texto plano se busca identificar si los elementos de marcado indican que acción realizaron los editores para confirmar que esta información sirve para la posterior simulación de la estrategia de ludificación. Con este fin, se evaluaron las siguientes preguntas:

1. ¿Es posible agrupar revisiones por el análisis de sus elementos de marcado? Sería relevante reconocer la presencia de patrones en el uso de elementos de marcado entre las revisiones. Por ejemplo, revisiones de diferentes artículos que usan el mismo conjunto de elementos de marcado.

2. De los cambios de elementos de marcado entre las revisiones (eliminaciones y adiciones), a los cuales llamaremos “evolución de revisiones”, se desea analizar y caracterizar esta evolución de revisiones a fin de responder, ¿Es posible agrupar diferentes patrones del uso de estos elementos en la evolución de revisiones? De hecho, sería deseable realizar este análisis con diferentes niveles de granularidad, por ejemplo, distinguiendo entre niveles bajos, medios o altos de adiciones o eliminaciones.
3. ¿Qué y cómo es la co-ocurrencia de esos cambios? Por ejemplo, hay dos elementos de marcado llamados X e Y , ¿Con que frecuencia se aplican X e Y juntas? ¿Qué tan posible es que dado X ocurra Y ? ¿La aplicación de X e Y en conjunto es un patrón o simplemente una coincidencia? Para responder estas preguntas se evaluarán las métricas de soporte, confianza o *lift* de estos conjuntos (X, Y) , donde cada métrica está destinada a responder a una pregunta en el orden respectivo que fueron formuladas.

A continuación se desarrolla un análisis sobre los elementos que definen estructura en las revisiones los cuales se obtienen a partir del lenguaje de marcado de Wikipedia, sobre el cual se introducirá un enfoque de aprendizaje no supervisado para aplicar *cluster analysis* (análisis de grupos) y minería de reglas de asociación para analizar contenido de los artículos y su evolución. El enfoque está estructurado en varios pasos. En primer lugar, el *cluster analysis* se aplica para estudiar el uso de estos elementos en una revisión y, entre revisiones. En segundo lugar, las reglas de minería de asociación se utilizan para realizar un análisis de correlación entre revisiones. Finalmente, una combinación de ambos análisis tiene el propósito de tener un mejor detalle en los patrones de uso de estos elementos de marcado en las ediciones de Wikipedia.

3.3. Resultados

El análisis se realizó sobre 126,000 de artículos⁴ de la versión en inglés de Wikipedia, que contienen más de 2,000,000 de revisiones a la fecha del 28 de diciembre de 2016. Como resultado, se obtuvo una clasificación para las revisiones y se detectó la existencia de patrones en la utilización de wiki-markups. Esos resultados pueden ser útiles para distinguir las intenciones del usuario, así como plantear la utilidad de los wiki-markups caracterizar las acciones de los usuarios en una futura simulación de la estrategia de ludificación. En esta sección se introducen las herramientas, métricas y los

⁴Lista de todos los artículos usados en <https://github.com/jonx18/Patterns-of-Markup-use-in-Wikipedias-Context/blob/master/articlelist.txt>

análisis con sus resultados divididos en 5 casos de estudio incrementales.

3.3.1. Herramientas

Para realizar el proceso de aprendizaje no supervisado se utilizó en el entorno de software R para la computación estadística⁵. El aprendizaje no supervisado se utiliza durante la etapa de análisis exploratorio para buscar patrones no descubiertos en conjuntos de datos no etiquetados. Para responder a las preguntas sobre la existencia de patrones de uso de wiki-markups, se utilizó el algoritmo K-means para realizar *cluster analysis*; fue seleccionado porque es rápido con grandes conjuntos de datos y permite una interpretación directa.[123].

Para responder a las preguntas relacionadas con las correlaciones de wiki-markups, se utilizaron algoritmos de minería de reglas de asociación. Más específicamente Eclat y Apriori para obtener métricas de soporte, confianza y el indicador *lift* para cada regla o coocurrencia. Esos algoritmos fueron seleccionados porque ambos son bien conocidos y están completamente documentados permitiendo una minería de reglas de asociación directa para un conjunto de elementos frecuentes.

Como ambos algoritmos están diseñados para trabajar con transacciones, en esta etapa del trabajo una transacción será considerada igual a una revisión. Dado que los datos sobre la aplicación/utilización o la supresión/eliminación de wiki-markups pueden ser variadas en cantidad por cada revisión, se ha realizado un proceso de agregación utilizando diferentes intervalos de valores que se especifican para cada transacción utilizada. Finalmente, si bien esos algoritmos son conocidos por trabajar sin conservación de estados, en este trabajo se estudió la evolución del contenido del artículo utilizando las diferencias entre las revisiones como se verá en los casos de estudio.

3.3.2. Metrics

Los algoritmos mencionados anteriormente utilizan algunas métricas en sus aplicaciones. K-means por ejemplo requiere una asignación del número k de *clusters* buscados. Por lo cual se utilizó *Calinski-Harabasz Index* [124] y *Average silhouette width* [125] que son métricas utilizadas para obtener un número de *clusters* recomendado. También se empleó el coeficiente de Jaccard para evaluar la estabilidad de los *clusters*.

El coeficiente de Jaccard da una medida de la similitud entre dos *clusters*, iterando aplicando *clustering* y el coeficiente de Jaccard se puede calcular la estabilidad de cada *cluster*. La estabilidad de cada *cluster* está representada por el valor medio de su coeficiente de Jaccard en todas las iteraciones. [123]

⁵Todo el código y el conjunto de datos utilizado se encuentran en: <https://github.com/jonx18/Patterns-of-Markup-use-in-Wikipedias-Context>

Numero de artículos	CH	ASW
3	17	11
4	18	18
5	20	20

Tabla 3.2: Clusters recomendados.

Eclat y Apriori en cambio, computan el soporte, la confianza y el indicador *lift*. El soporte de un elemento (un conjunto de elementos co-ocurrentes en Eclat y una regla en Apriori) es el número de transacciones que contienen el elemento dividido por el número total de transacciones. En Apriori, las reglas son como “si X , entonces Y ”. Esto significa que cada vez que se ve el conjunto de elementos X en una transacción, también se espera ver Y . La confianza puede representarse como $\text{soporte}(\text{unión}(X, Y)) / \text{soporte}(X)$, donde la *unión*(X, Y) se está refiriendo a la regla que contiene tanto X como Y . Por lo tanto, la confianza de la regla representa con qué frecuencia aparece Y cuando esta X . Por último, *lift* compara la probabilidad de que un patrón sea observado con la probabilidad de que se observe ese patrón sólo por casualidad. El *lift* de una regla viene dado por el $\text{soporte}(\text{unión}(X, Y)) / (\text{soporte}(X) * \text{soporte}(Y))$. Si el *lift* es 1, entonces X e Y son independientes.

3.3.3. 1er caso: Análisis de Grupos de Revisiones

Este caso trató de describir cómo se agrupan las revisiones por sus wiki-markups usando K-means para hacer el agrupamiento. Como el cálculo de clusters recomendado toma un tiempo excesivamente largo con el conjunto de datos completo, la búsqueda fue evaluada mediante una estrategia iterativa. Esta estrategia consistió en correr K-means sobre las revisiones de un número P de artículos, y finalmente, aumentar P en uno.

La Tabla 3.2 muestra los resultados de tres iteraciones. La primera columna muestra el número de artículos utilizados, y la segunda y tercera columna muestran el número de *clusters* recomendados por el criterio *Índice de Calinski-Harabasz* (CH) y *Average silhouette width* (ASW), respectivamente. En la primera fila, las recomendaciones fueron 17 y 11, pero en la segunda fila, esas recomendaciones comenzaron a ser mayores con valores de 18 en ambos casos. Finalmente, en la tercera fila, los valores se incrementan más.

El incremento constante del número de *clusters* recomendados mostró que los artículos creaban *clusters* para sí mismos. Esto se debió a que los wiki-markups entre las revisiones de un mismo artículo tiende a ser muy similares.

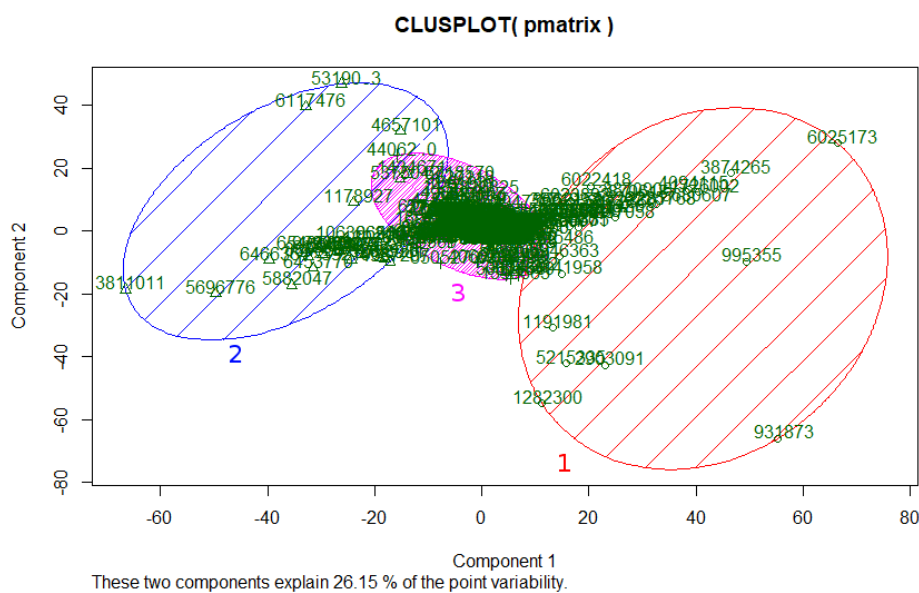


Figura 3.4: K-means clusters.

3.3.4. 2do caso: Análisis de Evolución de Revisiones

En este caso, se buscaron *clusters* en la evolución de las revisiones. Se utilizaron los cambios cuantitativos de wiki-markups entre las revisiones de los mismos artículos. Por ejemplo, si entre dos revisiones del mismo artículo se añadió un encabezado, la adición se mantiene como información de la revisión para calcular la agrupación con K-means. K-means se ejecutó con 3 clusters por recomendación de las métricas CH y ASW. Los *clusters* están representados en la Figura 3.4.

El *cluster 3* (en el centro de la figura) es el que tiene más elementos y se agrupan de manera más cohesiva que los otros dos cúmulos. Además, los otros dos cúmulos están en posiciones opuestas donde ambos tienen más elementos dispersos.

Luego se realizó una revisión manual de un subconjunto de elementos de cada *clusters* analizando el tipo de cambios que contenía. También se consideró el comentario de las revisiones relacionadas o si la misma estaba marcada como una reversión a un estado previo del artículo. A partir de este análisis y considerando las estadísticas descriptivas de cada *clusters*, se pudieron determinar tres tipos de revisiones, y cada una fue nombrada en base a sus características:

- **Ediciones vandálicas:** Esas son revisiones con un alto decremento

en wiki-markups. Además, esas revisiones representaban supresiones o sustituciones de contenido con intenciones perjudiciales para el artículo. Es representado por el cluster 1.

- **Ediciones de corrección:** Son revisiones que recuperan el contenido de las Ediciones vandálicas. En consecuencia, tiene un alto incremento de wiki-markups. Además, muchas de esas revisiones tienen comentarios que explican que son reversiones o correcciones a vandalismo. Está representado por el grupo 2.
- **Ediciones comunes:** Se trata de revisiones con cambios en un rango normal para las revisiones de artículos. Además, esas revisiones suelen tener comentarios detallados de los cambios o, a lo sumo, se marcan como reversiones propias. Está representada por el grupo 3.

3.3.5. 3er caso: Patrones de adición y eliminación

Este caso analizó la existencia de patrones de adición o eliminación de wiki-markups. Para analizar la evolución de las revisiones mediante transacciones, a la información se le realizó un proceso de agregación y etiquetado de los wiki-markups por medio de las siguientes reglas:

- Cuando se detectaba cualquier eliminación de un wiki-markup, se aplicaba la etiqueta “-Borrado”.
- Cuando se detectaba cualquier adición de un wiki-markup, se aplicaba la etiqueta “+Añadido”.
- Si no se detectaban cambios, el wiki-markup se borraba para evitar que los resultados fueran sesgados.

Para realizar este análisis, Eclat fue configurado con un soporte mínimo de 0.01 y un mínimo de 2 elementos. También, Apriori fue configurado con un soporte y confianza mínimos de 0.01 y un mínimo de 2 elementos en la primera parte de la regla. Esas configuraciones ayudaron a centrarse en las relaciones entre wiki-markups.

Como resultado⁶, se obtuvo la Tabla 3.3 para Eclat. La primera columna detalla la identificación de los resultados, la segunda columna muestra los wiki-markups que coocurren y el soporte está en la última columna. Cada conjunto de elementos está correlacionado con un determinado soporte, por ejemplo, el primer conjunto de elementos de la tabla representa que los wiki-markups “bloque de cita” y el indicador de inclusiones de plantillas se agregan en conjunto con un 3,39 % de soporte.

⁶Los resultados completos de las tablas se pueden ver en: <https://github.com/jonx18/Patterns-of-Markup-use-in-Wikipedias-Context/blob/master/3-Thirdcase/3-Thirdcaseresults.xlsx>

ID	Wiki-markups Co-Ocurrencias	Soporte
[1]	{blockquote=+Añadido,includeonly=+Añadido}	0.03391964
[2]	{internal=+Añadido,indent1=+Añadido}	0.02732596
[3]	{internal=+Añadido,bulletedelement=+Añadido}	0.02166337
[4]	{internal=-Borrado,indent1=-Borrado}	0.01552891
[5]	{internal=-Borrado,bulletedelement=-Borrado}	0.01432017
[6]	{blockquote=+Añadido,big=+Añadido}	0.01347192
[7]	{external=+Añadido,indent1=+Añadido}	0.01335142
[8]	{bulletedelement=+Añadido,indent1=+Añadido}	0.01320869
[...]
[20]	{italic=+Añadido,bulletedelement=+Añadido}	0.01021900

Tabla 3.3: Caso 3. Resultados de Eclat.

Id	X => Y	Soporte	Confianza	Lift
[1]	{s=+Añadido}=>{includeonly=+Añadido}	0.01112287	0.7554439	13.215852
[2]	{external=-Borrado}=>{indent1=-Borrado}	0.01038630	0.6004355	20.110261
[3]	{includeonly=+Añadido}=>{blockquote=+Añadido}	0.03391964	0.5933955	7.576177
[4]	{indent1=-Borrado}=>{internal=-Borrado}	0.01552891	0.5201064	12.117157
[5]	{includeonly=-Borrado}=>{blockquote=-Borrado}	0.01251420	0.5188473	16.603750
[6]	{external=+Añadido}=>{indent1=+Añadido}	0.01335142	0.5156500	9.520728
[7]	{indent1=+Añadido}=>{internal=+Añadido}	0.02732596	0.5045341	6.857714
[8]	{external=+Añadido}=>{bulletedelement=+Añadido}	0.01177999	0.4549593	5.379638
[...]
[20]	{blockquote=+Añadido}=>{bulletedelement=+Añadido}	0.01124470	0.1435667	1.697596

Tabla 3.4: Caso 3. Resultados de Apriori.

Los resultados de Apriori se muestran en la Tabla 3.4. En la primera columna está el identificador de la regla, en la segunda columna se presenta la regla en forma de “Si X entonces Y ”, y las columnas restantes representan las métricas de soporte, confianza y *lift* respectivamente. Las reglas de Apriori podrían leerse como “Si se añadió/eliminó algún wiki-markup, entonces se añadió/eliminó otro wiki-markup”, y la regla se evaluó con el soporte, la confianza y *lift*. Esos resultados fueron ordenados por confianza de mayor a menor. Las primeras siete reglas tenían una confianza superior al 50% y todas con *lift* superior a 1.

Los resultados mostraron que esas reglas y la coocurrencia sólo se dieron entre los mismos eventos, en otras palabras, la coocurrencia en los wiki-markups fueron adiciones o eliminaciones, pero no cruces de ellas. Como resultado, se puede afirmar la existencia de una correlación en la forma en que los wiki-markups se usaron en conjunto, pero se decidió continuar con una búsqueda más profunda en los siguientes casos.

ID	wiki-markup Co-Ocurrencias	Soporte
[1]	{blockquote=+Bajo+,includeonly=+Bajo+}	0,02150432
[2]	{internal=+Bajo+,indent1=+Bajo+}	0,01552964
[3]	{internal=+Bajo+,bulletedelement=+Bajo+}	0,01300579
[4]	{heading2=+Bajo+,includeonly=+Bajo+}	0,00800723
[5]	{external=+Bajo+,indent1=+Bajo+}	0,00736968
[6]	{blockquote=+Bajo+,big=+Bajo+}	0,00697477
[7]	{s=+Bajo+,includeonly=+Bajo+}	0,00685411
[8]	{bulletedelement=+Bajo+,indent1=+Bajo+}	0,0068472
[...]
[460]	{italic=+Bajo+,heading2=+Bajo+}	0,001000242

Tabla 3.5: Caso 4. Resultado de Eclat.

3.3.6. 4to caso: Granularidad fina de cambios

En este caso, se analizaron los cambios en la evolución de las revisiones en una granularidad más fina. La información de wiki-markups fue agregada y etiquetada según la cantidad de cambios de ese wiki-markup en la misma revisión, con tres rangos llamados Bajo, Medio y Alto. Por lo tanto, las reglas de agregación fueron:

- Cuando se detectó cualquier eliminación de wiki-markup, las etiquetas “-Bajo-”, “-Medio-”, “-Alto-” en base al rango de wiki-markups eliminados entre (0 : -2], (-2 : -10] y (-10:-Inf) respectivamente.
- Cuando se detectó cualquier aplicación de wiki-markup, las etiquetas “+Bajo+”, “+Medio+”, “+Alto+” en base al rango de wiki-markups aplicados entre (0 : 2], (2 : 10] y (10:Inf) respectivamente.
- Si no se detectaban cambios, el wiki-markup se borraba para evitar que los resultados fueran sesgados.

Para realizar este análisis, Eclat se configuró con un soporte mínimo de 0,001 y un mínimo de 2 elementos. También, Apriori fue configurado con un soporte y confianza mínimos de 0.001 y un mínimo de 2 elementos en la primera parte de la regla. Estas configuraciones ayudaron a centrarse en las relaciones entre wiki-markups y la frecuencia de sus cambios correctamente. Además, los mínimos seleccionados permiten obtener resultados más relevantes sobre el *lift*.

Como resultado⁷, se obtuvo la Tabla 3.5 para Eclat. La primera columna detalla la identificación de los resultados, la segunda columna muestra el wiki-markups que coocurren y el soporte está en la última columna. En la

⁷Los resultados completos de las tablas se pueden ver en: <https://github.com/jonx18/Patterns-of-Markup-use-in-Wikipedias-Context/blob/master/4-Fourthcase/4-Fourthcaseresults.xlsx>

Id	$X \Rightarrow Y$	Soporte	Confianza	Lift
[1]	$\{\textit{italic}=-\text{Alto}-\} \Rightarrow \{\textit{internal}=-\text{Alto}-\}$	0,0029	0,82068276	94,05664
[2]	$\{\textit{heading2}=-\text{Alto}-\} \Rightarrow \{\textit{bulletedelement}=-\text{Alto}-\}$	0,00158	0,80456758	132,92482
[3]	$\{\textit{external}=-\text{Alto}-\} \Rightarrow \{\textit{indent1}=-\text{Alto}-\}$	0,00192	0,79679373	150,12876
[4]	$\{\textit{indent1}=-\text{Alto}-\} \Rightarrow \{\textit{internal}=-\text{Alto}-\}$	0,00422	0,79543123	91,16262
[5]	$\{\textit{external}=-\text{Alto}-\} \Rightarrow \{\textit{internal}=-\text{Alto}-\}$	0,0019	0,78828856	90,34402
[6]	$\{\textit{external}=-\text{Alto}-\} \Rightarrow \{\textit{bulletedelement}=-\text{Alto}-\}$	0,00189	0,7870648	130,03314
[7]	$\{\textit{italic}=\text{+Alto+}\} \Rightarrow \{\textit{internal}=\text{+Alto+}\}$	0,00277	0,76948238	93,73189
[8]	$\{\textit{external}=\text{+Alto+}\} \Rightarrow \{\textit{internal}=\text{+Alto+}\}$	0,00179	0,75399282	91,84508
[...]
[298]	$\{\textit{reference}=\text{+Bajo+}\} \Rightarrow \{\textit{italicblod}=\text{+Bajo+}\}$	0,00104144	0,04053489	1,1357819

Tabla 3.6: Caso 4. Resultado de Apriori.

tabla se observan los elementos correlacionados con su respectivo soporte, por ejemplo, el primer elemento de la tabla representa que los wiki-markups “bloque de cita” y el indicador de inclusiones de plantillas se utilizan en conjunto en nivel bajo con 2.15 % de soporte. Los elementos obtenidos de esos resultados tuvieron menos soporte que los resultados del tercer caso.

Los resultados de Apriori se muestran en la Tabla 3.6. En la primera columna se encuentra el identificador de la regla, en la segunda columna se presenta la regla en forma de “Si X entonces Y ”, y las columnas restantes representan las métricas de soporte, confianza y *lift* respectivamente. Las reglas pueden leerse como “Si se añadiera/eliminaría algún wiki-markups en un nivel se añadiría/eliminaría otro wiki-markup en otro nivel”. De Apriori se obtuvieron 298 reglas, esas reglas tenían una mayor confianza que los resultados en el tercer caso; además, el *lift* era también alto. Por último, los resultados con mayor confianza estaban relacionados con eventos de grandes eliminaciones o adiciones, pero esas reglas tenían un soporte poco significativo.

A partir de esos resultados, se pudo confirmar la existencia de una correlación directa entre la aplicación o la eliminación en conjunto de los wiki-markup. Hubo menos soporte que en el segundo caso, pero ello se debió a que los eventos de eliminación y adición se dividieron en tres eventos más pequeños cada uno. Además de esto en los resultados de Apriori, todos los resultados fueron de nivel alto, esto ocurrió porque el límite de soporte reducido permitió revisiones de Vandalismo o Corrección y esos valores sesgaron las reglas. Este sesgo en los resultados fue corregido en la implementación del siguiente caso.

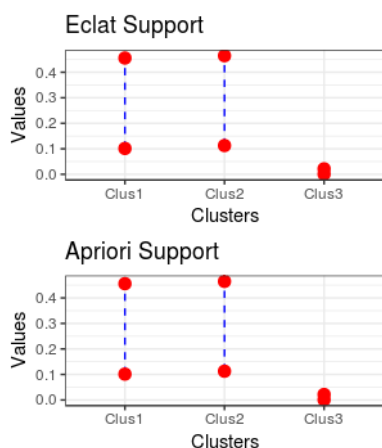


Figura 3.5: Rangos de valores en Eclat y Apriori para cada cluster.

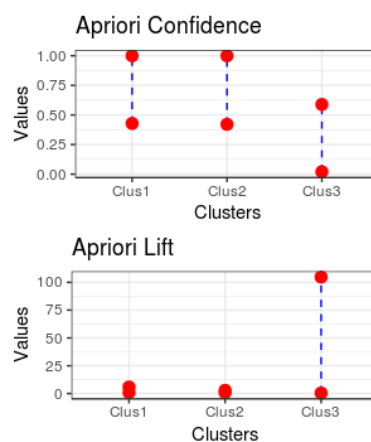


Figura 3.6: Rangos de valores en Eclat y Apriori para cada cluster.

3.3.7. 5to caso: Evolución de revisiones detalladas por grupos

En este caso se analizan las características de los *clusters* (Vandalismo, Corrección y Ediciones Comunes) encontrados en el segundo caso, pero utilizando el proceso de análisis perteneciente al cuarto caso.

Primero, para obtener una versión más refinada de los *clusters*, se hizo un proceso de *clustering* iterativo. En cada iteración, los *clusters* fueron evaluados con el coeficiente Jaccard, y se eliminaron los wiki-markup de los *clusters* con menor estabilidad. Esta iteración continuó hasta que el coeficiente de los *clusters* no aumentó más. Esos *clusters* se mantuvieron, y los elementos eliminados se asignaron a los *clusters* estabilizados que correspondían.

Después del proceso anterior, para cada *cluster* que corresponde a uno de los tres tipos de revisión descritos en el segundo caso, las revisiones se analizaron aplicando el proceso descrito en el cuarto caso. En las Figuras 3.5 y 3.6, se muestran los rangos de valores para cada métrica en Eclat y Apriori para cada *cluster*.

En cuanto al soporte, los grupos de Ediciones Vandálicas y de Corrección tuvieron el mayor soporte, y eso se debió a que esos grupos tuvieron menos transacciones y con artículos muy particulares. Por otro lado, el soporte de las Ediciones Comunes fue el más bajo, pero era de esperar porque había más variedad de wiki-markups añadidos o eliminados en diferentes magnitudes y este *cluster* era el más grande. En cuanto a la confianza, también los grupos de Ediciones Vandálicas y de Corrección tenían la mayor confianza. Esos niveles de confianza se debían a que la confianza es sensible a la frecuencia de los elementos Y , y en este caso, cuando se poseen elementos con fuerte soporte

Id	X =>Y	Soporte	Confianza	Lift
[1]	{indent2=+Bajo+}>{heading2=+Bajo+}	0,005381	0,588889	25,78025
[2]	{s=+Bajo+}>{includeonly=+Bajo+}	0,007411	0,58871	13,06034
[3]	{blockquote=+Alto+}>{includeonly=+Medio+}	0,002132	0,567568	68,17732
[4]	{s=-Medio-}>{includeonly=-Medio-}	0,001421	0,518519	62,28546
[5]	{blod=-Medio-}>{internal=-Alto-}	0,001117	0,5	104,7872
[6]	{blockquote=+Alto+}>{italicblod=+Bajo+}	0,001827	0,486486	13,7698
[7]	{includeonly=+Bajo+}>{blockquote=+Bajo+}	0,021421	0,475225	8,343972
[8]	{reference=-Medio-}>{indent1=-Medio-}	0,002843	0,451613	39,71774
[...]
[215]	{internal=-Bajo-}>{includeonly=-Bajo-}	0,001116751	0,02022059	0,4485874

Tabla 3.7: Caso 5. Resultados de Apriori para Ediciones Comunes

ID	Wiki-markups Co-ocurrencia	Soporte
[1]	{blockquote=+Bajo+,includeonly=+Bajo+}	0,021421
[2]	{internal=+Bajo+,indent1=+Bajo+}	0,016751
[3]	{internal=+Bajo+,bulletedelement=+Bajo+}	0,013096
[4]	{heading2=+Bajo+,includeonly=+Bajo+}	0,008122
[5]	{s=+Bajo+,includeonly=+Bajo+}	0,007411
[6]	{external=+Bajo+,indent1=+Bajo+}	0,00731
[7]	{reference=+Bajo+,indent1=+Bajo+}	0,00731
[8]	{bulletedelement=+Bajo+,indent1=+Bajo+}	0,007208
[...]
[251]	{blockquote=+Bajo+,bulletedelement=+Bajo+,includeonly=+Bajo+}	0,001015228

Tabla 3.8: Caso 5. Resultados de Eclat para Ediciones Comunes.

se producen valores de confianza más altos incluso si no existe una asociación entre los elementos[126]. Por otro lado, las Ediciones Comunes tenían un nivel de confianza máximo aceptable. Por último, sobre el *lift*, los grupos de las Ediciones Vandálicas y de Corrección tenían el peor nivel de *lift* indicando que esas reglas no eran patrones, en cambio, las Ediciones Comunes tenían niveles de *lift* aceptables indicando esas reglas como patrones.

La Tabla 3.7 tiene las reglas extraídas⁸ para el *cluster* 3 de las Ediciones Comunes. En esas reglas se puede ver la existencia de una tendencia a crecer en los artículos, es porque la porción de reglas con adiciones es mayor que las eliminaciones.

Las afirmaciones más significativas que se pudieron extraer de estas reglas fueron:

- Los enlaces internos de Wikipedia se utilizan normalmente con sangría y viñetas.

⁸Los resultados completos de las tablas se pueden ver en: <https://github.com/jonx18/Patterns-of-Markup-use-in-Wikipedias-Context/tree/master/5-Fifthcase>

- Es común ver que los enlaces y referencias externas se aplican y se eliminan junto con la sangría.
- Además, las citas y referencias se eliminan juntas, pero las adiciones de ambos wiki-markups juntos son ocasionales.
- A menudo, en la misma revisión que se añade un encabezado de tipo 2 se añade una sangría de tipo 2.
- Las adiciones y las eliminaciones casi nunca se realizan juntas; sólo se obtuvieron dos reglas que representan esta actividad y la más relevante fue el intercambio de un enlace externo a un enlace interno.

3.4. Resumen del capítulo

En este capítulo se presentó un análisis de patrones sobre los wiki-markups por el cual se demostró la existencia de patrones en el uso de los mismos. Estos patrones confirman la afirmación de que existen patrones de comportamiento en los datos históricos que permitirán la simulación de la actividad de los editores para la configuración automática de la estrategia de ludificación.

Como parte de este análisis, se presentaron estudios con diferentes granularidades de información y se demostró la relevancia de considerar la diferencia entre revisiones en lugar del estado final con la que estas se guardan. A partir de esto se considera que los wiki-markups pueden usarse como indicadores de las acciones de los usuarios durante su participación en la comunidad para comprender su comportamiento.

Junto con este estudio se presentan dos herramientas, Wikicrawler para la extracción de historiales de revisiones de artículos de Wikipedia y sus relacionados, y Wikigamificator para la extracción de eventos de dichos historiales. Ambas herramientas representan el primer paso en la integración de una CCC, en este caso Wikipedia, con una estrategia de ludificación simulada mediante datos históricos. Además, estas herramientas son un elemento importante a representar en el framework que se presentara en el Capítulo siguiente.

Capítulo 4

Incorporación de ludificación en CCC

En este capítulo se busca validar si es posible la integración de forma simulada de una estrategia de ludificación a una CCC, tomando como ejemplo a Wikipedia. Con este fin se caracteriza Wikipedia bajo el enfoque del framework MDA y de Metagame. Esta estrategia de ludificación se aplicará en forma de simulación sobre un subconjunto de la comunidad de Wikipedia, más exactamente la comunidad de agricultura. La posibilidad de simular la comunidad en la ludificación se basa en la existencia de patrones de comportamiento en los datos históricos como se mostró en el capítulo anterior. Además, se introducen las ventajas que presenta la integración de una estrategia de ludificación en primera instancia como una simulación. Finalmente, queda introducida la mecánica de simulación a optimizar en los siguientes capítulos de este trabajo de tesis.

4.1. Enfoque de MDA en Wikipedia

Para determinar cuál es el mejor modelo de ludificación en Wikipedia, es necesario establecer qué tipo de características tiene Wikipedia como CCC.

El conocimiento se puede categorizar en conocimiento individual y conocimiento colectivo. Mientras que el conocimiento individual es interno a un individuo, el conocimiento colectivo es generado, distribuido y compartido por los miembros de un grupo o comunidad. Wikipedia se presenta como un entorno de desarrollo que facilita el proceso de co-creación de contenido y generación de conocimiento colectivo. De este modo, los colaboradores comparten información e interviene en una reflexión colectiva sobre el tema del contenido. Esto lleva a un sentido de comunidad que comparte intereses y objetivos comunes [127].

Según [127], como una CCC, Wikipedia presenta un proceso de creación de conocimiento que involucra socialización, externalización y combinación de ideas. También la interacción de esta comunidad ocurre alrededor de la construcción del conocimiento. En este sentido, el valor que la comunidad busca al interactuar en este ambiente, se encuentra enfocado en la calidad más que en la cantidad [128] compartiendo y colaborando en el proceso de creación de conocimiento [127]. Si bien el aspecto clave del éxito de Wikipedia es la generación de contenido por parte de la comunidad, también es un desafío determinar la calidad de dicho contenido[129, 130].

Esto lleva a la conclusión de que las motivaciones en los colaboradores de Wikipedia son principalmente cognitivas, afectivas e integrativas[127]. Por lo cual un diseño cooperativo de ludificación basado en la promoción de la cooperación[131] es el mejor modelo de ludificación para Wikipedia.

Estas características de Wikipedia dan como resultado tres estéticas principales, expresadas en términos de MDA:

- Compañerismo (Fellowship), considerando el sistema como un framework social y colaborativo. La colaboración es un aspecto importante para hacer emerger el valor de calidad. Esto constituye el elemento principal en este diseño de juego.
- Desafío (Challenge), como propone la ludificación cooperativa, se refiere a los desafíos del equipo de una manera que promueve la superación de la comunidad, promoviendo la mejora del valor de la comunidad.
- Relación social (Social relatedness) está directamente relacionada con la ludificación cooperativa, refiriéndose a la satisfacción de ser parte de un esfuerzo grupal [131].

La dinámica está definida por las principales acciones de Wikipedia, agrupadas en tres categorías principales en congruencia con la estrategia de ludificación a aplicar (en este caso Metagame), que son:

- Contribuciones (Contributions), que representan las acciones vinculadas a la creación de un nuevo artículo, o agregar un nuevo párrafo en artículos preexistentes.
- Refuerzo (Reinforcement), acciones que se refieren a la edición de otros artículos con cambios menores, como agregar entradas de discusión en las páginas de charla de Wikipedia o comentarios en Stack Overflow.
- Difusión (Dissemination) que se refiere a acciones sociales de propagación y difusión, como agregar enlaces externos o internos o discutir la calidad de los artículos con la comunidad.

4.2. Simulación de Metagame

Esta sección presenta la aplicación de Metagame mediante un simulador que se ejecuta sobre los datos históricos de una porción de Wikipedia. Esta simulación generara acciones de los jugadores en Metagame basándose en las acciones de los editores que se encuentren en los datos históricos de los artículos. En este caso se seleccionaron los artículos ligados a la categoría Agricultura. Con este fin, se describen los artículos seleccionados en su versión inglesa a fin de comprender su estructura y los elementos que se utilizaran de los mismos para la simulación. Luego, se definirá la problemática de insertar Metagame a una CCC continuando con el ejemplo de la comunidad seleccionada para este capítulo. También buscara introducirse la importancia de simular la aplicación de Metagame sobre datos históricos en esta CCC en lugar de una implementación concreta sobre la misma. Finalmente se presentará una evaluación de los resultados de esta simulación que servirá de motivación para la creación de un framework de ajuste de ludificaciones a CCC.

4.2.1. La comunidad de Agricultura en Wikipedia

Wikipedia como una gran CCC incluye diferentes subcomunidades que se agrupan en categorías de acuerdo a sus áreas de conocimiento [132, 133]. Por lo cual, se presentará el análisis la Comunidad de Construcción de Conocimiento de la categoría de Agricultura¹ en la versión inglesa de Wikipedia².

La categoría de Agricultura se define, según Wikipedia, como *“el proceso de producción de alimentos, piensos, fibras y otros productos deseados mediante el cultivo de ciertas plantas y la cría de animales domésticos”*. Incluye 34 subcategorías y 2.394.712 artículos³.

Para el análisis de la simulación, se seleccionó la subcategoría `‘‘Agriculture by type’’` para representar una comunidad más pequeña de editores centrados en artículos sobre agricultura. Estos artículos están relacionados principalmente con la forma en que se produce la agricultura. Esta subcategoría incluye a su vez 16 subcategorías. La Tabla 4.1 detalla en la primera columna el nombre de la categoría (subcategoría de `‘‘Agriculture by type’’`), en la segunda columna el número de subcategorías y, en la última columna el número de artículos que incluye la categoría y sus subcategorías.

¹<https://en.wikipedia.org/wiki/Category:Agriculture> Accedido el 25 de abril de 2018

²<http://en.wikipedia.org> Accedido el 25 de abril de 2018

³Basado en el análisis de las páginas de PetScan con la siguiente configuración https://petscan.wmflabs.org/?language=en&project=wikipedia&depth=10&categories=Agriculture&ns%5B0%5D=1&interface_language=en&active_tab= Accedido el 25 de abril de 2018

Nombre de categoría	#Sub-categoría	#Artículos
Aquaculture	7	107
Beekeeping	11	94
Cannabis cultivation	3	22
Commercial farming	2	5
Dairy farming	6	39
Domesticated animals	18	83
Forestry	30	52
Hemp agriculture	2	1
Livestock	25	158
Orchards	1	12
Organic farming	10	76
Permaculture	4	30
Pig farming	2	15
Poultry farming	3	44
Sustainable agriculture	16	184
Viticulture	4	59

Tabla 4.1: Agriculture by type en Wikipedia.

Contribución	Refuerzo	Difusión
Agregar un nuevo párrafo	Añadir entrada de conversación	Compartir en WP
Crear un nuevo artículo	Modificar un párrafo	Añadir enlace interno
Añade una referencia	Borrar una frase o menos	Añadir enlace externo
Añade los encabezamientos	Hacer un cambio menor	Añadir redirecciones
Añadir imagen	Mover frases	
Añade una tabla o una fila		
Añadir lista o elementos		

Tabla 4.2: Eventos de Wikipedia en el contexto de Metagame.

4.2.2. Definición de la problemática

Insertar Metagame a un proyecto de construcción de conocimiento de la comunidad implica varios pasos, los cuales analizaremos desde la perspectiva de integrar la estrategia de ludificación a la CCC de Agricultura de Wikipedia. Desde la perspectiva de la Gestión de servicios de TI (ITSM por su nombre en inglés) para la mejora de los procesos [70], el problema de insertar Metagame sobre la comunidad de Agricultura de la Wikipedia podría definirse en tres actividades clave: definir el proceso, construir el modelo de simulación del proceso y ejecutar la experimentación del modelo de simulación. A continuación, se presenta en detalle cada actividad en el contexto de la comunidad de Agricultura.

1. **Definir la organización del proceso:** identificar las acciones en la edición de los artículos del proyecto que actuará como acciones en el

juego, tales como ediciones, añadir un enlace, añadir un encabezamiento, eliminar un párrafo, etc.

2. **Construir el modelo de simulación del proceso:** Se debe definir la transformación de los eventos del proyecto (en este caso ediciones de Wikipedia) a las acciones del juego para crear una experiencia agradable. Para ello, hemos definido dos reglas de mapeo: Eventos de la CCC objetivo a acciones de Metagame y, nivel de dificultad para la adquisición de insignias.

En primer lugar, **el mapeo entre las acciones en Wikipedia y las acciones definidas en Metagame**. Este mapeo se detalla en la Tabla 4.2 donde cada columna representa una lista de acciones, la primera columna enumera las acciones que se consideran como contribuciones, la segunda las que se consideran como refuerzos y la tercera las de difusión. Además, si bien no se menciona en el listado, se tendrá en cuenta cada revisión como una acción de Login para indicar los comienzos de actividad de un editor. Claramente, la aproximación del mapeo contempla la adición de elementos como una clara contribución y la edición de elementos existentes como refuerzos. Las acciones de difusión son sólo las que se expresan en el contexto de Wikipedia, como la acción de enlazar un artículo que a otro artículo mediante enlaces internos o el proyecto Wikidata. Lamentablemente, las difusiones en redes sociales como Facebook no están registradas ni en el historial de revisiones ni en otros proyectos de la familia Wikimedia, por lo cual no serán analizadas aquí.

Por último, en la Tabla 4.3 se definen **los niveles de dificultad para la adquisición de las insignias**. La primera columna contiene el número de acciones requeridas, la segunda el tipo de acción, la tercera columna tiene el nombre de la insignia y la última columna el número de niveles. La primera columna muestra cuántas acciones de un tipo tiene que realizar un jugador para obtener una placa. Por ejemplo, la primera línea debe leerse como *“el jugador que realice 1 Login recibirá la insignia Yo estuve aquí con nivel 1”*, en consecuencia la tercera línea podría leerse como *“el jugador que realice 4 (2*2) Contribuciones recibirá la insignia Unidad de Trabajo con nivel 2”*.

Como vemos, se han definido 10 niveles para las insignias **Unidad de Trabajo**, **Ida y vuelta** y **Gritar a viva voz**. El valor que multiplica el número de acciones en la primera columna indica la facilidad asociada a esa acción. Cuanto más grande es el multiplicador, más fácil se considera la acción. Esta escala de dificultad se planteo en base a como se considero la dificultad de las acciones de cada tipo, considerando que las acciones mas difíciles para los editores son las de diseminación, seguidas por contribución y finalmente refuerzo se

# Acciones	Tipo	Insignia	Niveles
1	Login	Yo estuve aquí	1
2	Login	Bienvenido de vuelta	2
2*nivel	Contribución	Unidad de Trabajo	1..10
10*nivel	Refuerzo	Ida y vuelta	1..10
1*nivel	Diseminación	Gritar a viva voz	1..10

Tabla 4.3: Puntos para ganar insignias en Metagame.

decidió brindar una dificultad incremental siguiendo dicho orden. Esto conlleva a que las acciones de diseminación son las que requieren menos acciones para obtener insignias y las de refuerzo las que mas. Por último, como Metagame se define como un juego, los “editores” de Wikipedia son representados también como jugadores de Metagame.

3. **Experimentación con el modelo de simulación:** Se vuelven a ejecutar todas las acciones registradas en el historial de ediciones de Wikipedia sobre Metagame en forma secuencial, y luego de esto se analizan las insignias y rangos que obtuvieron cada uno de los editores.

En los siguientes apartados se detalla la metodología implementada para realizar las actividades de simulación.

4.2.3. Metodología

La simulación en el uso de Metagame con los proyectos de Agricultura de Wikipedia se organizó en tres tareas. La primera tarea es la encargada de obtener de Wikipedia todos los artículos de Agricultura con su historial de revisiones, páginas de discusión, y otros artículos con un enlace a ellos. Esta tarea es detallada en la sección “**Extracción de Wikipedia**”, aprovechando la información existente generada previamente por los editores en Wikipedia, evitando así forzar la participación de los editores en una estrategia de ludificación aun sin analizar. La siguiente tarea, detallada en la sección “**Generación de actividades**”, genera una lista cronológica de todas las acciones de edición en Wikipedia para transformarlas en base a Metagame siguiendo las reglas descritas en la Tabla 4.2. Esta lista se usará como entrada para ejecutar la simulación de Metagame. Finalmente, Metagame se configura con las reglas de adquisición de insignias que se detallan en la Tabla 4.3. La Figura 4.1 presenta el flujo de los datos en forma descendente comenzando con las fuentes de información hasta la ludificación donde cada paso recibe de entrada los datos del paso anterior de poseerlo. En las siguientes subsecciones describen estas tres tareas.



Figura 4.1: Wikigamification con Metagame.

Extracción de Wikipedia

Esta tarea obtiene de Wikipedia todo el historial de revisiones de las páginas relevantes. Para ello se utilizó la herramienta **WikiHistoryCrawler** introducida previamente en el Capítulo 3. En este contexto, las páginas que pertenecen directamente a las subcategorías de “Agriculture by type” (Tabla 4.1) de la Wikipedia en su versión inglesa. Además, también se incluyen sus páginas de discusión y cualquier página de Wikipedia con un enlace a ellas. Un ejemplo de cómo se incluyen páginas enlazadas puede ser el siguiente; **Humus**⁴ es una página relevante porque pertenece a la categoría **Organic farming**. En cambio **Organic food** se incluye porque tiene un enlace al artículo **Humus**, aunque **Organic food** no pertenece a una subcategoría de **Agriculture by type**. Además, como en Wikipedia, un artículo podría pertenecer a más de una categoría, solo se seleccionaron artículos con categorías disjuntas.

Generación de actividades

A partir del historial de revisiones obtenido en la tarea anterior, la herramienta **Wikigamificator**, también introducida previamente en el Capítulo 3, genera la lista cronológica con todas las acciones de edición en Wiki-

⁴<https://en.wikipedia.org/wiki/Humus> Accedido el 25 de abril de 2018

pedia pero con acciones de Metagame⁵. Los cambios son analizados con Sweble[122] y mapeados a las acciones listadas en la Tabla 4.2 (por ejemplo Añadir título o mover párrafo). Llamamos a esta lista **Dataset de eventos**. Este proceso se muestra en la Figura 4.1 como **Mapeo de Eventos**.

Configuración de Metagame

Para la configuración se ha considerado cada subcategoría de la Tabla 4.1 como un proyecto específico. Por lo tanto, un editor que edite una página en una subcategoría específica está participando en el proyecto de la subcategoría. Todos estos proyectos se crean en Metagame con las instrucciones de la línea 2 hasta la 22.

Luego, hemos definido 10 niveles de insignias de Contribución, Refuerzo y Difusión y 2 niveles de insignias de Login para cada proyecto. Líneas 25 a 44 en el Listado 4.2.3. La configuración de Metagame se hizo como muestra el Listado 4.2.3.

```

1 #Proyectos
2 projects= [
3   "Category:Aquaculture",
4   "Category:Beekeeping",
5   "Category:Cannabis cultivation",
6   "Category:Commercial farming",
7   "Category:Dairy farming",
8   "Category:Domesticated animals",
9   "Category:Forestry",
10  "Category:Hemp agriculture",
11  "Category:Livestock",
12  "Category:Orchards",
13  "Category:Organic farming",
14  "Category:Permaculture",
15  "Category:Pig farming",
16  "Category:Poultry farming",
17  "Category:Sustainable agriculture",
18  "Category:Viticulture"
19 ]
20
21 projects.each do |pname|
22   proje = Project.create(name:pname)
23
24 #Insignias de tutorial
25 Badge.create(name:"i-was here",project_id:proje.id,
26   points:1, badge_type:"login")
27 Badge.create(name:"welcome-back",project_id:proje.id,
28   points:2, badge_type:"login")
29 (1..10).each do |i|
30   #Insignias de Unidad de trabajo

```

⁵Los editores no registrados se agrupan con el nombre "Anónimo"

```
31     Badge.create(name:“#{i}-contribution”,
32     points:(2*i), project_id:proje.id,
33     badge_type:“contribution”)
34
35     #Insignias Ida y Vuelta
36     Badge.create(name:“#{i}-reinforcement”,
37     points:(10*i),project_id:proje.id,
38     badge_type:“reinforcement”)
39
40     #Insignias grita a viva voz
41     Badge.create(name:“Share-on-#{i}”,
42     points:(1*i), project_id:proje.id,
43     badge_type:“dissemination”)
44     end
45 end
```

Listado 3.2.3 Código de configuración de Metagame

4.2.4. Evaluación

La evaluación se llevó a cabo con el fin de comprender como se ajustaría la estrategia de ludificación con esa configuración particular a una determinada comunidad. Con ese objetivo se buscaron responder a las siguientes preguntas:

- P1: ¿Cómo se distribuyeron los editores en función de su actividad?
- P2: ¿Qué cantidad de los eventos fueron producidos por cada tipo de editor?
- P3: ¿Cómo fue la distribución de los eventos en los proyectos?
- P4: ¿Cómo es la evolución de las acciones a lo largo del tiempo?
- P5: ¿Cuál fue la distribución de rango de los jugadores en Metagame?
- P6: ¿Cómo fue la entrega de las insignias a lo largo del tiempo?

Sobre el conjunto de datos utilizado, se obtuvo el historial de revisiones de 4.690 artículos, incluidas las secciones de discusión y los artículos relacionados extraídos en abril de 2018. Esto se trasladó en un conjunto de 9.939.296 acciones de Metagame con jugadores Anónimos (8.826.965 sin Anónimos). En total, 29.783 jugadores fueron identificados. Esas revisiones se hicieron entre abril de 2001 y abril de 2018, es decir, revisiones de 17 años. Todo el código y el conjunto de datos de eventos utilizados en la evaluación se puede encontrar en <https://github.com/cientopolis/wikigamification>.

Se ha realizado un análisis previo de las actividades de Wikipedia teniendo en cuenta los editores anónimos. Luego de ese análisis, debido a que

Metagame requiere identificar por nombre a los jugadores, se realizó la simulación sin los editores anónimos.

4.2.5. Resultados

A partir del análisis de los jugadores en el dataset de acciones, se definió una división en grupos de jugadores por la intensidad de la actividad inspirada en [134]. Para dividir los activadores por su frecuencia de actividad se utilizaron como líneas de corte la media, la mediana y el desvío estándar del número de acciones de cada jugador. Por lo cual los grupos se dividieron de la siguiente manera:

- **Turista** es el jugador que tiene menos de 10 actividades, es decir, menos de la mediana.
- **Interesado** es el jugador que tiene más de 10 actividades pero menos de 296 actividades, es decir, menos de la media.
- **Promedio** es el jugador que tiene más de 296 actividades pero menos de 4924 actividades, es decir, menos que el desvío estándar.
- **Activo** es el jugador que tiene más de 4924 actividades.

Para responder a la pregunta P1, analizamos la proporción del total de jugadores que pertenecen a cada grupo como se muestra en la Figura 4.2. En este análisis, las acciones anónimas se consideraron realizadas por un solo jugador. Como es de esperar, la mayoría de los jugadores son turistas (52 %) o interesados (39,5 %), y habían hecho menos de 10 acciones o 296 acciones, respectivamente. Por otro lado, los jugadores activos son menos del 1 %.

Continuando con el estudio de los jugadores para responder a la pregunta P2, la Figura 4.3 muestra la proporción de acciones que pertenecen a cada grupo de jugadores, en este caso teniendo en cuenta también los jugadores Anónimos. Muestra que la mayoría de las acciones fueron producidas por los jugadores activos (52 %). Por el contrario, los jugadores turistas produjeron sólo el 0,74 % de las acciones. Además, el 11,2 % de las actividades fueron producidas por los jugadores anónimos. Como conclusión, tanto la Figura 4.2 como la Figura 4.3 muestran que un número reducido de jugadores produjeron más contenido que los otros jugadores debido a su alto nivel de actividad. Sin embargo, es importante señalar la relevancia de la producción de los jugadores Anónimos.

Para responder a la pregunta P3 analizamos la distribución de eventos por proyecto. La Figura 4.4 muestra un gráfico de barras con el análisis. El eje “Categorías” detalla el nombre de cada categoría de ‘ ‘Agriculture by type’ ’ incluyendo el número de artículos que contiene y, en el eje “Porcentaje de Acciones” aparece una barra con las proporciones y tipos de accio-

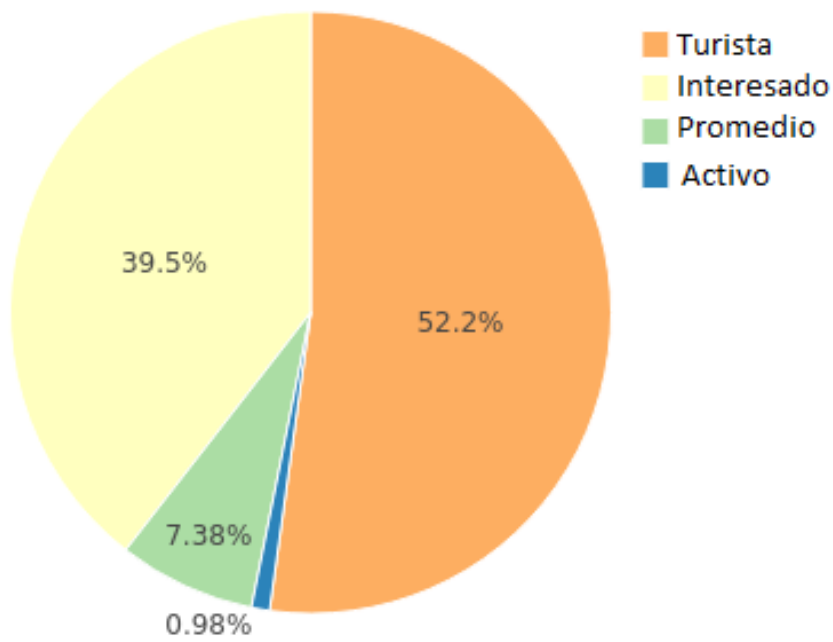


Figura 4.2: Tipo de jugadores por nivel de actividad.

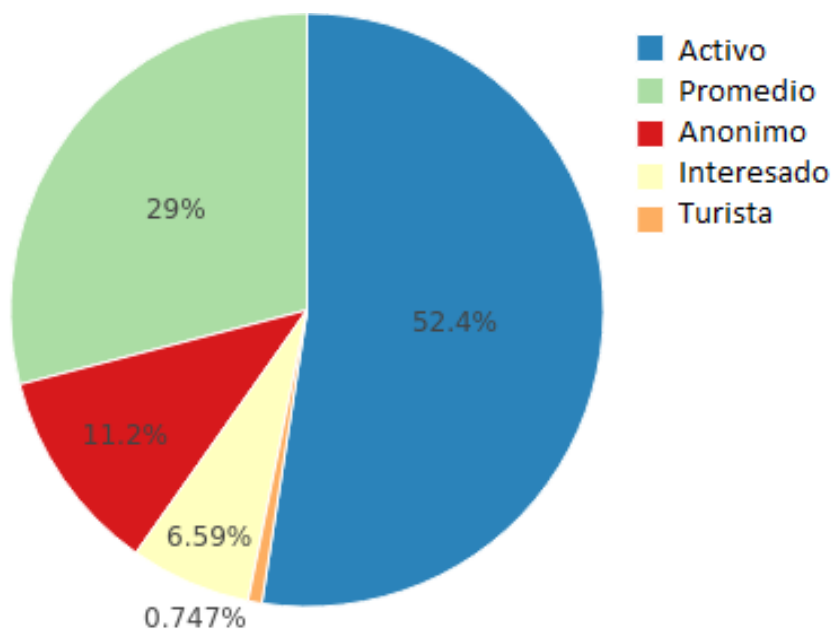


Figura 4.3: Número de actividades por tipo de jugador.

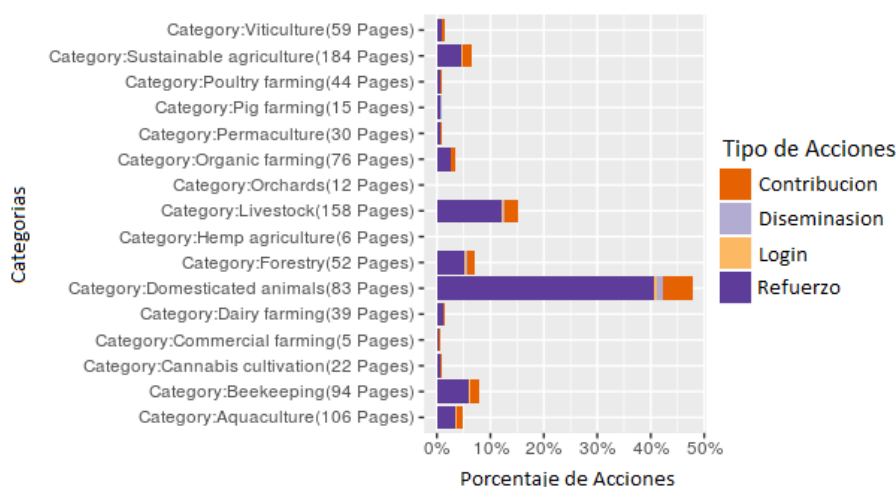


Figura 4.4: Distribución de eventos por proyecto.

nes. Según el gráfico, en general, todas las categorías tienen una proporción similar de acciones. Las acciones más representativas son las de refuerzo, luego las de menor proporción de contribuciones y finalmente las de difusión y login. Sorprendentemente, la categoría con más acciones recibidas fue “Animales domesticados” (Domesticated animals) que sólo tenía 83 páginas, indicando que el volumen de acciones no está necesariamente correlacionado con el volumen de páginas de una categoría.

En cuanto a la evolución de las acciones a lo largo del tiempo (pregunta P4) analizamos la acumulación de actividades con un gráfico de líneas. La Figura 4.5 muestra la acumulación de las acciones de Metagame por tipo. La acumulación se refiere al número de acciones que se registraron hasta un día determinado. El eje “Número de Acciones” representa el número de acciones en millones, y el eje “Tiempo” describe el momento en que cada acción fue realizada en Wikipedia. Hay una línea para cada tipo de acción desde 2001 hasta 2018.

En cuanto al inicio de las acciones, el gráfico muestra que en 2005 se iniciaron las actividades más intensas. Entre 2005 y 2010 se produjo un rápido crecimiento en el número de acciones de “Refuerzo”. También, desde 2010 hasta la última acción en abril de 2018, el número de acciones siguió creciendo pero más lentamente. Respecto a cada tipo de acción, las acciones de “Refuerzo” crecieron más rápido y más que las otras. Las acciones de “Contribución” mostraron un crecimiento lento. Por otro lado, las acciones de “Difusión” nunca tuvieron un crecimiento relevante y, las acciones de “Login” no pueden ser vistas.

La gran diferencia en la cantidad de acciones de “Refuerzo” en com-

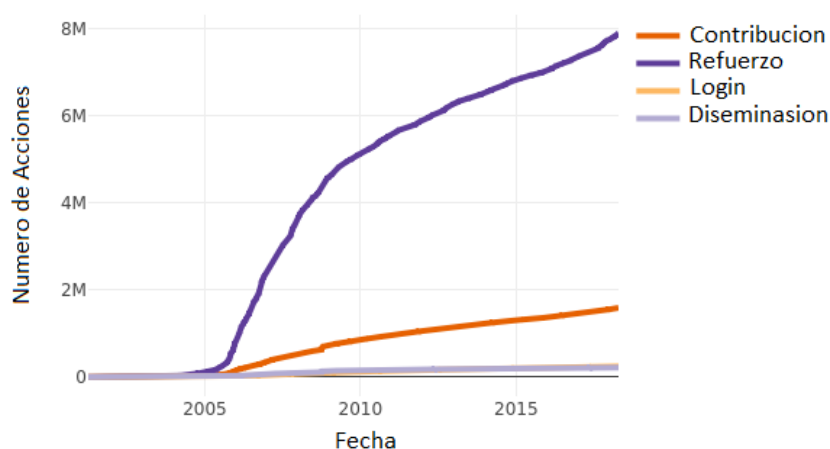


Figura 4.5: Las acciones de Metagame en el tiempo.

paración con las acciones de “Contribución” representa que la mayoría de las actividades que hemos mapeado como “Refuerzo” podrían ser realmente “Contribución” en Wikipedia. La construcción de conocimiento en Wikipedia se hace en porciones muy pequeñas, y algunas de ellas pueden ser confundidas como refuerzos en lugar de contribuciones.

Sobre los resultados de Metagame, la Figura 4.6 muestra para cada proyecto la distribución de los jugadores por rango. El eje “Categorías” presenta el nombre de cada categoría con el número de artículos que contiene, en el eje “Porcentaje de rangos” existe para cada categoría una barra que representa la proporción de jugadores que participan en esta categoría diferenciados por su rango por color.

Debido a que el rango pertenece al jugador y éste podría participar en muchos proyectos, es interesante ver cómo los proyectos con menos artículos como la categoría “Agricultura de cáñamo” o “Acuicultura” se llevaron a cabo por jugadores del rango más alto obtenido en esta simulación. Esto se relacionó con que los artículos menos interesantes o relevantes fueron apoyados por los editores más comprometidos. Además, se pudo observar que la mayoría de los editores que interactuaban en algún proyecto eran editores ocasionales y por eso tenían el rango de “Visitante”. Esta figura y la siguiente ayudaron a responder al P5.

La Figura 4.7 muestra la distribución total de los jugadores en sus rangos. Los rangos obtenidos durante la simulación fueron “Visitante”, “Explorador” y “Ciudadano Científico”. Además, algunos rangos nunca fueron alcanzados como “Científico Ciudadano Prolífico” o “Científico Ciudadano Comprometido”. Se demostró que la dificultad era demasiado alta porque en los 17 años que abarcó esta simulación nadie pudo obtener un rango su-

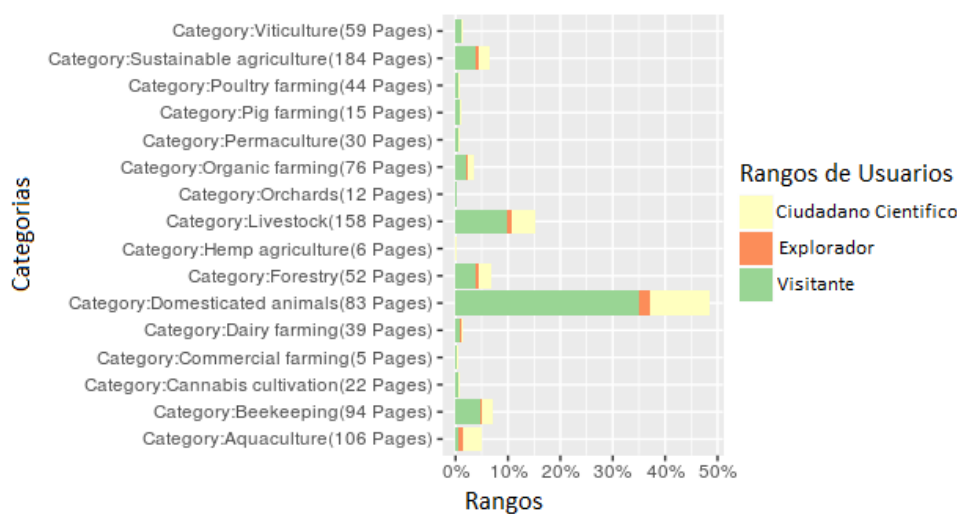


Figura 4.6: Jugadores de Metagame por proyecto y por rango.

perior al de “Ciudadano Científico”. Esta dificultad está relacionada con la forma en que se consideraron algunas de las acciones. Por ejemplo, algunas acciones de “Refuerzo” podrían ser consideradas como “Contribución” porque en Wikipedia algunas contribuciones son más pequeñas de lo que se consideró, como nuevas frases en lugar de párrafos. Además, la distribución del conjunto de datos en muchos proyectos podría ser demasiado pequeña, tal vez si algunos de esos proyectos se fusionaran el jugador podría reunir más insignias y acceder a rangos más altos.

En cuanto a la entrega de las insignias en el tiempo (pregunta P6) analizamos la acumulación de insignias con un gráfico de líneas. La Figura 4.8 muestra la acumulación de insignias entregadas cada día por tipo de insignia. El eje “Número de Insignias” representa el número de insignias en miles, y el eje “Tiempo” describe el momento en que cada insignia fue ganada por un jugador en Metagame. Hay una línea para cada tipo de insignia desde 2001 hasta 2018. Una línea cerca de la parte inferior del gráfico representa que las insignias de este tipo de acciones se obtuvieron menos veces. Además, las líneas que estaban más cerca entre sí se refieren a que esas líneas obtuvieron insignias un número similar de veces. Además, una insignia que no se obtuvo muchas veces podría ser considerada más difícil de obtener. Un ejemplo de ello fue el caso de las insignias de difusión.

Sobre las líneas de insignias más cercanas, se podrían considerar que están equilibradas en lo que respecta a dificultad. Un ejemplo de ello fueron las insignias de contribución, y las insignias de refuerzo porque, aunque había más acciones de refuerzo como se mencionó anteriormente, los requisitos para



Figura 4.7: Metagame. Porcentaje de jugadores por rangos.

las insignias de refuerzo eran 5 veces más grandes que para las insignias de contribución. Finalmente, otra cosa a considerar en este paso fue si las actividades de difusión tienen que ser más fáciles, por ejemplo, añadiendo más tipo de acciones de difusión o si es mejor premiar más las acciones actuales. Si esta implementación se hubiera realizado directamente sobre la comunidad de Wikipedia, este caso sería de suma importancia, porque esta dificultad hace casi imposible el acceso a los rangos más altos y eso podría haber frustrado a los jugadores.

4.3. Resumen del capítulo

En este capítulo se presentó un ejemplo de ludificación para analizar la posibilidad de integración de las ludificaciones en CCC. Aplicando una ludificación preexistente, Metagame, la cual puede analizarse desde la perspectiva de MDA. A partir de esto se presentó una integración simulada de la ludificación en la comunidad de Agricultura de Wikipedia lo cual permite observar los siguientes aspectos:

- Se pudo confirmar la factibilidad de simular la aplicación de una estrategia de ludificación mediante datos históricos.
- La importancia de la simulación de la ludificación con la CCC objetivo para evitar resultados disruptivos. Como por ejemplo los que podrían haberse generados al aplicar los ajustes presentados en este capítulo sobre la comunidad de Wikipedia directamente sin antes evaluarlos.

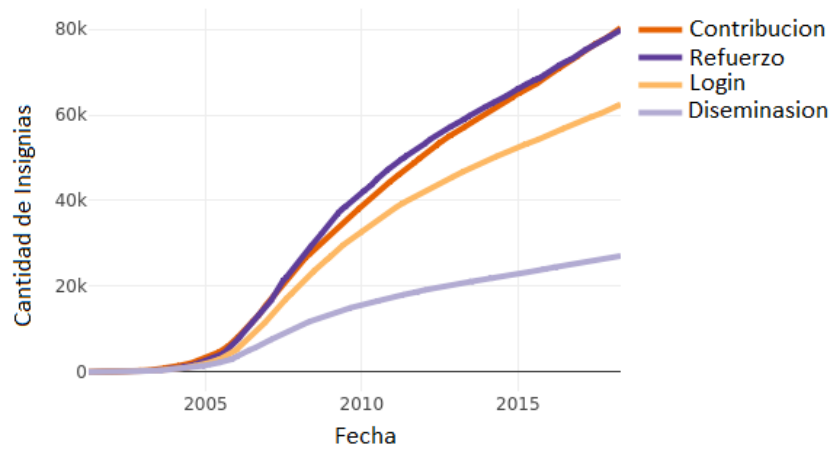


Figura 4.8: Entrega de insignias en el tiempo.

- La gran cantidad de variables a tener en cuenta para una ludificación sencilla como Metagame en una CCC. Desde el mapeo de las acciones e insignias, hasta el mismo comportamiento de los miembros de la CCC.
- Aspectos propios de esta CCC en particular como bien puede ser que la mayoría de los aportes surgen de un grupo reducido de editores muy activos y editores anónimos.

En el próximo capítulo se observará una propuesta de solución no intrusiva para la configuración de ludificaciones para distintas CCC.

Capítulo 5

Enfoque de solución

Este capítulo toma la problemática de generar una configuración óptima y los efectos adversos obtenidos a partir de la configuración deficiente presentada en el ejemplo del Capítulo 4 a fin de presentar una propuesta de solución. Con este fin, presenta un enfoque de solución no intrusivo, ni disruptivo para hallar la mejor configuración de una estrategia de ludificación en una CCC de forma automática. Si bien ya se posee noción a partir del análisis anterior, primero se introduce la idea de la simulación de las estrategias de ludificación como una forma de evaluar su aplicación en una CCC en base a sus datos históricos. Finalmente se presenta una estrategia de Reinforcement Learning como metodología que nos permitirá obtener una configuración para la estrategia de ludificación aplicada en una CCC, junto con un agente entrenado para generar esa configuración.

5.1. Simulación de la ludificación

Aunque el proceso de ludificar se considera un proceso iterativo de diseño, desarrollo, evaluación, seguimiento y adaptación[135], este proceso es complejo y podría ser disruptivo para la comunidad de no aplicarse correctamente. Una forma no disruptiva de probar la mecánica o la configuración de una estrategia de ludificación en una comunidad específica se presentó previamente en el Capítulo 4. En ese capítulo se simula la aplicación de una estrategia de ludificación con una configuración fija, utilizando un historial de actividad en una CCC. De esta forma se evitan los cambios que lleguen directamente a la comunidad evitando cambios que podrían resultar confusos o molestos para los miembros.

Otra ventaja de aplicar una estrategia de ludificación de forma simulada sobre datos históricos es que brinda flexibilidad sobre la implementación de la misma. La simulación permite que se sume complejidad a la ludificación

brindando accesos programáticos a la configuración. De estos accesos se obtienen facilidades en el acceso al estado interno de la ludificación a lo largo de toda la simulación. Estas capacidades adicionales son de gran valor a la hora de poder utilizar técnicas de configuraciones automáticas.

5.2. Aprendizaje por refuerzo como enfoque

El aprendizaje por refuerzo o Reinforcement Learning (RL), es una técnica de aprendizaje automático que difiere de los algoritmos de aprendizaje supervisado porque no requiere de pares correctos de entrada/salida como información para el entrenamiento. En contraste, requiere de tan solo un objetivo definido que se utiliza para establecer las políticas de optimización (comúnmente mediante maximización), de esta forma es posible aprender todas las características de dichas políticas en un entorno desconocido[88].

Por lo tanto, en el contexto de querer ludificar una CCC este enfoque se ajusta mejor que por ejemplo los algoritmos tradicionales de aprendizaje supervisado. Esto se debe a que no hay pares correctos de entrada/salida que definan una configuración correcta de una estrategia de ludificación. En estas técnicas se busca aprender la relación entre dichos pares para generar luego de forma automática las salidas a partir de una entrada. Sin embargo, al no disponer de esos datos no es posible aplicar técnicas típicas de aprendizaje supervisado. Sin embargo, al tratarse de la aplicación de una estrategia de ludificación podría definirse una política de satisfacción en su desempeño para una CCC como nuestro entorno desconocido, y esta política podría actuar como función de recompensa.

Dada la posibilidad de evaluar la aplicación de una estrategia de ludificación utilizando una política de satisfacción, **la problemática de obtener configuraciones óptimas mediante una simulación con datos históricos puede considerarse como un problema de aprendizaje por refuerzo donde un agente de RL busque maximizar la política de satisfacción recomendando cambios a la configuración actual a fin de optimizarla**. Los elementos de un proceso de RL, como el que se introdujo en el Capítulo 2, para esta problemática son el agente de RL y un entorno con la lógica para la aplicación de algoritmos de RL que contendrá nuestra simulación.

Como perspectiva general del enfoque, el entorno comienza en un estado s_t con una configuración inicial, comúnmente aleatoria para ayudar a la generalización del agente. Junto con este estado se computa la función de recompensa $R(s_t)$, la cual sirve de métrica para comprender que tan bien se ajusta la configuración al objetivo de esa función y a la simulación en sí. Además, la función de recompensa indica que tan bien se está desempeñando el agente, ya que es el que realiza modificaciones en la configuración para

ajustarla.

Por lo tanto, el agente que a partir de ese estado s_t realiza una acción que modifica la configuración actual para el siguiente ciclo de ejecución, $a_t + 1$, buscando maximizar la función de recompensa $R(s_t + 1)$. En la sentencia anterior $a_t \in A$ con A se encuentra representando el espacio de acciones que puede realizar el agente sobre el entorno, lo cual para nuestra problemática consiste en acciones que cambien la configuración en alguno de sus valores. Con la configuración modificada, el entorno retorna al agente los resultados de evaluar esta nueva configuración $s_t + 1$ y la recompensa $R(s_t + 1)$. Con $s_t + x \in S$ donde S se encuentra representando los posibles estados internos del simulador según la configuración actual y el conjunto de datos históricos utilizado para la simulación.

Este proceso es un proceso iterativo que continuaría hasta encontrar una configuración aceptable o un límite máximo de cambios en la configuración. Mediante la repetición de este proceso se espera el agente aprenda a realizar un conjunto de acciones de la forma $[a_t, a_t + 1 \dots a_t + n]$ que maximizasen la función de recompensa $R(s_t + n)$. Además, esta función dará espacio a definir que se considera que la configuración de la estrategia de ludificación sea exitosa.

5.3. Resumen del capítulo

Este capítulo presenta una propuesta de solución a la problemática de integrar correctamente una ludificación a una CCC mediante la obtención de configuraciones óptimas desde la perspectiva de RL. Este enfoque se basa en la simulación de la ludificación sobre datos históricos reales de la comunidad objetivo.

Además, se presentan las ventajas del uso de datos históricos para la evaluación del comportamiento de usuarios y analizar las dinámicas internas de una comunidad. Luego se introduce el concepto de la simulación de la estrategia de ludificación que se busca optimizar. Esto permite abstraer a la comunidad del proceso iterativo de la configuración de una estrategia de ludificación, el cual podría ser disruptivo para la comunidad. Finalmente, la simulación también permite la aplicación de las técnicas de RL para la configuración automática.

Capítulo 6

Framework de reinforcement learning para automatización de configuración en CCC

En este capítulo se introduce una implementación prototípica que resuelve la problemática presentada en el Capítulo 4. Esta problemática consiste en encontrar configuraciones óptimas para una ludificación a fin de evitar configuraciones dañinas o disruptivas para una CCC. La implementación que se presenta aquí se basa en el enfoque previamente descrito en el Capítulo 5.

Por lo cual, se presenta un framework de Reinforcement Learning para el entrenamiento de agentes que se encarguen de configurar automáticamente estrategias de ludificación para una CCC. En orden de introducir el framework, primero se realizará una introducción de las tecnologías a utilizar, a fin de tener comprensión de las bases sobre las que se construyó el framework presentado en este trabajo doctoral. Posteriormente se presentará el framework propiamente dicho con detalles sobre la implementación de sus distintos elementos y su capacidad de extensión.

6.1. Presentación tecnologías

En esta sección se presentarán las tecnologías utilizadas para la realización de este framework junto con las relacionadas a los agentes que interactuarán con el mismo. Como base sobre la que se desarrolla el framework se introduce el conjunto de herramientas GYM de Open AI para entrenamiento de agentes por RL que permitirá adoptar un estándar utilizado por múltiples agentes de RL permitiendo así una mejor integración de los mismos. Luego

se presentan también las librerías con las que se construyen estos agentes a fin comprender similitudes y diferencias de los mismos.

6.1.1. OpenAI Gym

GYM es una librería con un conjunto de herramientas para el desarrollo y comparación de algoritmos de RL[105]. Esta librería está centrada en facilitar la comparación de algoritmos de RL, dado que estos algoritmos se centran en la relación entre agentes de RL y el entorno (environment), GYM brinda herramientas para la estandarización de los entornos para que la efectividad de los agentes sea comparable. GYM define solo estandarización para el entorno y no para el agente a fin de brindar libertad en la estructura del mismo para permitir todas las posibles implementaciones.

La principal abstracción de GYM es la interfaz *Env*, que es la interfaz que unifica el entorno. Sus principales métodos son:

- *reset(self)*: Reinicia el estado del entorno y devuelve un conjunto de observaciones, perteneciente al *observation_space*.
- *step(self, action)*: Avanza una acción en la ejecución del entorno. Retorna un conjunto de observaciones(*observation*), la recompensa actual (*reward*), información sobre el fin de ejecución(*done*) e información de debug(*info*).
- *render(self, mode = 'human')*: Es un método utilizado para visualizar un paso del entorno de forma legible por un ser humano.

Si bien esta interfaz permite la implementación genérica de un entorno, GYM brinda también herramientas para la generalización de las observaciones y las acciones definiendo un espacio de observaciones y un espacio de acciones que pertenecen a la clase *Space* la cual subclasifica en diversos tipos de espacios algunos de los cuales se listan a continuación:

- *Discrete*: El espacio discreto permite un rango fijo de números no negativos.
- *Box*: El espacio de matriz representa una matriz de n dimensiones.
- *Tuple*: El espacio de tupla representa una tupla conformada por espacios simples similares.
- *Dict*: Es el espacio utilizado para definir un diccionario de espacios.

6.1.2. Deep learning

Esta sección introduce dos conjuntos de herramientas para el aprendizaje profundo o Deep Learning que son utilizados por librerías de agentes de RL.

TensorFlow y Keras

Inicialmente desarrollada como parte del proyecto ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), Keras es una biblioteca de código abierto para redes neuronales. Actúa como capa de abstracción entre diferentes frameworks para aprendizaje automático como puede ser TensorFlow, CNTK, o Theano. Entre sus abstracciones brinda varios tipos de funciones de activación para las redes neuronales, así como varios tipos de capas predefinidas como pueden ser capas convolucionales, además permite definir las redes neuronales como procesos secuenciales o mediante definiciones funcionales.

Si bien Keras permite abstraer funciones para múltiples frameworks de aprendizaje automático, en este trabajo se utilizó junto con TensorFlow. TensorFlow es una plataforma de código abierto de aprendizaje automático que está compuesta de un ecosistema flexible de herramientas, librerías y recursos para permitir la implementación del estado del arte en aprendizaje automático, así como su aplicación en entornos de producción. Esta plataforma al momento de la escritura de la tesis¹ se encontraba en su versión 2. Entre esas herramientas, integra Keras como una API de alto nivel de TensorFlow para construir y entrenar modelos de aprendizaje profundo, con tres ventajas clave:

Amigable al usuario: Keras tiene una interfaz simple y consistente optimizada para diferentes casos de uso. Proporciona información clara y procesable sobre los errores del usuario.

Modular y configurable: Los modelos en Keras se fabrican conectando bloques de código configurables entre sí, con pocas restricciones.

Fácil de extender: Escribir bloques de código personalizados para plasmar nuevas ideas que surjan de investigación permite crear nuevas capas, métricas y desarrollar modelos de estado-del-arte.

Particularmente estas herramientas dan soporte a la librería de agentes de RL llamada KerasRL [136] de la cual se introducirá con más detalle en el Capítulo 7 durante una comparación del desempeño de agentes con el framework presentado en este trabajo.

Pytorch

Pytorch es un framework de código abierto, punto a punto para el aprendizaje automático creado por Facebook en el 2016. Este framework está basado en la librería de código abierto Torch. Particularmente el paquete para python de esta librería brinda dos funcionalidades principales:

¹18/01/2021

- Computación utilizando tensores de forma similar a la librería Numpy pero acelerados por GPU.
- Construcción de redes neuronales en base a un sistema propio de graduación automática.

Se diferencia de otras librerías por este último punto. Su método de contracción de redes neuronales permite la modificación de las mismas de forma arbitraria sin generar sobrecargas, a diferencia de las otras librerías o frameworks que suelen manejar una visión más estática de sus redes.

Particularmente este framework da soporte a la librería de agentes de RL llamada Stable Baseline 3 [137] de la que también se introducirá mayor detalle en el Capítulo 7 durante una comparación del desempeño de agentes con el framework presentado en este trabajo.

6.2. Historical Framework

Aplicando la estandarización de entornos introducida por GYM, en este trabajo se presenta el framework Historical Reinforcement Learning². Este framework brinda herramientas para la simulación de software configurable utilizando datos históricos.

El diseño del framework puede observarse en el diagrama UML de clases presente en la Figura 6.1. En esta Figura, se representa en la parte superior la relación agente-simulador mediante el conocimiento de la clase *Agent* y la interfaz *Env* implementada por la clase *Historical Enviroment*. Esta clase se integra con:

- Los procesos de manejo de acciones, delegando en la jerarquía que surge de la clase *AbstractActionHandler*.
- El manejo de recompensas para el proceso de aprendizaje, modelado por la clase *AbstractRewarder*.
- El simulador de la estrategia de ludificación y al procesamiento de datos históricos a partir de la clase *AbstractSimulator*.

Cada uno de estos elementos se detallarán en las subsecciones siguientes, así como la interacción entre ellos y el flujo de control introducido por el framework junto con sus puntos de extensión.

²El código y los casos de estudio del framework pueden encontrarse en: <https://github.com/jonx18/Historical-RL>.

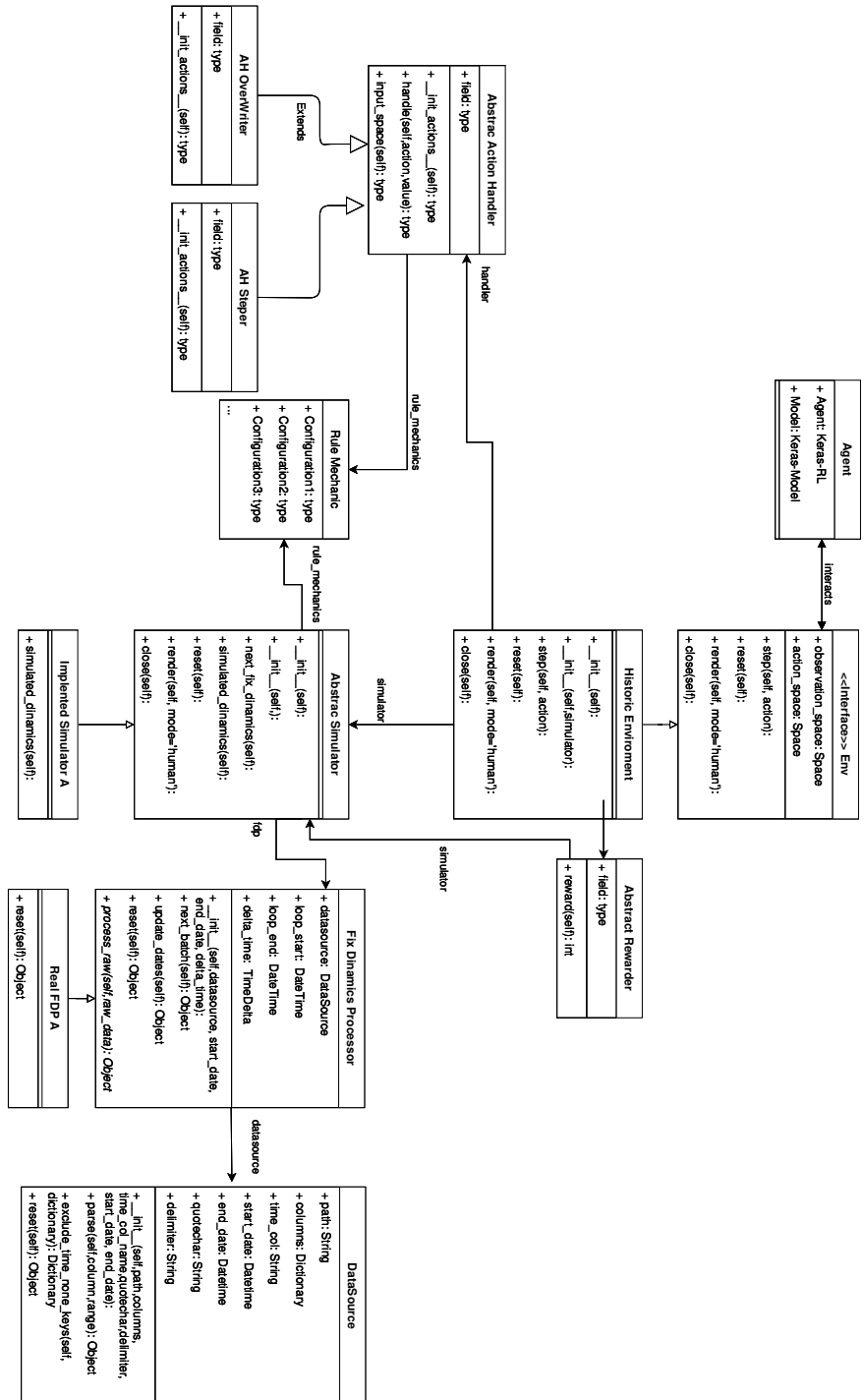


Figura 6.1: Historical Framework

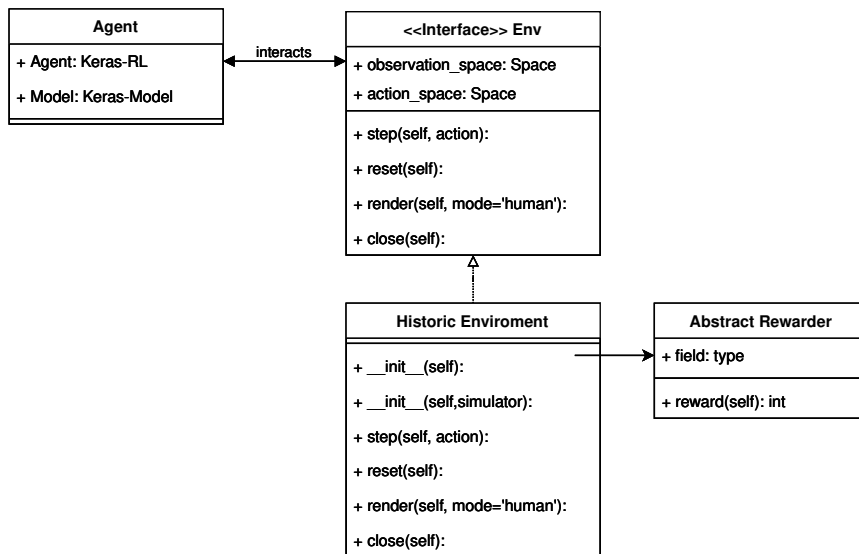


Figura 6.2: Relación entre las clases Env y Historical Enviroment

6.2.1. Historical Enviroment

A fin de respetar el estándar propuesto en GYM, se requiere estructurar el framework cumpliendo el uso de las interfaces que este brinda. Por lo cual, en la Figura 6.2 se puede observar que la clase *Historical Enviroment* implementa la interfaz *Env* de GYM. Esta es la clase de partida contra la que interactuarán los diversos agentes de RL.

La clase *Historical Enviroment* concentra la lógica relacionada al aprendizaje por RL. El objetivo de esta clase es separar la lógica del entorno relacionada con el aprendizaje, de la lógica del entorno sobre el que se desea aprender. Está compuesta a su vez por un simulador que representa la estrategia de ludificación a configurar, un manejador de acciones, y una abstracción de la estrategia de recompensa.

Como parte de las responsabilidades de este elemento están las de recuperar un espacio de observaciones para informar al agente. En este caso las observaciones se componen de una parte totalmente ligada al simulador y su estado interno, de la cual se le delega al simulador su generación. Junto con otras dos partes, una referente a la configuración del ciclo de aprendizaje, comúnmente utilizada para brindar información de depuración o informar de la finalización del mismo de requerirse, y otra relacionada con la recompensa.

Para el cálculo de recompensa se necesita evaluar cuando se considera que la simulación sea exitosa. La definición de éxito de una simulación puede variar de una simulación a otra, o pueden querer intercambiarse para comparar el comportamiento de distintos agentes. Con este fin se implementa

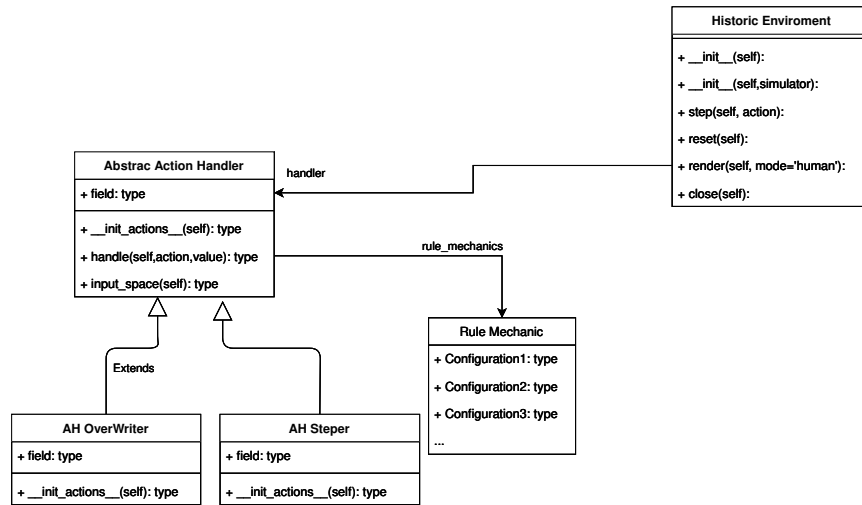


Figura 6.3: Jerarquía para el manejo de acciones

la función de recompensa como una implementación del patrón de diseño Strategy en la clase *AbstractRewarder*. La clase define el método abstracto `#reward() : int` que cual retorna un entero positivo que representa la puntuación de recompensa sobre la simulación actual, este valor se intentara maximizar por parte del agente. Para realizar dicho calculo además se posee una referencia al simulador a fin de tener acceso a información sobre como funcionó la configuración actual.

6.2.2. Manejo de acciones

El manejo de acciones consiste en generar y exponer las acciones que puede realizar el agente sobre la configuración del simulador, así como también las acciones del agente y procesarlas para que tengan impacto en la configuración. La clase *Historical Enviroment*, como se observa en la Figura 6.3, delega esta tarea en la jerarquía encabezada por la clase *AbstractActionHandler*, esta jerarquía también implementa el patrón Strategy a fin de admitir diferentes formas del manejo de acciones.

Estas acciones modificaran la configuración del simulador, por ejemplo, incrementando o decrementando el valor de alguna variable. Por lo cual dicha configuración queda separada del simulador en forma de valores encapsulados en la clase *RuleMechanics*.

La clase *RuleMechanics* encapsula un listado de variables que representan los valores actuales de los campos de configuración, así como también información sobre el tipo y rango de valores admitidos, además de un conjunto de funciones para el manejo de las mismas. En esta clase se brindan funciones y variables de ejemplo que permitan cubrir a mayoría de las posi-

bles opciones de configuración.

Debido a la amplia posibilidad de configuraciones que se pueden definir en las variables, se necesita poder definir como se irían modificando las mismas y como esas modificaciones se mapean con cada una de las acciones que forman el espacio de acción. De estas tareas se ocupan las subclases de *AbstractActionHandler* las cuales definen como se mapea cada acción sobre una variable de configuración a una acción individual para el agente.

Este mapeo de variables a acciones depende de cómo se decida realizar la manipulación de las mismas, a continuación, se listan dos ejemplos basados en las clases brindadas por el framework:

- *AHOverWriter* representa un manejador de acciones centrado en sobrescribir los valores de configuración. Con este fin utiliza dos parámetros, el identificador de la acción y el valor a asignar de la misma. Al tener una función que sobrescribe el valor de la configuración con un valor específico se obtiene una única acción para cada variable de la configuración.
- *AHStepper* representa un manejador de acciones el cual evalúa modificaciones de la configuración en forma de pequeños cambios. Para lograr estos cambios se asigna cada configuración a una acción de incremento y una de decremento en una unidad fija definida en la instanciación (por defecto la unidad mínima de distancia).

6.2.3. Abstracción del simulador

La abstracción del simulador se puede observar en la Figura 6.4 en la clase *AbstractSimulator*. Esta clase es el punto de anclaje de los distintos simuladores que se deseen configurar, conteniendo a su vez una relación con el modelo de objetos de la aplicación a simular.

Además, define el ciclo de ejecución de la simulación en lo referente al procesamiento de datos. Dicho ciclo permitirá que se evalúe la simulación ya sea de forma completa o de forma escalonada en el tiempo a lo largo del conjunto de datos históricos recibiendo los datos fijos y datos dinámicos.

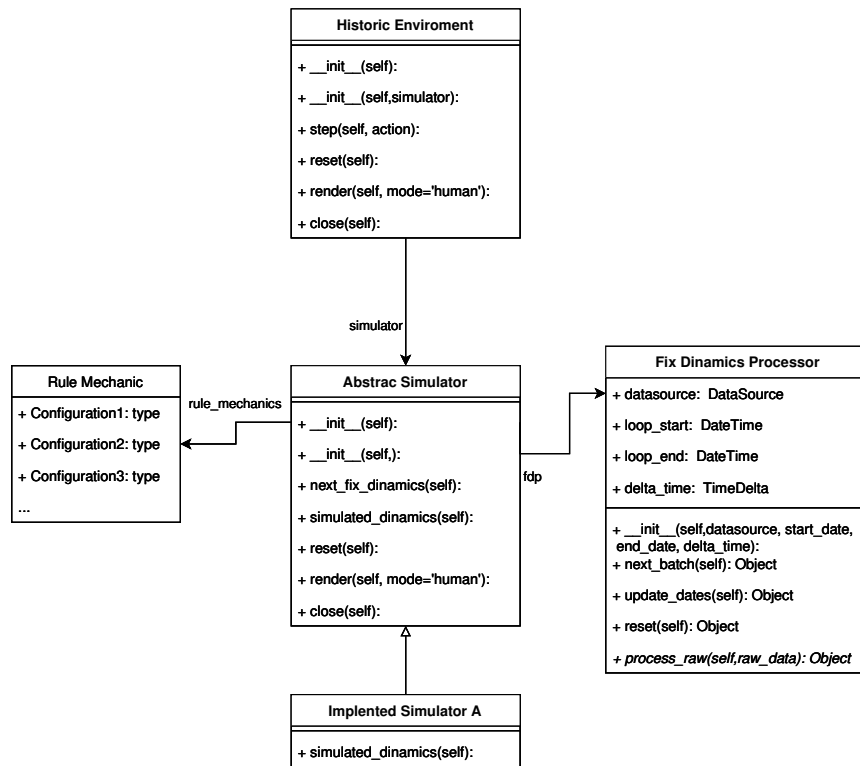


Figura 6.4: Elementos del simulador.

Los datos fijos son los que no requieren procesamiento relacionado con las configuraciones o rule mechanics, estos datos se obtienen del componente *FixDinamicsProcessor* del cual se hablará posteriormente, pero es relevante mencionar que son datos que cambiarán con menor frecuencia por su directa relación con los datos históricos sobre los que se corre la simulación.

Los datos dinámicos en cambio son datos que se generan a partir de aplicar las mecánicas relacionadas a la configuración y que por lo tanto variarán ante cada cambio de la misma. Estos datos se deberán actualizar en cada paso del aprendizaje y son los que definirán principalmente los resultados de la simulación. Además, los datos dinámicos por lo tanto dependerán mayoritariamente las mecánicas del simulador concreto y su relación con su configuración.

6.2.4. Procesamiento de datos históricos

El procesamiento de datos históricos con los cuales se alimentará la simulación se da en dos etapas. Una etapa de obtención de datos en crudo o sin procesar. Continuada de una segunda etapa relacionada con la creación de

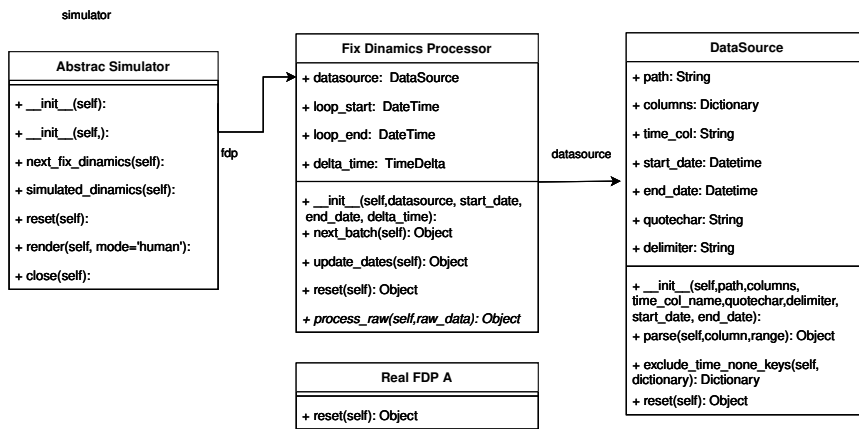


Figura 6.5: Manejadores de datos

componentes del modelo de datos. Estas etapas se relacionan principalmente con las clases que se pueden observar en el UML de la Figura 6.5.

La primera etapa se representa con la clase *DataSource*, la cual contiene la lógica de conexión con la fuente de los datos históricos (por defecto se brinda una implementación para carga de datos desde csv) y el desplazamiento temporal sobre los datos históricos. En esta etapa se prioriza la obtención rápida de datos del periodo requerido. Este es uno de los puntos que generan mayor carga de tiempo y recursos del aprendizaje.

La segunda etapa se da en la clase *FixDinamicsProcessor*, esta clase posee conocimiento del dominio del software de la estrategia de ludificación y transforma los datos en crudo a objetos y clases del modelo mediante la reimplementación del método `#process_raw(raw_data) : Object`. Por ejemplo, en el contexto de Wikipedia, los datos retornados por este mensaje serian todas las instancias de jugadores y acciones que realizaron en el periodo de datos cargado. Por otro lado, otra responsabilidad del componente es la de control de iteración sobre los datos históricos permitiendo abstraer al simulador de la iteración propia del proceso de RL.

Esta forma de procesamiento de los datos busca minimizar la interacción de entrada/salida del simulador, dado que para la simulación por datos históricos este es un proceso costoso en tiempo y por lo cual impacta en gran medida en el tiempo de aprendizaje del agente.

6.3. Resumen del capítulo

En este capítulo se presentó un framework de RL llamado Historical Reinforcement Learning para la configuración de sistemas basados en simulaciones históricas. Este framework permite la interacción con diferentes

agentes de RL gracias a la implementación del estándar establecido por GYM, el cual junto con el resto de las tecnologías utilizadas para la implementación del framework son presentados al comienzo del capítulo.

Posteriormente se presenta el framework del cual se describen tanto sus componentes como las responsabilidades y capacidades de extensión que brindan cada uno. La aplicación y extensión de este framework se puede observar en los casos de prueba presentados en el Capítulo 7.

Capítulo 7

Casos de estudio

En este capítulo se evaluará el enfoque propuesto mediante casos de estudio implementados aplicando el framework Historical RL. Dichos casos de estudio se desarrollan en dos CCC, la edición inglesa de Wikipedia la cual se introdujo en capítulos previos, y Stack Overflow otra CCC basada en la colaboración mediante preguntas y respuestas. Estas CCC se utilizan a lo largo de tres casos de estudio.

El objetivo principal de los casos es validar que un enfoque de RL permite la obtención de configuraciones óptimas para una determinada estrategia de ludificación mediante el uso de datos históricos. Además, los dos primeros casos evalúan la generalización del enfoque en distintas comunidades apoyándose en el uso del framework a fin de minimizar la cantidad de cambios necesarios.

También, se evalúa el impacto del uso de datos históricos en las configuraciones obtenidas, así como aspectos relacionados con su efectividad en periodos de tiempo distintos al aplicado. El tercer caso, si bien se desarrolla sobre la comunidad de Wikipedia, tiene como fin evaluar la flexibilidad del enfoque apoyándose en el framework para trabajar con diferentes agentes de RL existentes a fin de validar la posibilidad de integrar futuros avances en el área de RL.

Durante la evaluación de la integración del framework a Wikipedia en el primer caso de estudio, se evaluarán dos subcomunidades que serán tratadas como comunidades diferentes. Una comunidad compuesta por los artículos de mayor calidad y otra por artículos pertenecientes una estrategia de ludificación interna de la propia Wikipedia llamada Wikicup.

A fin de describir el proceso de integración con el framework para evaluar el enfoque, se introducen los datos utilizados y la motivación con la que fueron seleccionados. Posteriormente se explican extensiones realizadas al

framework para esta integración. También se presenta la evaluación de la integración y finalmente se presentan los resultados obtenidos. El mismo proceso será abordado para el caso de la CCC Stack Overflow.

Finalmente, el capítulo concluirá con un caso de estudio que realiza una comparación entre agentes de configuración. Este caso de estudio tendrá como objetivo demostrar la flexibilidad del framework en su interacción con agentes de otros frameworks que proveen agentes de RL. Además, permitirá evaluar el desempeño de estos agentes. También se evaluará el impacto del estado y las recompensas comunicadas a los agentes durante el entrenamiento.

7.1. Wikipedia

Este caso de estudio retoma el ejemplo introducido en el Capítulo 4 de ludificar Wikipedia, pero desde la perspectiva del framework Historical RL. Para ello se utilizará la simulación de la estrategia de ludificación introducida previamente adaptada para su aplicación con el framework. En esta sección se introducirán las definiciones y modificaciones necesarias para la integración del framework y la estrategia de ludificación.

Luego se presentarán los materiales y métodos utilizados para la evaluación. Esto incluye la estrategia de ludificación analizada desde la perspectiva de MDA a fin de adecuarla al framework. También se describen los conjuntos de datos utilizados para las evaluaciones y los objetivos detrás de su selección. Finalmente se presentan los resultados y una breve discusión de los mismos.

7.1.1. Integración del framework

Para comenzar la integración del framework Historical RL, debemos iniciar trabajando en el simulador de la estrategia de ludificación en Wikipedia, el cual se alimenta usando información histórica. Esta información se extrae mediante el proceso introducido en el Capítulo 4 y se integra con el framework Historical RL acoplando el Wikicrawler y el Wikigamificator al *DataSource* del framework a fin de alimentar a la simulación de los datos históricos.

Además, para integrar el simulador al *Historical Environment* siguiendo la estructura presentada en el Capítulo 6, es necesario realizar algunas modificaciones de entrada y salida, como suele ocurrir en varias aplicaciones de aprendizaje de refuerzo, en las que es necesario especificar detalles del entorno simulado [105].

Como se muestra en el Capítulo 4, la estrategia de ludificación es configurada de forma manual, permitiendo modificar la cantidad de puntos neces-

rios para ganar insignias(badges) de cada tipo. Esto da lugar a que la primera modificación requerida sea permitir que esa configuración sea realizada de forma automática permitiendo realizar modificaciones en la configuración de forma independiente. Para lograrlo, se debe permitir una configuración parametrizada y lo suficientemente flexible para adaptarse a las diferentes CCC.

Para ello los puntos requeridos por insignia están definidos por la función $inputs(x)$ en 7.1. Esta función indica para cada nuevo nivel de insignia cuantas acciones extras del mismo tipo tiene que realizar el jugador para alcanzar ese nuevo nivel.

La función tiene un parámetro de entrada x que denota el nivel de insignia para la que se quiere conocer la cantidad de acciones nuevas necesarias desde el nivel previo, con $1 \leq x \leq \#max_badges$. El valor $\#max_badges$, es un valor que será configurable para el agente y representa el número máximo de insignias de cada tipo, siendo compartido entre los 4 tipos de insignias.

La función consiste en una función lineal de x con los valores m para la pendiente y b como ordenada al origen. Esta función representa la longitud de los pasos entre niveles. Solo con esta parte de la función se obtendría un incremento lineal de la dificultad, este tipo de dificultad puede generar estrategias de ludificación demasiado difíciles o que desmotiven la realización de esas actividades. A fin de evitar un incremento lineal de la dificultad de las insignias, la función lineal $(m \times x + b)$ es multiplicada por una función de oscilación que consiste en el seno(sin) de otra función lineal con los valores m_oc para la pendiente y b_oc como ordenada al origen. Esta función de oscilación por su parte representa la dirección en la que se dan los pasos entre niveles. Finalmente, se calcula un máximo para asegurar que el valor mínimo resultante sea mayor a cero ya que de otra forma se culminarían en ganancia automática de múltiples insignias.

$$inputs(x) = \max(\sin(m_{oc} \times x + b_{oc}) \times (m \times x + b), 1) \quad (7.1)$$

Los valores de estas variables serán seleccionados por el agente para cada tipo de insignia. Con estos valores definidos se podría aplicar una configuración similar a la presentada en el Listado 7.1.1. Aquí se observa una configuración para Metagame semejante a la presentada en el Capítulo 4.

También se presenta una implementación de la función $inputs(x)$ así como listados con posibles valores para las variables b, m, b_oc y m_oc los cuales son accedidos según el tipo de insignia y un único valor para la variable max_badges que es compartida por todas las insignias.

```

1 #Proyectos
2 projects= [
3   "Project_name",....
4 ]
5
6 #Funcion input
7 def inputs(x, m_oc, b_oc, m, b)
8   [1,Math.sin(m_oc*b_oc)*(m*b)].max
9 end
10
11 #Insignias
12
13 max_badges = 10
14
15 #Orden de los tipos de accion
16 #[Contribucion, Refuerzo, Diseminacion, Login]
17 m_oc_list = [5, 3, 4, 3]
18 b_oc_list = [5, 3, 4, 3]
19 m_list = [5, 3, 4, 3]
20 b_list = [5, 3, 4, 3]
21
22 projects.each do |pname|
23   proje = Project.create(name:pname)
24
25   (1..max_badges).each do |i|
26     #Insignias de Contribucion
27     Badge.create(name:"#{i}-contribution",
28       points:inputs(i, m_oc_list[0], b_oc_list[0], m_list[0], b_list[0]), project_id:proje.id,
29       badge_type:"contribution")
30
31     #Insignias Refuerzo
32     Badge.create(name:"#{i}-reinforcement",
33       points:inputs(i, m_oc_list[1], b_oc_list[1], m_list[1], b_list[1]),project_id:proje.id,
34       badge_type:"reinforcement")
35
36     #Insignias Diseminacion
37     Badge.create(name:"#{i}-disemination",
38       points:inputs(i, m_oc_list[2], b_oc_list[2], m_list[2], b_list[2]), project_id:proje.id,
39       badge_type:"dissemination")
40
41     #Insignias Login
42     Badge.create(name:"#{i}-login",
43       points:inputs(i, m_oc_list[3], b_oc_list[3], m_list[3], b_list[3]), project_id:proje.id,
44       badge_type:"login")
45   end
46 end

```

Listado 7.1.1 Codigo de configuracion de con Historical RL

Lo siguiente que se requiere es definir un espacio de observaciones que representara el estado del simulador. En este caso, está definido por el Conjunto *state* presente en 7.2, el mismo toma el valor de las variables que forman parte de la configuración, para representar el estado actual de la

configuración.

En este caso nuestras variables de configuración son b , m , b_{oc} y m_{oc} por cada clase de insignia lo cual son 16 variables y $\#max_badges$ que es una variable compartida entre todas las clases de insignias. También se incluye dentro del espacio de observación información del desarrollo de la simulación con la configuración actual, consistiendo en la cantidad de usuarios que alcanzaron cada rango. Debido al número de variante de jugadores según el fragmento de información histórica utilizada, la información de rango se da como el porcentaje del total de jugadores que pertenecen a este rango.

$$state = [\#max_badges, b, m, b_{oc}, m_{oc}, percentUsersInRank(1), \dots, percentUsersInRank(n)] \quad (7.2)$$

Donde: $percentUsersInRank(i)$ = retorna el porcentaje del total de usuarios que alcanzaron el rango i en la simulación (Visitante es el rango 1 y Ciudadano Científico Comprometido es el rango 5).

Mediante el *state* el agente podrá determinar las acciones a realizar en la siguiente iteración. Para ello requiere que se defina el espacio de acción sobre el cual podrá actuar. El espacio de acción se define mediante el conjunto de configuraciones y el manejador de acciones seleccionado.

En este caso se seleccionó como manejador el *AHSteper* por lo cual el agente puede incrementar o decrementar el valor de las configuraciones definidas en las *RuleMechanics* que son $\#max_badges$, b , m , b_{oc} y m_{oc} . Esta modificación se realiza en pasos individuales de valor 1 iterativamente para encontrar la configuración óptima para la estrategia de ludificación con un máximo de 320 cambios a fin de evitar periodos infinitos de configuración por parte del agente.

Para saber cuál es la mejor acción a realizar, el agente necesita una recompensa que signifique que la simulación fue correcta. En este punto es necesaria la definición de una medida de “correctitud” de la simulación la cual se correlaciona con el concepto de estética de MDA. Se definieron dos recompensas para esta evaluación lo cual se tradujo posteriormente en dos subclases de *AbstractRewarder*, una para el contexto de WikiCup y otra para el contexto de artículos destacados por Wikipedia.

7.1.2. Materiales y métodos

En este caso de estudio, los materiales para los experimentos se obtendrán como se mencionó anteriormente de la Wikipedia inglesa. La selección de los artículos de Wikipedia tomó dos enfoques diferentes. Uno basado en la replicación de una estrategia de ludificación preexistente en Wikipedia

conocida como Wikicup, y otro guiado por el interés de fomentar la generación de artículos de calidad usando artículos destacados. En ambos casos utilizando Metagame como estrategia de ludificación.

Escenario de WikiCup

WikiCup es una forma de ludificación existente en Wikipedia. Es un torneo de edición en el que los participantes colaboran en muchos artículos anotando puntos. La puntuación depende de la calidad del artículo, por ejemplo, si se participa en un “artículo destacado” se recibirán más puntos que si se participa en un “buen artículo”. El torneo está dividido en 5 etapas eliminatorias, donde en la última etapa el jugador con más puntos es el ganador. Los objetivos por los que se elige utilizar la WikiCup para entrenar al agente son:

- En primera instancia validar que un enfoque de Reinforcement Learning permite la generación de configuraciones óptimas para una determinada estrategia de ludificación, en este caso Metagame, en una determinada CCC, en este caso Wikipedia, manteniendo el comportamiento intrínseco de la comunidad.
- Validar que el enfoque es generalizable a diferentes CCC.
- Evaluar si a partir de los datos históricos del comportamiento de la comunidad también se puede captar y mantener comportamientos relacionados a una estrategia de ludificación distinta a la cual se pretende configurar, como sería el caso de Wikicup.

En base a estos objetivos, la función de recompensa para el contexto de WikiCup incrementara su valor cuando los jugadores que obtuvieron más puntos en la WikiCup obtengan mejores rangos en la simulación. La Función 7.3 representa esta recompensa donde se usa el rendimiento de los participantes de la Wikicup para medir el valor de sus comportamientos y evaluar el avance que obtienen en la simulación.

$$reward = \sum_{i=1}^5 i \times sumPoints(usersInRank(i)) \quad (7.3)$$

donde:

$usersInRank(i)$ = retorna el conjunto de usuarios que alcanzaron el rango i en la simulación (Visitante es el rango 1 y Ciudadano Científico Comprometido es el rango 5).

$sumPoints(U)$ = retorna la suma de los puntos obtenidos en la Wikicup por los usuarios en el conjunto U .

Esta recompensa se aplicó con un conjunto de datos obtenidos de la Wikicup 2018 mediante el proceso introducido en el Capítulo 4. El conjunto de datos obtenido tiene 549 artículos y 1.522.011 acciones. También se obtuvo la lista de participantes con sus puestos y puntos obtenidos durante la WikiCup, ya que los mismos se requiere en la función de recompensa.

Escenario de artículos destacados

Los artículos destacados se consideran los mejores artículos que ofrece Wikipedia, según lo determinan los editores de Wikipedia¹. Dado que solo el 0.01 % de los artículos alcanza esta categoría, puede decirse que un artículo destacado ejemplifica el mejor trabajo de los editores y se distingue por las normas profesionales de redacción, presentación y fuentes de referencia. También se los distingue con un icono de una estrella de bronce como se puede observar en la Figura 7.1.

La función de recompensa de los artículos destacados, presentada en la Función 7.4 está orientada a mejorar la colaboración en las CCC buscando conservar las buenas prácticas aplicadas en los artículos destacados de Wikipedia.

La recompensa guía al agente a buscar configuraciones que premien las acciones de refuerzo sobre las de contribución, y las de diseminación sobre ambas considerando priorizar la colaboración y la obtención de colaboradores. De esta forma se premian las acciones que se consideran relevantes para una CCC, al mismo tiempo que se trabaja sobre un conjunto de datos de artículos destacados para aprender y conservar las mejores prácticas de la comunidad.

Por lo tanto, los objetivos planteados de la selección de los artículos destacados junto con esta función de recompensa son:

- Continuar con la validación del enfoque de Reinforcement Learning en este caso manteniendo el comportamiento intrínseco del desarrollo de artículos destacados. Particularmente mediante el uso de dos conjuntos de artículos destacados que se describirán a continuación con sus propios objetivos para el agente.
- Validar que el enfoque es generalizable a diferentes CCC, mediante la simulación de dos subcomunidades diferentes a la de Wikicup.
- Evaluar si a partir de los datos históricos del comportamiento de la comunidad también se puede captar y mantener comportamientos condicionados por la función de recompensa seleccionada a fin de guiar

¹Artículos destacados de Wikipedia: https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

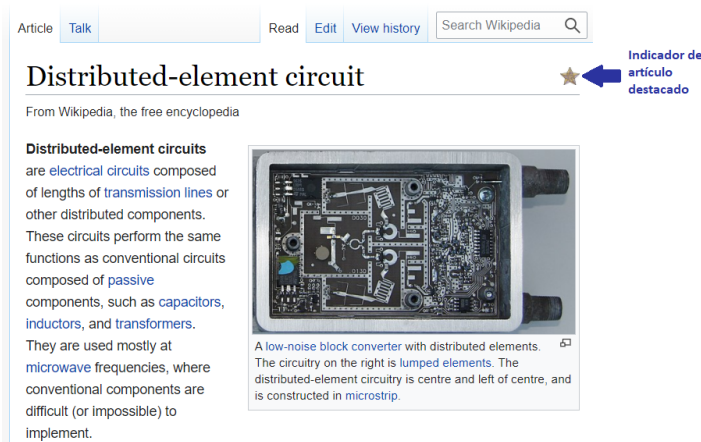


Figura 7.1: Artículo destacado con distintivo de una estrella de bronce.

a la comunidad en objetivos personalizados sin necesidad de que se cambie el comportamiento de la misma inicialmente.

$$reward = \sum_{u \in Users} rank(u) \times (ci(u) + ri(u) \times 5 + di(u) \times 10) \quad (7.4)$$

Donde:

U = usuarios que proporcionaron información para un artículo destacado.

$rank(u)$ = el rango (1..5) del usuario u en el diseño de la ludificación.

$ci(u)$ = recuento de las entradas de contribución del usuario u .

$ri(u)$ = recuento de las entradas de refuerzo del usuario u .

$di(u)$ = recuento de las entradas de diseminación del usuario u .

El conjunto de datos a utilizar se extrajo el 14 de febrero de 2019. En dicha fecha se encontraron 4540 artículos destacados, además de las páginas de discusión de esos artículos destacados, y páginas que contienen enlaces a estos artículos destacados. La cantidad total de datos obtenidos fue de 10.862 artículos, o 349.173.640 acciones de ludificación obtenidas mediante el mismo proceso descrito en el Capítulo 4.

Debido a la enorme cantidad de datos y al tiempo requerido para procesarlos, el conjunto de datos fue analizado buscando patrones para hacer una selección de artículos y reducir el conjunto de datos. Las variables seleccionadas para hacer este análisis fueron la edad del artículo, que es el número

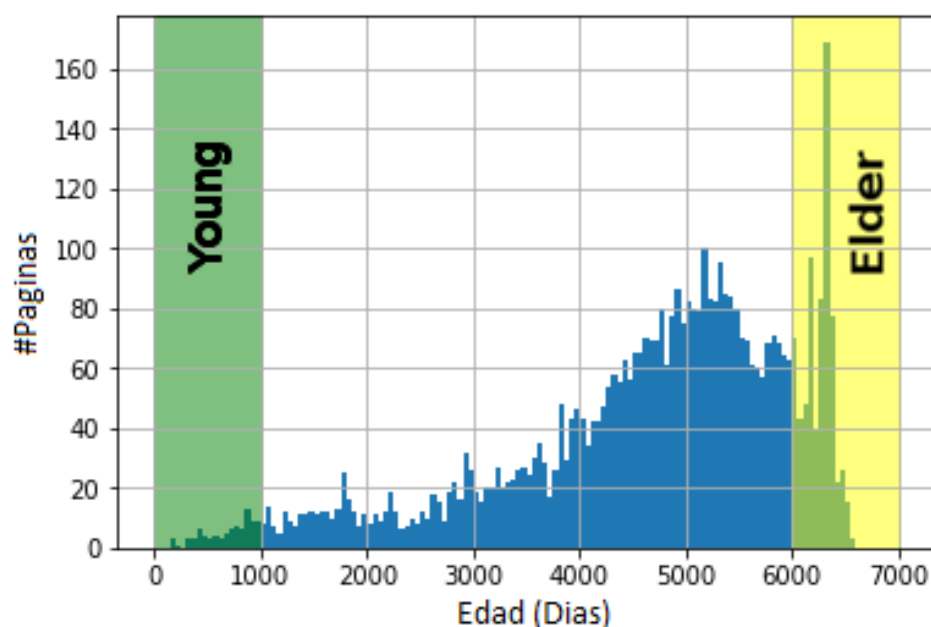


Figura 7.2: Distribución del tiempo de vida de los artículos destacados.

de días desde la creación del artículo, y el número de días como artículo destacado.

La Figura 7.2 se usó para analizar la edad del artículo. El eje X muestra la edad en lapsos de 50 días, y el eje Y muestra el número de páginas con esta edad. Por ejemplo, hay 70 páginas con una edad de entre 6000 y 6050 días. Se puede ver que a medida que la edad aumenta, el número de páginas tiende a aumentar también, llega a un tope entre las edades 5000 y 6500 días, o en otras palabras, en los artículos más antiguos (Elder). También, hay otro grupo de artículos que son jóvenes (Young) pero que aún comienzan a volverse destacados entre las edades menores a 1000 días.

El tiempo como artículo destacado se midió como los días entre la fecha en que el artículo fue marcado como destacado y la última edición sobre la edad del artículo. En la Figura 7.3, el eje X muestra el número de días como artículos destacados en lapsos de 50 días y el eje Y muestra el número de páginas con este tiempo como destacado. Por ejemplo, hay 80 páginas que llevan entre 3950 y 4000 días como artículos destacados. Además, el número de páginas para cada conjunto de días es más equilibrado, al menos hasta que llega a los 3800 días donde hay un incremento que alcanza el pico cerca de los 4000 días.

Finalmente, el cruce de esas métricas como se puede ver en la Figura

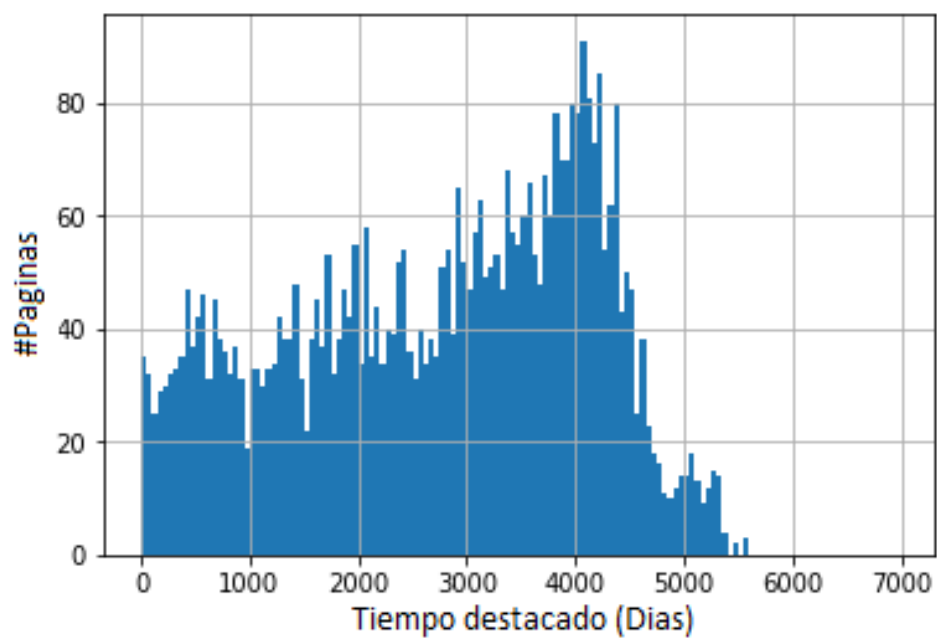


Figura 7.3: Tiempo como artículos destacados.

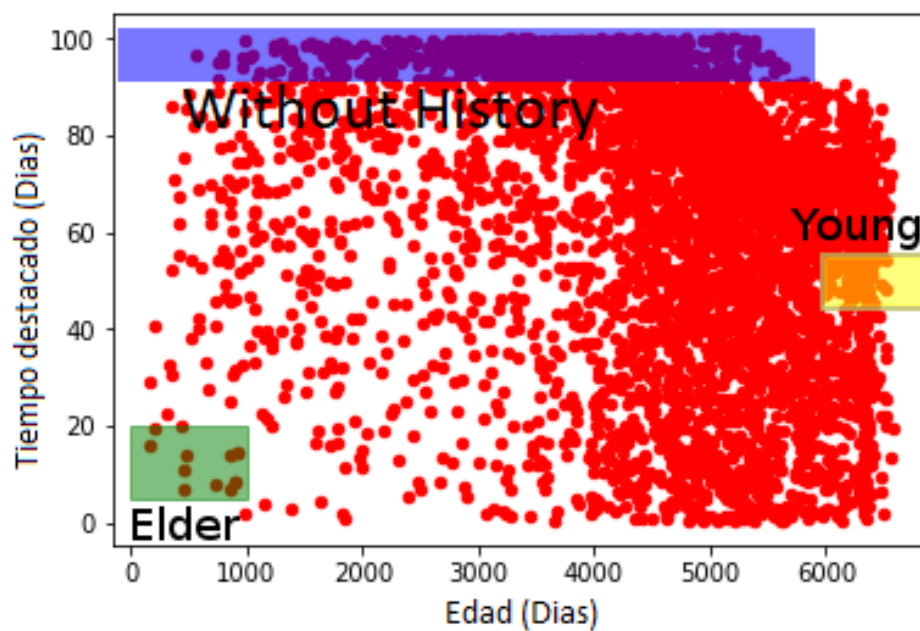


Figura 7.4: Artículos destacados, tiempo de vida y tiempo como destacados.

7.4 donde el eje X muestra la edad en lapsos de 50 días, y el eje Y muestra el porcentaje de esta edad como destacado, revelan que es posible ver tres separaciones. A partir de estas separaciones, se obtuvieron tres grupos.

Primero, un grupo sin historial de cambios, el cual se caracteriza por que desde el primer cambio en el historial hasta el momento de este estudio fue siempre o casi siempre considerado como artículo destacado. Este grupo se representa en azul y con la etiqueta “Without History”. Esos artículos fueron creados a partir de artículos nominados o ya marcados para ser destacados. Sin embargo, al obtener la nominación o la marca de destacados propiamente dicha, estos artículos fueron mudados a una nueva página por ejemplo mediante un cambio en el título. En consecuencia, se pierde toda la historia de cambios de su vida como artículos no destacados. Esos artículos no fueron considerados útiles para este estudio dado que carecen de la información histórica de interés para esta evaluación.

Para poder entrenar el agente de forma que logre promover el proceso de creación de artículos de estacados, se analizó artículos que llevan como destacados un tiempo de entre el 5% y 20% del total de tiempo desde su creación, y con una edad inferior a los 1000 días. Este grupo representa los artículos más nuevos que fueron destacados de una manera rápida, y se espera que este grupo muestre cómo lograr el nivel de un artículo destacado de una manera rápida.

También se revisaron artículos que llevan como destacados un tiempo de entre el 45%-55% del total de tiempo desde su creación, y con 6000 o más días de edad. Este grupo representa artículos con un tiempo equilibrado entre el antes y después de comenzar a ser destacados, por lo cual se espera que tenga información no sólo de cómo lograr la calidad de destacado, sino también, cómo podría ser conservada. Con lo cual se esperaría que el agente además aprenda a promover tanto la creación como el mantenimiento de artículos destacados. Ya con ambos conjuntos de datos se entrenó al agente para obtener configuraciones óptimas de la estrategia de ludificación para cada uno.

De los grupos, el segundo y el tercer grupo se conservaron, se pueden interpretar como un grupo de artículos jóvenes (marcados como Young y con un área verde) y uno de artículos antiguos (marcados como Elder y con un área amarilla) respectivamente en las Figuras 7.2 y, 7.4.

7.1.3. Resultados

Esta sección presenta los resultados de la simulación a partir de las configuraciones propuestas por el agente después de aprender de las diferentes comunidades con sus respectivas funciones de recompensa.

Tipo de acción	# <i>max_badges</i>	Oscillation		Steps variables	
		<i>m_oc</i>	<i>b_oc</i>	<i>m</i>	<i>b</i>
Contribución	11	5	5	6	12
Refuerzo	11	3	6	0	6
Diseminación	11	4	2	2	0
Login	11	3	3	2	1

Tabla 7.1: Configuración de variables de la función $input(x)$ para WikiCup.

Escenario de Wikicup

El agente de RL se entrenó utilizando el conjunto de datos de la Wikicup 2018, y la función de recompensa detallada en la Función 7.3, durante 100 episodios. El agente recomendó una configuración con 11 insignias por tipo, distribuidas en base a los valores de la Función $input(x)$ definida en 7.1, con los parámetros de la Tabla 7.1.

En la Figura 7.5 se observa ver el número de acciones nuevas requeridas para avanzar entre los niveles de las insignias, permitiendo al cumplir esa cantidad, alcanzar el siguiente nivel. Por ejemplo, en las insignias de Contribución representadas con color naranja, se puede observar en esta línea que a la altura del nivel 2 se obtuvo un valor mayor de 10. Esto representa que para llegar al nivel dos desde el nivel 1 se necesitan más de 10 nuevas acciones de Contribución. Además, podemos ver como el agente propuso alternar fases de alto esfuerzo con fases de bajo esfuerzo. Esto significa que el esfuerzo del jugador se divide en periodos de trabajo duro y periodos de asimilación de habilidades adquiridas, permitiendo un flujo equilibrado en la actividad del editor.

Basándose en los valores de nuevas acciones requeridos entre niveles, la Figura 7.6 muestra la escala de dificultad por cada nivel de insignia. En el eje X están los diferentes niveles y en el eje Y está la acumulación de acciones requeridas. Finalmente, cada tipo está representado por diferentes colores. Por ejemplo, para alcanzar el nivel máximo de la insignia de contribución, también en color naranja, se solicitan 306 acciones de contribución. En esta figura, es posible ver una importante diferencia entre la evolución de la insignia de contribución en comparación con las otras insignias; las insignias de contribución se vuelven más desafiantes, dado que se requieren más acciones por cada nueva insignia a un ritmo más rápido. Presumiblemente, es porque estas acciones son el principal objetivo de los jugadores en el WikiCup.

Esta configuración de dificultad define la forma en que se entregan las insignias, esto se puede ver en la Figura 7.7. En la figura cada línea representa un tipo de acción donde el valor del eje X representa el tiempo y el eje Y la acumulación de las insignias entregadas hasta determinado día. Por ejemplo,

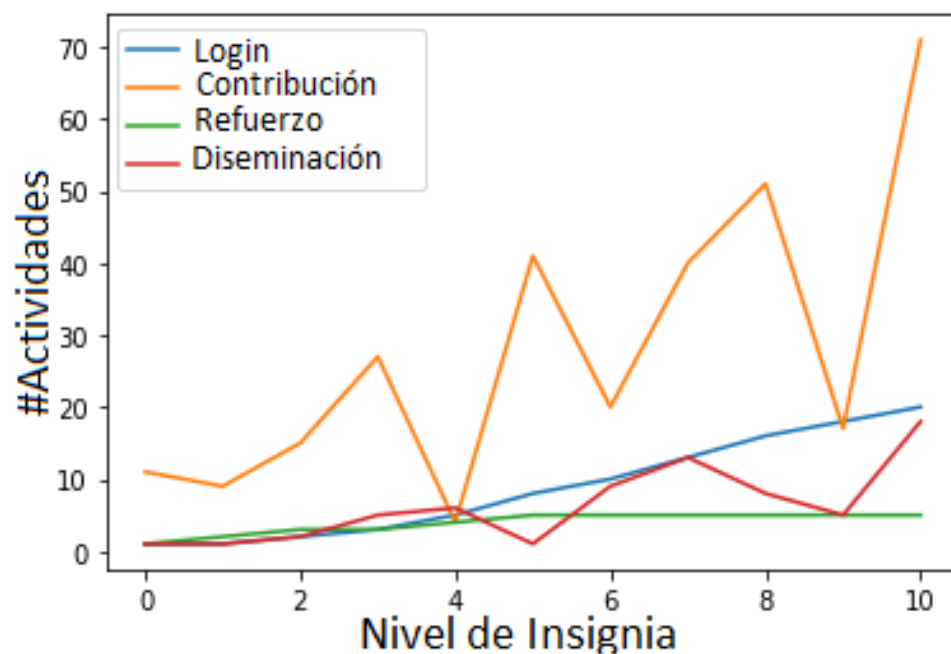


Figura 7.5: Cantidad de acciones requeridas para obtener un nuevo nivel de insignia en Wikicup.

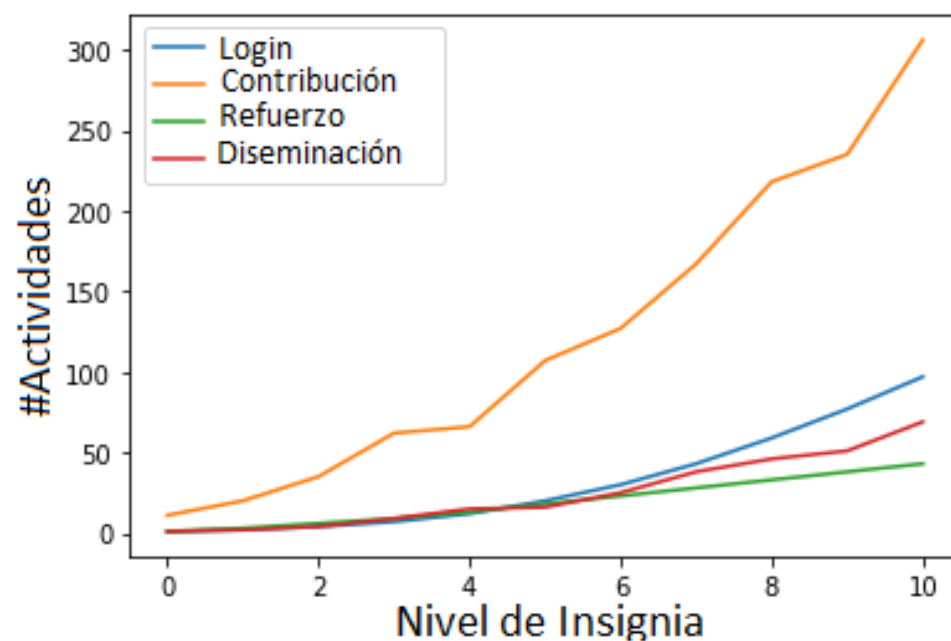


Figura 7.6: Escala de dificultad para WikiCup.

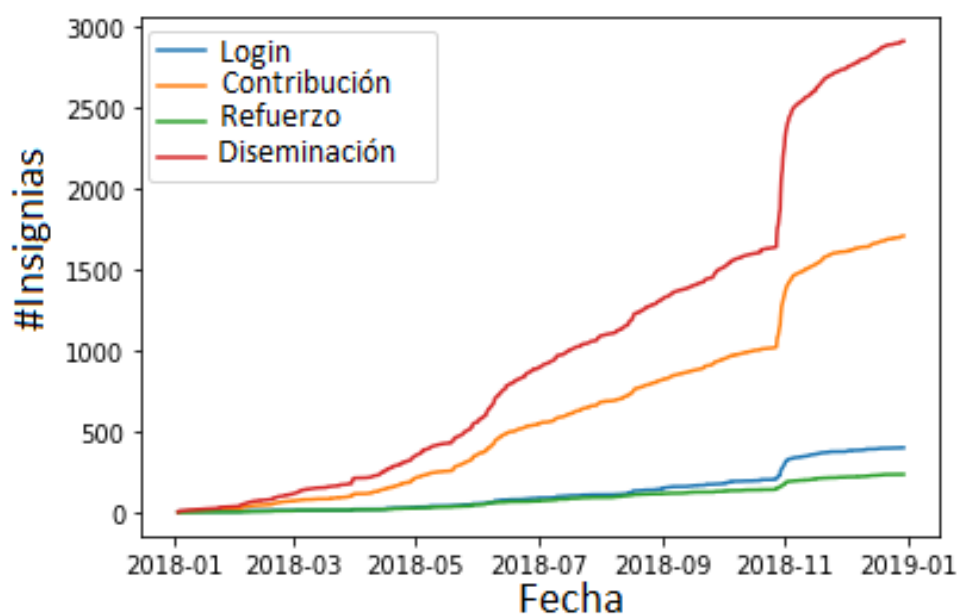


Figura 7.7: Insignias entregadas a lo largo del tiempo para WikiCup.

Jugadores			
Rango 1	Rango 2	Rango 3	Rango 4
79.46 %	12.2 %	3.15 %	0.35 %

Tabla 7.2: Porcentaje de jugadores por rango en Wikicup.

en noviembre de 2018 se entregaron más de 1500 insignias de contribución. Como se puede observar, las insignias de Diseminación y Contribución son entregadas con mayor frecuencia. De esta forma, el agente centra sus esfuerzos por un lado en las principales acciones de la Wikicup, las acciones de Contribución, para centrarse en los mejores participantes y por otro lado en las insignias más difíciles de ganar en esta simulación, las de Diseminación, a fin de permitir a los jugadores avanzar en los rangos.

Estas figuras y sus resultados ya nos permiten observar que el agente logra adaptar la configuración mediante el enfoque de RL a fin de permitir el ajuste de la estrategia de ludificación. Además, este ajuste se realiza sobre datos históricos que no varían permitiendo conservar el comportamiento de la comunidad, siendo la estrategia de ludificación la que se adapte a la comunidad y no la comunidad a la estrategia.

La simulación con esta configuración da como resultado la siguiente distribución de jugadores por rangos como se observa en la Tabla 7.2. Resultando en una distribución que coloca a significativamente más jugadores

en los rangos inferiores, y sólo unos pocos en los superiores. Además, sólo se muestra hasta el rango 4; esto se debe a que los 30 días de actividad a partir del rango 5 eran difíciles de mantener por los jugadores en los conjuntos de datos. Esto limita al agente, ya que esos días no forman parte de la configuración para adaptarlo.

Finalmente, en la Tabla 7.3 se muestra para los participantes de la WikiCup, el rango obtenido por la configuración, la puntuación total real obtenida y la última etapa alcanzada en la WikiCup 2018. Por ejemplo, el jugador “Courcelles” (ganador, con 8231) obtuvo el rango 3 en la simulación, y “SounderBruce” (finalista, con 2675 puntos) obtuvo el rango 4. Esto puede parecer contradictorio al principio; dado que mientras existen jugadores que obtienen rangos más altos que la mayoría de los jugadores, el ganador de Wikicup obtiene un rango más bajo que un finalista. Una razón para esto es que la estrategia de ludificación también valora acciones que no son tenidas en cuenta por la Wikicup (como las acciones de diseminación), y que la Wikicup por otro lado diferencia el puntaje en base a la calidad del artículo en el que se participa (mientras que la estrategia de ludificación no lo hace). Esto permite considerar a esta configuración como una aproximación adecuada a la configuración del Wikicup. Por lo cual podemos considerar que el enfoque de RL sobre datos históricos permite además captar y mantener comportamientos relacionados a una estrategia de ludificación distinta a la cual se pretende configurar, en este caso Wikicup.

Escenario de artículos destacados

El agente se entrenó utilizando ambos grupos de artículos (Jóvenes y Antiguos como se menciona en la Sección 7.1.2), y la Función 7.4 como recompensa durante 100 episodios. El aprendizaje del agente sobre los dos grupos presenta diferentes resultados sobre las posibles configuraciones, y por lo tanto la distribución de las insignias a lo largo del tiempo y los jugadores sobre los rangos fueron diferentes. Las Tablas 7.4 y 7.5, presentan las configuraciones para la estrategia de ludificación obtenidas por el agente con los conjuntos de datos jóvenes y antiguos respectivamente.

Las Figuras 7.8 y 7.9 muestran, por cada tipo de acción y nivel, el número de nuevas acciones de este tipo que se requieren para obtener una nueva insignia. Por ejemplo, para la insignia de contribución del nivel 2 en el grupo de los artículos antiguos se solicitan 3 nuevas acciones de contribución y en cambio de ese nivel al siguiente solo 1 una acción más.

Las Figuras 7.10 y 7.11 ayudan a entender el escalado de la dificultad para cada tipo de insignia en el grupo de artículos jóvenes y antiguos respectivamente. En las figuras cada línea representa un tipo de acción donde el valor del eje X es el nivel de la insignia y el eje Y representa la acumula-

Jugador	Rango	Puntaje de WikiCup	Etapa en WikiCup
SounderBruce	Rango 4	2675	Finalista
Zanhe	Rango 4	700	Ronda 4
Courcelles	Rango 3	8231	Ganador
Kosack	Rango 3	2069	Segundo Puesto
Czar	Rango 3	109	Ronda 3
Ruby2010	Rango 3	19	Ronda 2
Toreightyone	Rango 3	0	Ronda 1
Kees08	Rango 2	2681	Tercer Puesto
Iazyges	Rango 2	1824	Finalista
Ceranthor	Rango 2	1535	Finalista
Nova Crystallis	Rango 2	1505	Finalista
Numerounovedant	Rango 2	1179	Ronda 4
Argento Surfer	Rango 2	724	Ronda 4
HaEr48	Rango 2	610	Ronda 4
Farang Rak Tham	Rango 2	579	Ronda 3
Yashthepunisher	Rango 2	347	Ronda 3
Muboshgu	Rango 2	316	Ronda 3
Carbrera	Rango 2	288	Ronda 2
1.02 editor	Rango 2	80	Ronda 3
Wizardman	Rango 2	51	Ronda 2

Tabla 7.3: Rango y puntaje de jugadores en WikiCup.

Tipo de acción	$\#max_badges$	Oscillation		Steps variables	
		m_oc	b_oc	m	b
Contribución	10	5	3	0	5
Refuerzo	10	4	3	1	2
Diseminación	10	5	1	5	2
Login	10	0	2	1	1

Tabla 7.4: Configuración de variables de la función input(x) para el grupo de artículos destacados jóvenes.

Tipo de acción	$\#max_badges$	Oscillation		Steps variables	
		m_oc	b_oc	m	b
Contribución	10	5	4	1	2
Refuerzo	10	2	3	0	5
Diseminación	10	4	3	4	5
Login	10	2	1	1	2

Tabla 7.5: Configuración de variables de la función input(x) para el grupo de artículos destacados antiguos.

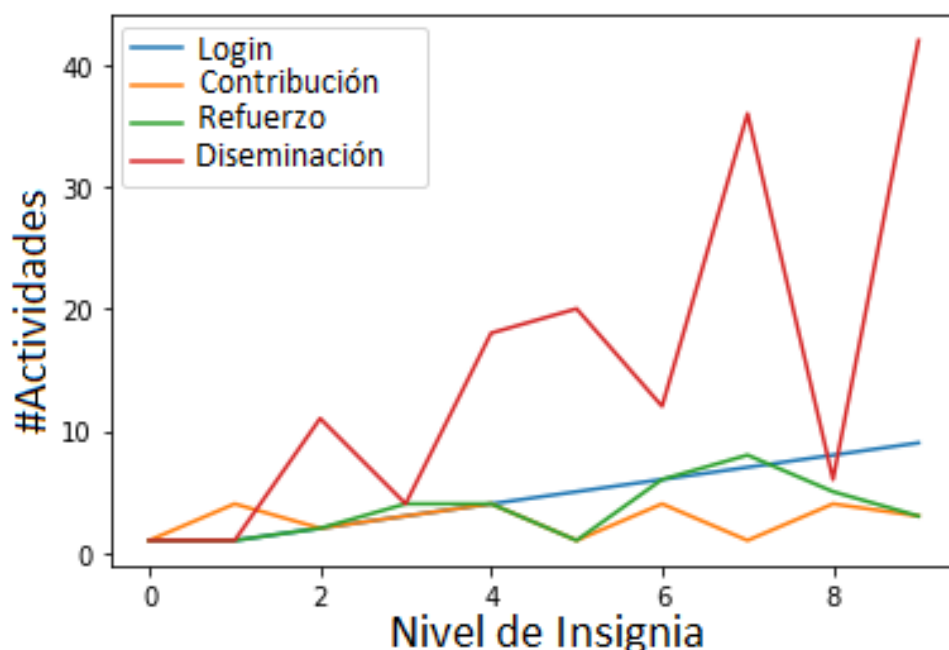


Figura 7.8: Cantidad de acciones requeridas para obtener un nuevo nivel de insignia en el grupo de artículos jóvenes.

ción total de acciones solicitadas para este nivel. Por ejemplo, para alcanzar el nivel máximo de la insignia de contribución en el grupo de jóvenes se solicitan 27 acciones de contribución en total.

Las Figuras anteriores conservan la idea de mantener flujo de trabajo equilibrado como hemos mencionado en el caso de Wikicup. Sin embargo, en el caso de los artículos jóvenes destacados, tiende a haber más linealidad en las dificultades con la excepción de las insignias de diseminación. Esto puede sugerir que el agente comienza dando más relevancia a las entradas de diseminación, y más tarde cambia su atención hacia las otras entradas.

Ese escalado de dificultad define la forma en que se entregaron las insignias en cada grupo. Esto se puede ver en las Figuras 7.12 y 7.13 para el grupo de artículos jóvenes y antiguos respectivamente. En las figuras cada línea representa un tipo de acción donde el valor del eje X representa el tiempo y el eje Y la acumulación de las insignias entregadas hasta un día determinado. Por ejemplo, en el 2017 el grupo de jóvenes alcanzará las 100 insignias de contribución entregadas.

Las Figuras 7.12 y 7.13 reflejan la importancia de las acciones de diseminación dada por la recompensa al agente; por lo cual, las insignias de diseminación escalan más rápidamente en dificultad y tienen más alternan-

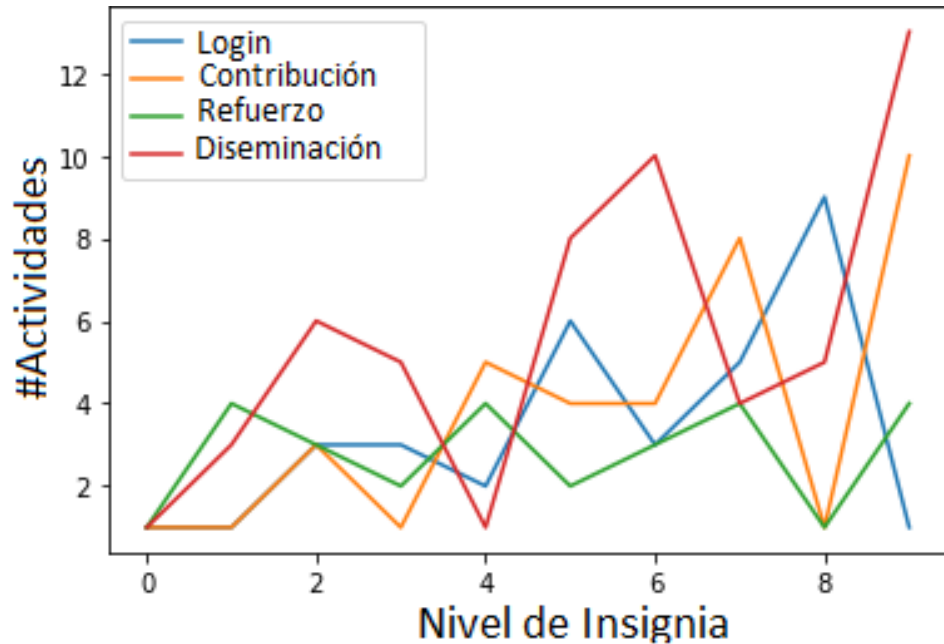


Figura 7.9: Cantidad de acciones requeridas para obtener un nuevo nivel de insignia en el grupo de artículos antiguos.

cia de dificultades. Además, la diferencia de dificultad entre las insignias de diseminación y las demás insignias es menor en el conjunto de datos antiguos. Esto se debe a que tiene más diversidad de acciones que permiten al agente hacer más difícil la adquisición de insignias, manteniendo al mismo tiempo un buen resultado en la clasificación de los jugadores.

Finalmente, la simulación con estas configuraciones da como resultado la siguiente distribución en rangos de los jugadores por grupo de artículos en la Tabla 7.6. Se puede señalar que en el conjunto de datos jóvenes hay una mejor distribución en los rangos que en el conjunto de datos antiguos; parece estar relacionado con el elevado número de insignias de diseminación obtenidos, a pesar de la gran dificultad que plantea para los mismos el agente. Porque esta insignia representa las acciones menos comunes en las actividades del editor, pero aun así son necesarias para alcanzar el tercer rango.

7.1.4. Análisis de resultados

En esta sección se presenta un análisis de los resultados obtenidos en la aplicación de Historical RL para optimizar la configuración para una estrategia de ludificación en diferentes partes de una misma CCC.

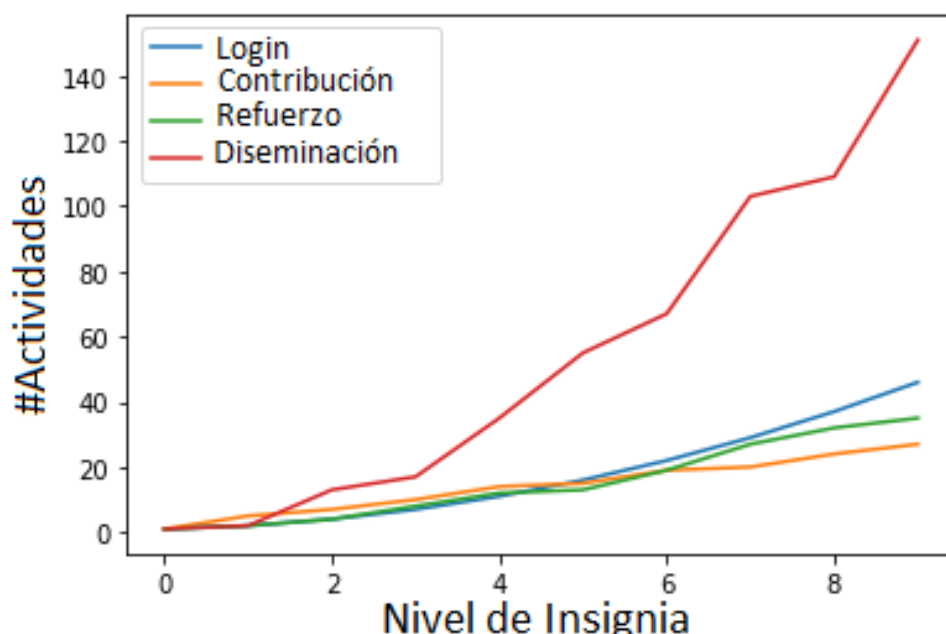


Figura 7.10: Escalado de dificultad en el grupo de artículos jóvenes.

	Jugadores			
	Rango 1	Rango 2	Rango 3	Rango 4
Joven	83.17 %	8.57 %	4.12 %	0.0 %
Antiguo	85.44 %	10.07 %	2.23 %	0.0 %

Tabla 7.6: Rango de jugadores en artículos destacados.

Se puede observar que los conjuntos de datos de artículos jóvenes y de la Wikicup son conjuntos de datos de períodos cortos, en los cuales se entregaron más insignias de diseminación que insignias de contribución. Sin embargo, el conjunto de datos de artículos antiguos, que es un conjunto de datos de un periodo de vida más largo, entregó principalmente insignias de contribución. Esto indica que en los comienzos de un artículo se considera más importante el uso de citas y la comunicación, mientras que en el largo plazo se centra en el incremento de contenido.

La función de recompensa aplicada con el dataset de los artículos destacados guió al agente a aprender a brindar mayor importancia a las insignias de diseminación mediante la configuración de la estrategia de ludificación. Esto se debe a que la función de recompensa prioriza esta actividad. Por otra parte, la entrega de insignias de refuerzo se mantiene en un equilibrio similar en ambos conjuntos de datos, pero se presenta un pequeño incremento de la dificultad para la obtención de insignias de contribución en el conjunto de

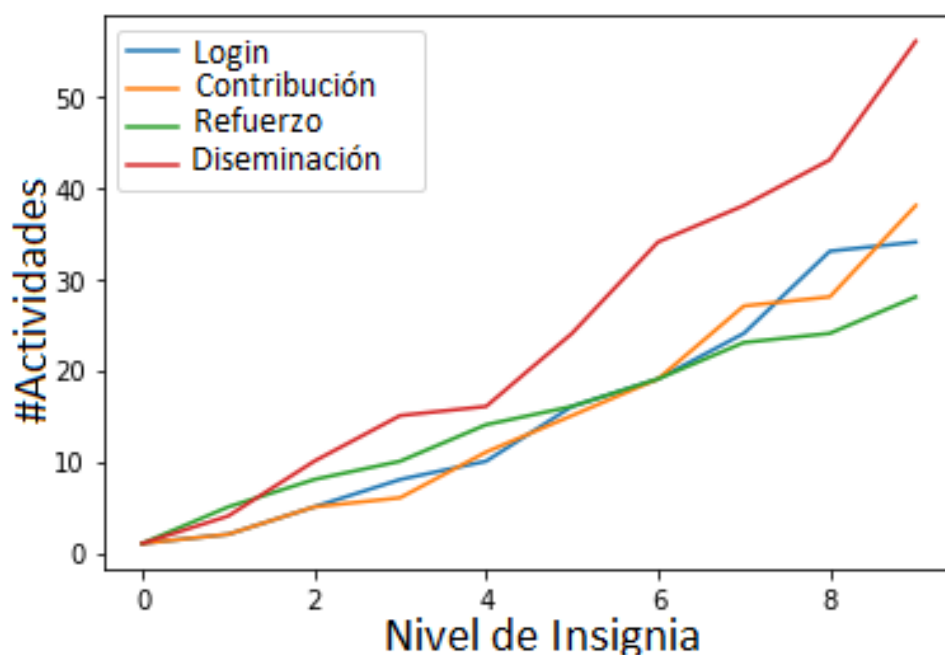


Figura 7.11: Escalado de dificultad en el grupo de artículos antiguos.

datos de largo período de vida.

Este balance de dificultad tiene sentido porque la función de recompensa pretende mejorar el aumento de participación y la capacidad de comunicación, basada en acciones de diseminación y refuerzo. Por lo tanto, la dificultad demuestra la importancia de las actividades de diseminación y ayuda a mantener equilibrado el número de insignias entregadas. También como se mencionó anteriormente, en el conjunto de datos de artículos antiguos hay más contribuciones desde el principio, por lo que el agente concluye en dar menos recompensa a ese tipo actividades e incrementar su dificultad.

En el caso de estudio de la WikiCup, el objetivo de la función de recompensa fue guiar al agente para adaptar la estrategia ludificación a la heurística de la WikiCup. Esta heurística se centra en un grupo de usuarios que hacen contribuciones a los artículos para ganar puntos, de esta manera la configuración encontrada enfatiza la dificultad en las insignias de las contribuciones. También, los bajos requerimientos en otros tipos de insignias apuntan a dar oportunidad de avanzar en los rangos de la ludificación a los participantes. Con esto el agente genera una configuración que ajusta la estrategia de ludificación para conservar las heurísticas de Wikicup con la mayor fidelidad posible dentro de las limitaciones mencionadas previamente. Por ejemplo, el hecho de que la estrategia de ludificación de Metagame

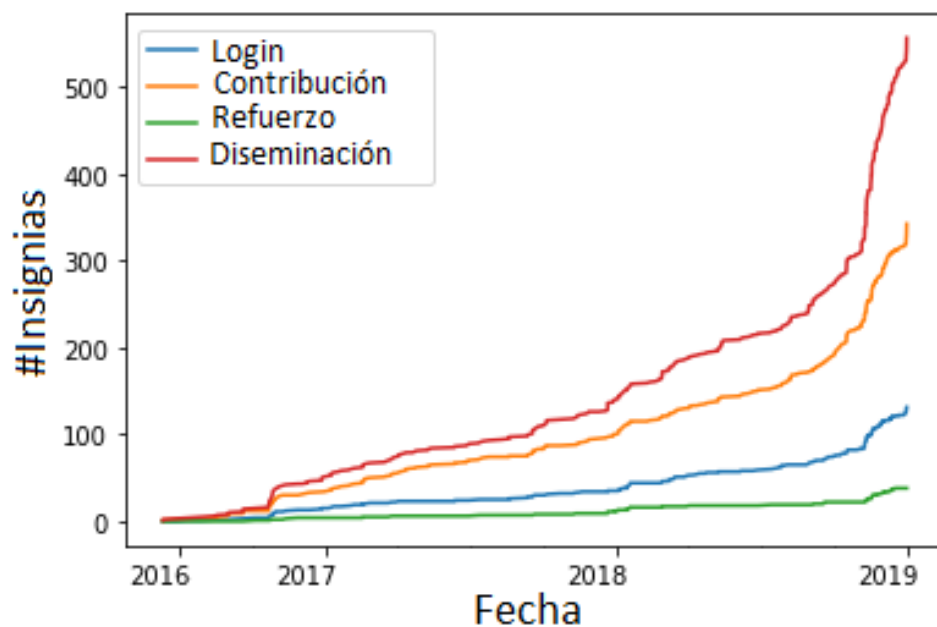


Figura 7.12: Insignias entregadas en el tiempo para los artículos jóvenes.

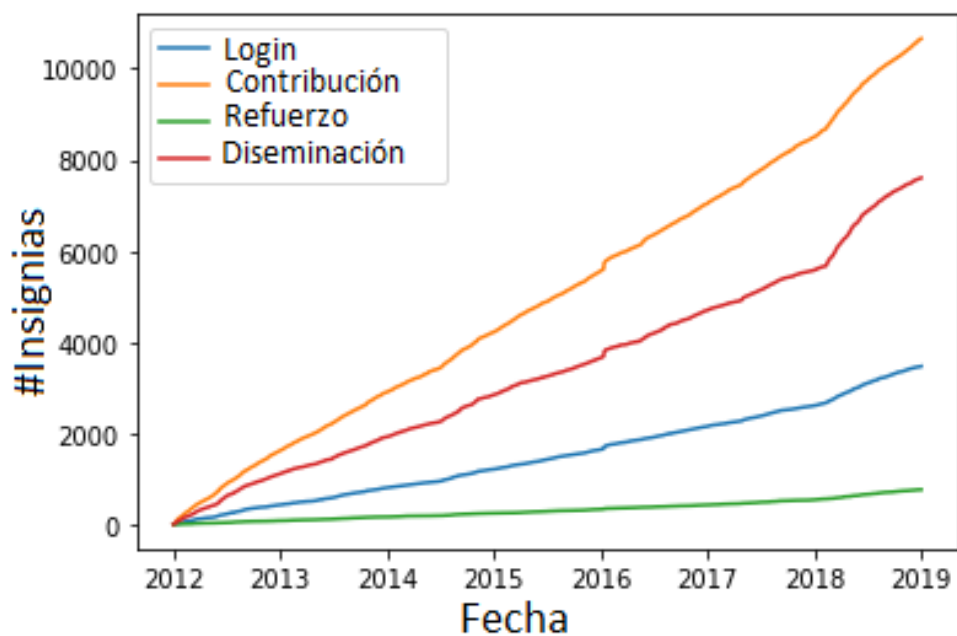


Figura 7.13: Insignias entregadas en el tiempo para los artículos antiguos.

	Jugadores			
	Rango 1	Rango 2	Rango 3	Rango 4
Manual	96.2 %	2.28 %	1.52 %	0.0 %
Jóvenes	83.17 %	8.57 %	4.12 %	0.0 %
Antiguos	85.44 %	10.07 %	2.23 %	0.0 %
Wikicup	79.46 %	12.2 %	3.15 %	0.35 %

Tabla 7.7: Comparativa de Rangos.

evalúa actividades de diseminación que no se contemplan en la Wikicup.

Como se muestra en la Tabla 7.6, los jugadores que ganaron más puntos o llegaron a las etapas más avanzadas en WikiCup también obtuvieron niveles más altos en la ludificación. Como los puntos se cuentan solo hasta que son desclasificados en Wikicup, hay jugadores que continúan con un comportamiento similar al de los mejores jugadores sin sumar puntos en WikiCup, pero en base a la configuración seleccionada consiguen un alto rango en la ludificación.

Un punto importante en una ludificación es la posibilidad de avance en ella por parte de los jugadores. En este punto la Tabla 7.7 muestra una comparativa de los rangos obtenidos con la estrategia presentada en este caso en contraste con la distribución de rangos en la configuración manual presentada en el Capítulo 4. En esta tabla se puede ver el incremento de jugadores en rangos más altos, esto significa que los jugadores son capaces de alcanzar rangos altos sin perturbar sus actividades típicas.

7.1.5. Conclusiones

Este caso de estudio presenta una forma de optimizar un diseño ludificado para recompensar los comportamientos deseados de los usuarios sin perturbar la funcionalidad de una CCC existente, en este caso Wikipedia.

Dentro de este caso de estudio se valida la aplicación del enfoque de Reinforcement Learning mediante el framework Historical RL el cual permitió la generación de configuraciones óptimas para la estrategia de ludificación seleccionada en la CCC de Wikipedia. Particularmente en las comunidades de la Wikicup y los artículos destacados. Esto se logró manteniendo el comportamiento intrínseco de la comunidad mediante su simulación.

Además, se aplicaron dos funciones de recompensa sobre las comunidades elegidas. Esto permitió demostrar cómo diferentes funciones de recompensa podrían optimizar la configuración para lograr diferentes objetivos sin distorsionar las actividades habituales de las comunidades. Por ejemplo, permitiendo que se transfieran heurísticas de una estrategia de ludificación como Wikicup a Metagame con sus limitaciones.

Finalmente, el uso de comunidades diferentes permite comenzar a validar que el enfoque sea generalizable a diferentes CCC. El siguiente paso hacia este objetivo se validará en el siguiente caso de uso donde se aplicará el enfoque a una CCC diferente a las de Wikipedia.

7.2. Stack Overflow

En este caso de estudio plantea validar la capacidad del enfoque de Reinforcement Learning para generalizar su aplicación a distintas CCC. Esto se desea realizar mediante el framework Historical RL, como en el caso de estudio anterior, pero para una CCC completamente distinta como es el caso de Stack Overflow. Para ello se utilizará la simulación de la estrategia de ludificación introducida previamente con el menor número de adaptaciones posibles. Como parte de este caso de estudio se introducirán las adaptaciones necesarias para la integración con la nueva comunidad.

Luego se presentará la metodología de evaluación para este caso de estudio. Esto incluye la definición del conjunto de datos y la forma de utilización de los mismos. Con estos datos se presentarán los resultados de las configuraciones obtenidas. Finalmente se presenta una breve discusión de los resultados y problemáticas halladas durante este proceso de integración.

7.2.1. Ludificación de Stack Overflow

Con el objetivo de simular una estrategia de ludificación sobre Stack Overflow con el framework Historical RL, se decidió aplicar Metagame como en el caso de estudio 7.1. Para permitir el acoplamiento de Stack Overflow al simulador debemos ser capaces de detectar las acciones que se relacionan con la estrategia de ludificación, obtener el historial de acciones y finalmente integrarlas al flujo de la simulación.

Este proceso se realizaba para el caso de Wikipedia mediante el uso del Wikicrawler el cual extraía los historiales de revisión, y el Wikigamificator que procesaba estos historiales mapeando las ediciones detectadas a acciones de la simulación. Durante la integración de la simulación con el framework, los procesos de recolección y mapeo se vincularon directamente con el *DataSource* como punto de entrada de los datos a partir del cual los mismo ya pasaban a ser acciones de la estrategia de ludificación. Esta abstracción brindada por el *DataSource* nos brinda un punto de entrada, sobre el cual para cambiar de CCC solo debemos redefinir el flujo de trabajo para la extracción y el mapeo de acciones que este contiene.

Adaptación del DataSource

Para realizar la adaptación del *DataSource* se requiere identificar: la fuente de datos brindada por la CCC a utilizar, el tipo de datos que pueden obtenerse de la comunidad, y el mapeo de los mismos a acciones de la estrategia de ludificación objetivo.

Stack Overflow, como parte de la red Stack Exchange, brinda acceso a una API de BigQuery para realizar consulta sobre la propia base de datos de la comunidad. Este método de consulta permite obtener información histórica de las acciones de la comunidad con las cuales alimentar al simulador. Si bien el modelo entidad relación² de los datos de la comunidad se encuentra disponible para consultas, en la Figura 7.14 puede observarse una simplificación del mismo. En esta figura se observan solo 5 de las tablas con las propiedades consideradas de relevancia para este caso de estudio.

La tabla *Users* contiene la información de los usuarios, particularmente de interés por la propiedad *DisplayName* que es el nombre con el cual el usuario se identifica y la propiedad *CreationDate* que indica la fecha de creación del usuario. La tabla *Post* representa tanto preguntas como respuestas dentro de la comunidad las cuales se diferencian mediante la propiedad *PostType*, luego también posee la propiedad *CreationDate* que indica la fecha de creación del post, la propiedad *OwnerUserID* que identifica al usuario creador, y finalmente la propiedad *Tags* la cual representa etiquetas que permiten clasificar en el caso de preguntas a que categoría o temática pertenecen. Esta última propiedad servirá además para delimitar un subconjunto de la comunidad de Stack Overflow con la cual trabajar.

Relacionado con los cambios históricos dentro de los posts, en la tabla *PostHistory* se representan todos los cambios que se realizaron para un post ya sea pregunta o respuesta el cual se identifica por la propiedad *PostID*, y está vinculada al usuario que realizó dicho cambio mediante la propiedad *UserID*. También posee las propiedades *CreationDate*, *Comments*, y *Text* que representan la fecha de creación del cambio, comentarios sobre el cambio y contenido del cambio respectivamente. Finalmente, durante un proceso de edición o creación por parte de un usuario pueden generarse múltiples registros, los cuales se crean bajo un mismo valor en la propiedad *RevisionGUID* y se diferencian entre sí según el tipo de cambio, el cual se define en la propiedad *PostHistoryType*³.

Los posts además pueden recibir comentario de los usuarios los cuales se consideran en la tabla *Comments* la cual tiene como propiedades *PostID*,

²<https://sedeschema.github.io/>. Visitado el 10/11/20.

³Un listado de todos los valores de *PostHistoryType* puede observarse en la columna revisión de la siguiente consulta:<https://data.stackexchange.com/stackoverflow/query/36599/show-all-types>. Visitado el 10/11/20.

UserID, *CreationDate*, y *Text* para representar el post objetivo, el dueño, la fecha de creación y el contenido del comentario respectivamente. Por último, la tabla *PostsLinks* representa la vinculación entre diferentes posts, que se realiza mediante la creación de enlaces. Estos enlaces guardan su fecha de creación, junto con los posts fuente y destino en las propiedades *CreationDate*, *PostID*, y *RelatedPostID* respectivamente.

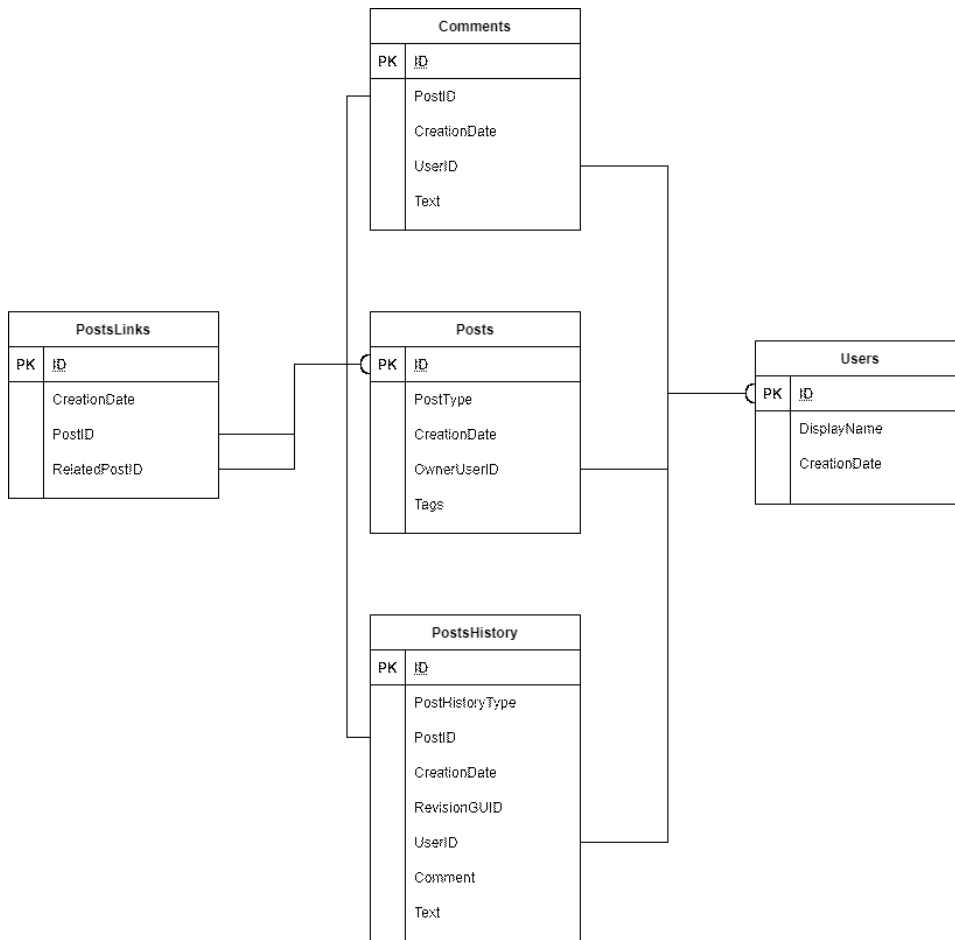


Figura 7.14: Diagrama entidad relación simplificado de Stack Overflow.

En base a estos datos, la Tabla 7.8 presenta el mapeo de acciones de Stack Overflow a acciones de la estrategia de ludificación. En la tabla cada columna representa una lista de acciones, la primera columna enumera las acciones que se consideran como contribuciones, la segunda las que se consideran como refuerzos y la tercera las de diseminación. Además, si bien no se menciona en el listado, como en el caso de Wikipedia se requiere un indicativo de actividad o presencia del usuario para representar acciones

Contribución	Refuerzo	Diseminación
Creación de preguntas	Edición de preguntas	Creación de tags
Creación de respuestas	Edición de respuestas	Edición de tags
Creación de comentarios	Rollback de preguntas	RollBack de tags
Protección de posts	Rollback de respuestas	Migración de posts
Bloqueos posts	Cambio de estados de los posts	Creación de enlaces
	Fusión de preguntas	Discusión de comentarios
	Realizar de sugerencias de cambio	Publicación en Twitter de posts
	Aplicación de sugerencias de cambio	Pregunta marcada como tendencia
		Pregunta resaltada por la comunidad de usuarios

Tabla 7.8: Eventos de Stack Overflow en el contexto de la ludificación.

de Login. En este caso, se decidió utilizar como indicadores la creación de comentarios y las primeras ocurrencias de la propiedad *RevisionGUID* de la tabla *PostHistory* dado que de esta forma se agruparía el conjunto de ediciones realizadas como una única actividad.

Este mapeo considera principalmente a las acciones de creación (crear preguntas, respuestas, comentarios) como acciones de Contribución, mientras que todo lo referente a ediciones, recuperación de datos, cambio de estados, o agrupaciones de contenido son considerado como Refuerzos al contenido ya existente. Finalmente, a diferencia del caso de Wikipedia, Stack Overflow presenta registro de sus herramientas de Diseminación como pueden ser publicaciones en Twitter, marcado de preguntas destacadas por la comunidad, y también se tuvo en cuenta todo lo que fuera interconexión interna de la comunidad como creación de enlaces o etiquetas.

Finalmente, este proceso de extracción de datos, y el posterior mapeo a acciones de ludificación será realizado por el *DataSource* permitiendo así la integración de Stack Overflow con nuestra simulación con el menor número de cambios posibles.

7.2.2. Metodología de evaluación

En este caso de estudio la información obtenida para la evaluación se tomó de la comunidad de usuarios del lenguaje de programación Python, de Stack Overflow. Para obtener esta información se filtró mediante la etiqueta (propiedad *Tags*) con el valor “Python” a todas las preguntas objetivo de las que se analizó el historial de cambios junto con los datos introducidos previamente.

De esta comunidad se tomaron todas las preguntas realizadas durante el 2018. Esto se traduce en 242.822 preguntas, más las respuestas, comentarios y los historiales de actividad que se traducen en 3.337.788 acciones de ludi-

ficación. Con este conjunto de datos se esperan validar los siguientes puntos del enfoque:

- Continuar validando, al igual que en el caso de uso anterior mediante la función de recompensa, que el enfoque de Reinforcement Learning permita la generación de configuraciones óptimas para la estrategia de ludificación, en la CCC de Stack Overflow, manteniendo el comportamiento intrínseco de la comunidad.
- Validar mediante la obtención de configuraciones que este enfoque es generalizable para diferentes CCC, en este caso una comunidad que genera piezas de conocimiento en forma de preguntas y respuestas en lugar de artículos como sucede en Wikipedia.
- Evaluar que las configuraciones obtenidas en un periodo anterior de la CCC posean continuidad en el tiempo a partir de conservar distribuciones similares a las observadas en periodos anteriores. Resultando estas configuraciones efectivas para periodos posteriores de tiempo en la misma comunidad.
- Comprobar si dichas configuraciones pueden también optimizarse mediante el mismo enfoque sobre nuevos datos para obtener mejores resultados en las funciones de recompensa y la distribución de insignias y usuarios en rangos.

Con estos objetivos en mente, el conjunto de datos se dividió en dos partes de 6 meses cada una representando la primera y la segunda mitad del año. Esta división se realizó con el objetivo de entrenar al agente en la primera mitad y evaluar su desempeño en la segunda mitad del conjunto para analizar la progresión de las configuraciones obtenidas por el agente en una misma comunidad más allá del periodo conocido por el mismo.

La evaluación a su vez se dividió a su vez en 2 etapas: Una etapa de entrenamiento del agente y obtención de configuración en el primer conjunto de datos; y una segunda etapa en el segundo conjunto de datos donde se consideraron 3 casos, uno evaluando la configuración obtenida del primer conjunto de datos en este nuevo conjunto, otro obteniendo una nueva configuración para este conjunto de datos utilizando el agente previamente entrenado en el primer conjunto de datos, y finalmente con etapas de entrenamiento cortas en este nuevo conjunto de datos para obtener una nueva configuración.

Todas estas evaluaciones se realizaron aplicando la función de recompensa utilizada para los artículos destacados del caso de prueba de la Sección 7.1 detallada en la Función 7.3. La elección de esta función se debió a que, como se mencionó anteriormente, la misma premia el comportamiento que se considera relevante en una CCC. Esta función también servirá de métrica

Tipo de acción	$\#max_badges$	m_{oc}	b_{oc}	m	b
Contribución	10	2	5	6	1
Refuerzo	10	3	9	7	7
Diseminación	10	14	19	14	13
Login	10	4	4	5	1

Tabla 7.9: Configuración para comunidad de python, primer semestre del 2018.

para evaluar las diferentes configuraciones encontradas por el agente durante las evaluaciones, a fin de facilitar la comparación sobre el rendimiento de una configuración, siendo que una mejor configuración será aquella que posea un valor más alto al evaluar la función de recompensa.

7.2.3. Resultados

Primer semestre

Durante el proceso de entrenamiento, el agente de RL se entrenó utilizando el primer semestre del conjunto de datos. El agente fue entrenado durante 100 episodios, obteniendo una puntuación máxima de 7.445.496 puntos otorgados por la función recompensa en combinación con el conjunto de datos. El agente recomendó una configuración con 10 insignias por tipo con los parámetros que se presentan en la Tabla 7.9. La configuración de insignias de diseminación recibe los valores más altos en las variables de configuración relacionadas a la oscilación (m_{oc} y b_{oc}) y al tamaño de los pasos (m y b) siendo estos más grandes que las otras insignias.

La Figura 7.15 muestra un análisis de la dificultad comparando el número de nuevas acciones necesarias para obtener cada insignia por tipo de acción. Los niveles de insignia están en el eje x , y el eje y indica el número de nuevas acciones necesarias para obtener una insignia. Por ejemplo, un jugador tiene que realizar 25 acciones de diseminación para conseguir el nivel 1 de la insignia Diseminación (la línea crece de 0 a 1), entonces el mismo jugador sólo necesita 7 nuevas acciones de diseminación para ascender al nivel 2 de Diseminación (la línea disminuye de 1 a 2).

Las oscilaciones en las líneas de la Figura 7.15 significan que el esfuerzo del jugador para conseguir una nueva insignia no es siempre el mismo. El jugador tendrá periodos de trabajo duro que requerirán muchas acciones y luego un periodo de asimilación de habilidades, en la línea de tener un flujo equilibrado como hemos mencionado anteriormente. Sin embargo, en el caso de las insignias de refuerzo, la configuración propuso tener un incremento lineal de la dificultad a lo largo del juego.

El escalado de la dificultad se puede ver en la Figura 7.16 que muestra

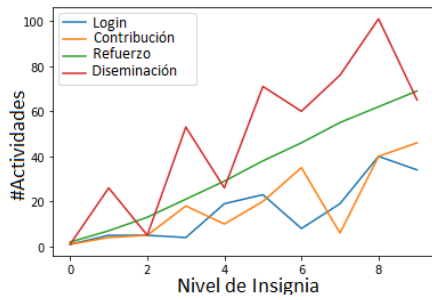


Figura 7.15: Primer semestre. Nuevas acciones necesarias para obtener cada insignia.

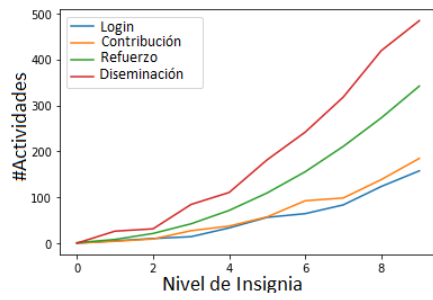


Figura 7.16: Primer semestre. Total de acciones necesarias para obtener cada insignia.

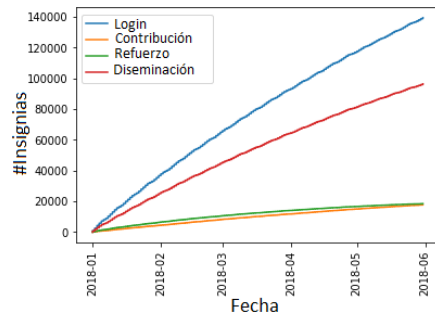


Figura 7.17: Primer semestre. Entrega de insignias a los usuarios a lo largo del tiempo.

	Rango 1	Rango 2	Rango 3	Rango 4	Rango 5
Real	61129	5084	3658	332	1446
Porcentual	85.31 %	7.09 %	5.1 %	0.46 %	2.01 %

Tabla 7.10: Rango de jugadores en comunidad de python, primer semestre del 2018.

la acumulación de acciones necesarias para obtener cada insignia. Al igual que en la figura anterior, el eje x representa los niveles de insignia y el eje y denota el total de entradas requeridas desde el comienzo para ganar alguna insignia. En esta figura, es posible ver una diferencia entre la dificultad para ganar las insignias de diseminación y refuerzo en comparación con las otras insignias; se vuelven más desafiantes, a un ritmo más rápido que las otras. Esto se debe a que estas acciones son el objetivo principal de la recompensa y también el número de acciones de refuerzo representa gran parte de las actividades de los usuarios de la comunidad.

La Figura 7.17 muestra cómo se habrían asignado insignias con este conjunto de datos y configuración. El eje x es el tiempo a lo largo de los seis primeros meses de datos, y el eje y , el número de insignias entregadas. Por ejemplo, en febrero de 2018, el juego entregó cerca de 20.000 insignias de Diseminación. Como podemos ver, existe una correlación entre las acciones de refuerzo y las de contribución porque ambas líneas crecen con valores similares.

Por último, la Tabla 7.10 muestra la distribución de jugadores en cada uno de los rangos tanto con el número entero como con la representación porcentual del mismo sobre el total de usuarios de este conjunto de datos. A partir de estos datos, podemos observar la presencia de usuarios en todos los rangos y una proporción coherente de los mismos: la distribución de los rangos es similar a una pirámide con más presencia en los rangos más fáciles y un pequeño grupo de jugadores en el rango más difícil. Junto con una distribución esperada de insignias basada en la función de recompensa elegida, donde las insignias de diseminación son las más relevantes.

Segundo semestre - Aplicando configuración del primer semestre

Además de observar cómo se desenvuelve el agente buscando una configuración para la estrategia de ludificación simulada sobre Stack Overflow, se desea validar que dicha configuración sea efectiva en periodos posteriores a los conocidos por el agente. Por lo cual, como primera evaluación en el conjunto de datos perteneciente al segundo semestre del 2018 de la comunidad de python de Stack Overflow, se tomó la configuración obtenida por el agente en el entrenamiento sobre el conjunto de datos previo (Tabla 7.9) y se aplicó a la simulación con este nuevo conjunto de datos. Como se mencionó

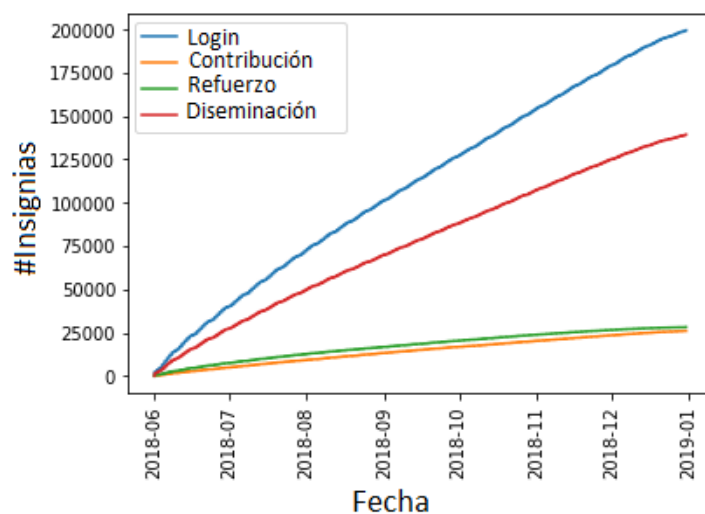


Figura 7.18: Segundo semestre - Configuración del primer semestre. Entrega de insignias a los usuarios a lo largo del tiempo.

anteriormente la métrica de evaluación fue el valor de recompensa otorgada a esta configuración sobre la simulación, este valor fue de 11.862.732 puntos. Si bien el valor es significativamente mayor, esto se debe a que la actividad en el segundo semestre también es significativamente mayor. Esto permite observar que los valores de recompensa están correlacionados al tamaño del conjunto de datos de la simulación.

Al igual que en la primera mitad, la información relacionada con la dificultad y escalado de adquisición de insignias es el mismo debido ya que se utiliza la misma configuración. En cambio, la distribución en el tiempo de las insignias aumenta significativamente, como se muestra en la Figura 7.18. Esto está relacionado con el incremento de la actividad de los usuarios con respecto al conjunto de datos anterior. Sin embargo, la distribución de las insignias es similar en ambos conjuntos de datos y significa que la configuración tiene solidez para la misma comunidad a lo largo del tiempo.

En la Tabla 7.11 se presenta la distribución de jugadores en cada uno de los rangos. Estos datos muestran una proporción esperada debido a la continuidad en el tiempo del conjunto de datos. Junto a esto, podemos observar que el aumento de los valores de recompensa está relacionado con el número de jugadores por rango, siendo casi el doble en los rangos más altos, aunque en realidad, la proporción entre los dos semestres es casi la misma. Esto significa que la configuración generada por el agente se ajusta correctamente más allá del período conocido por el agente.

	Rango 1	Rango 2	Rango 3	Rango 4	Rango 5
Real	86090	7365	5483	658	2125
Porcentual	84.63 %	7.24 %	5.39 %	0.64 %	2.08 %

Tabla 7.11: Rango de jugadores en comunidad de python, segundo semestre del 2018 con la configuración del primer semestre.

Segundo semestre - Generando nueva configuración con el agente pre-entrenado

Además de evaluar las configuraciones en periodos posteriores en las que fueron generadas, se quiere evaluar al agente en dichos periodos. Particularmente se quiere evaluar si en periodos posteriores, una configuración puede optimizarse aún más a partir de los nuevos datos históricos. Por lo cual, la segunda parte de la evaluación se realizó mediante la generación de una nueva configuración con el agente pre-entrenado en el conjunto de datos del primer semestre.

Para obtener esta nueva configuración, se le presentó al agente el entorno con la simulación de la ludificación cargada con los datos del segundo semestre de la comunidad. De esta forma, se le permitió al agente interactuar modificando las configuraciones del entorno durante una única secuencia de configuración (a lo sumo 320 acciones de modificación). Durante este proceso el agente recibió tanto la información del estado de la simulación por cada modificación realizada a la configuración, así como la recompensa relacionada. Sin embargo, este proceso se diferenció del realizado en el entrenamiento del agente sobre el primer conjunto de datos en que no se permitió aprendizaje por parte del agente sobre el nuevo conjunto de datos.

El agente obtuvo por parte de la función de recompensa una puntuación máxima de 9.858.138 puntos por esta nueva configuración, la cual que se presenta en la Tabla 7.12. Como se mencionó anteriormente, debido a las diferencias en la participación durante el primer semestre y el segundo, los rangos de los puntajes resultan diferentes, presentando rangos menores en los puntajes del primer semestre en comparación a los rangos de los puntajes del segundo semestre. Por lo cual, el puntaje de esta configuración es mayor al puntaje de la configuración anterior en el primer semestre. Sin embargo, la puntuación obtenida con la configuración actual es menor respecto de la configuración anterior evaluada en el mismo conjunto de datos. Esto indica que la nueva configuración es en primera instancia inferior a la configuración obtenida en el primer semestre.

Continuando con el formato de presentación de los casos anteriores las Figuras 7.19, 7.20, y 7.21 la información relacionada con la dificultad, el escalado y la distribución de las insignias para este conjunto de datos con la

Tipo de acción	<i>#max_badges</i>	<i>m_{oc}</i>	<i>b_{oc}</i>	<i>m</i>	<i>b</i>
Contribución	10	5	6	7	8
Refuerzo	10	9	11	12	14
Diseminación	10	14	15	16	17
Login	10	13	2	3	4

Tabla 7.12: Nueva configuración para comunidad de python, por el agente pre-entrenado, segundo semestre del 2018.

	Rango 1	Rango 2	Rango 3	Rango 4	Rango 5
Real	82128	18750	77	86	680
Porcentual	80.73 %	18.43 %	0.07 %	0.08 %	0.66 %

Tabla 7.13: Rango de jugadores en comunidad de python, segundo semestre del 2018 con la configuración nueva del agente pre-entrenado.

configuración obtenida del entrenamiento. De estas figuras se puede observar, que si bien se sigue buscando el equilibrio entre periodos de alta y baja actividad, los saltos entre estos en algunos casos como las insignias de Diseminación, son muy altos. Además, por la poca diferencia entre las cantidades acumuladas de insignias requeridas se puede determinar que la dificultad entre las mismas es demasiado similar y apenas siguiendo los lineamientos de la función de recompensa.

En conjunto con estas figuras se presenta la Tabla 7.13 que muestra la distribución de jugadores en cada uno de los rangos. En esta tabla a su vez se observa una disminución de la cantidad de usuarios en los rangos más alto indicando que la configuración dificultaría el avance de los jugadores.

Por lo tanto, se puede observar que esta configuración es inferior a la generada inicialmente. Esto se debe probablemente a los cambios en la información que se le brinda al agente como datos de entrada, ya que los datos del segundo semestre tienen valores mucho más altos de los conocidos por el agente. Por lo tanto, para poder cumplir el objetivo de mejorar la configuración del segundo semestre, se deberá enseñar al agente a manejar estos nuevos valores como se verá en la siguiente evaluación.

Segundo semestre - Obteniendo una nueva configuración por re-entrenamiento del agente

En base al resultado anterior, para lograr obtener una configuración optimizada para este periodo de tiempo se intentó adaptar al agente pre-entrenado, a fin de no recurrir a un proceso de entrenamiento desde 0, a los nuevos valores de actividades y usuarios presentes en los datos del segundo semestre de la comunidad. Para ello se realizaron 3 nuevas etapas de

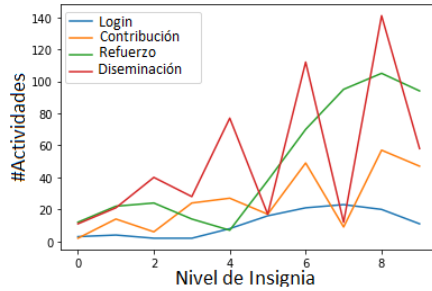


Figura 7.19: Segundo semestre - Configuración Nueva. Nuevas acciones necesarias para obtener cada insignia.

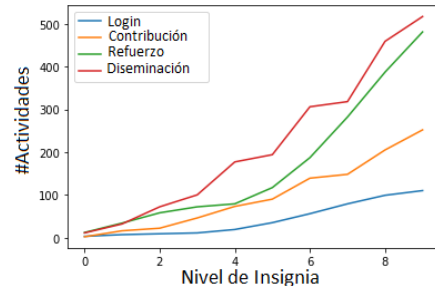


Figura 7.20: Segundo semestre - Configuración Nueva. Total de acciones necesarias para obtener cada insignia.

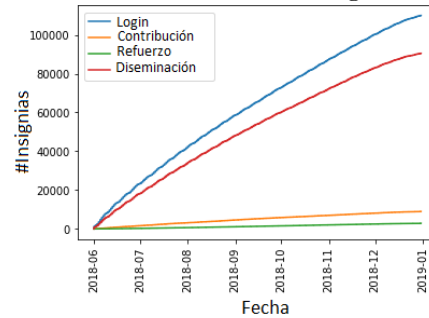


Figura 7.21: Segundo semestre - Configuración Nueva. Entrega de insignias a los usuarios a lo largo del tiempo.

entrenamiento de diferentes longitudes. Esta metodología de evaluación se utilizó para observar además cuánto entrenamiento podría requerir un agente pre-entrenado para adaptarse a nuevos contextos similares. Los periodos seleccionados fueron 1, 10 y 20 episodios.

En cada uno de los periodos (1, 10, 20), el agente generó configuraciones que se pueden observar en las Tablas 7.14, 7.15, y 7.16, por las cuales recibió una puntuación máxima de la función de recompensa de 11.661.873, 11.233.837, y 11.917.352 respectivamente. A partir de los valores de recompensa, se puede observar primero una clara mejora respecto al caso anterior del agente pre-entrenado, lo cual nos permite comprobar que un reentrenamiento por corto que sea sobre el nuevo conjunto de datos permite al agente adaptarse a los nuevos valores por más dispares que sean. Sin embargo, no es hasta el periodo de 20 episodios donde se logra finalmente superar la

Tipo de acción	<i>#max_badges</i>	<i>m_{oc}</i>	<i>b_{oc}</i>	<i>m</i>	<i>b</i>
Contribución	11	6	6	9	1
Refuerzo	11	3	9	7	9
Diseminación	11	13	19	18	18
Login	11	2	3	0	4

Tabla 7.14: Nueva configuración para comunidad de python, por el agente reentrenado en 1 episodio, segundo semestre del 2018.

Tipo de acción	<i>#max_badges</i>	<i>m_{oc}</i>	<i>b_{oc}</i>	<i>m</i>	<i>b</i>
Contribución	10	6	4	6	2
Refuerzo	10	9	10	11	13
Diseminación	10	14	19	17	14
Login	10	0	3	6	6

Tabla 7.15: Nueva configuración para comunidad de python, por el agente reentrenado en 10 episodios, segundo semestre del 2018.

Tipo de acción	<i>#max_badges</i>	<i>m_{oc}</i>	<i>b_{oc}</i>	<i>m</i>	<i>b</i>
Contribución	11	7	4	1	11
Refuerzo	11	13	9	10	8
Diseminación	11	19	6	18	14
Login	11	3	2	0	5

Tabla 7.16: Nueva configuración para comunidad de python, por el agente reentrenado en 20 episodios, segundo semestre del 2018.

puntuación obtenida de la configuración del primer semestre.

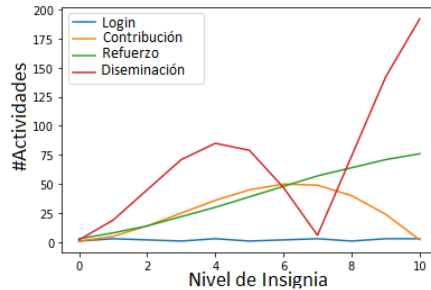


Figura 7.22: Segundo semestre - Configuración por reentrenar 1 episodio. Nuevas acciones necesarias para obtener cada insignia.

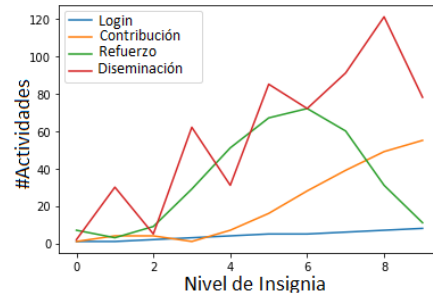


Figura 7.23: Segundo semestre - Configuración por reentrenar 10 episodios. Nuevas acciones necesarias para obtener cada insignia.

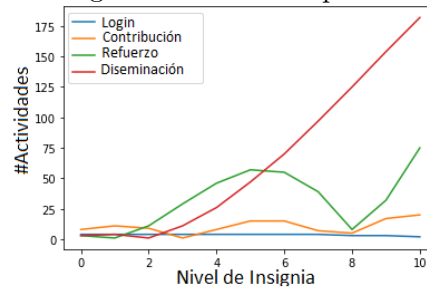


Figura 7.24: Segundo semestre - Configuración por reentrenar 20 episodios. Nuevas acciones necesarias para obtener cada insignia.

Las Figuras 7.22, 7.23 y 7.24 muestran para cada periodo el número de nuevas acciones necesarias para obtener cada insignia según el tipo de acción. De manera similar, las Figuras 7.25, 7.26 y 7.27 muestran el total de acciones que se requieren para obtener cada insignia en cada periodo. En estas figuras podemos ver como inicialmente el agente selecciona un mayor grado de dificultad para insignias de contribución que a medida que se avanza en los periodos la misma disminuye. También observamos como la dificultad de las insignias de diseminación y refuerzo van en aumento hasta lograr marcar una separación con las otras insignias. En estos cambios se puede observar al agente siguiendo la heurística de la función de recompensa en la forma de priorizar los tipos de acciones.

Las Figuras 7.28, 7.29 y 7.30 muestran por periodo cómo se entregaron las insignias a los usuarios a lo largo del tiempo. En estas figuras se observa

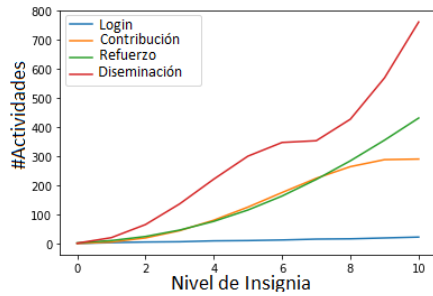


Figura 7.25: Segundo semestre - Configuración por reentrenar 1 episodio. Total de acciones necesarias para obtener cada insignia.

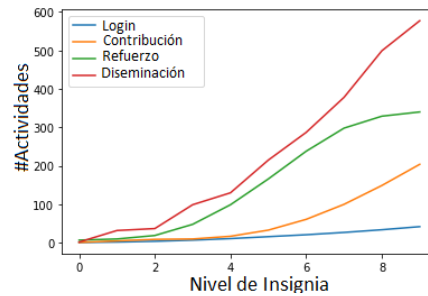


Figura 7.26: Segundo semestre - Configuración por reentrenar 10 episodios. Total de acciones necesarias para obtener cada insignia.

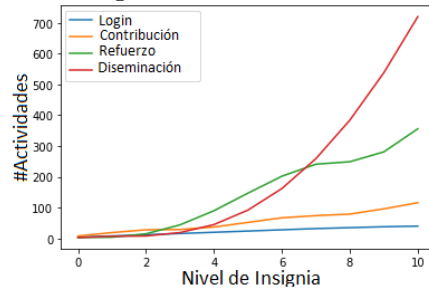


Figura 7.27: Segundo semestre - Configuración por reentrenar 20 episodios. Total de acciones necesarias para obtener cada insignia.

además que en la medida que se entrenó por más periodos las insignias entregadas disminuyeron en cantidad como consecuencia del aumento de dificultad, pero manteniendo un mejor balance entre los tipos de insignias, siendo este balance el solicitado por la función de recompensa.

Finalmente, la Tabla 7.17 muestra la distribución de jugadores a partir de la configuración obtenida de cada periodo de entrenamiento, en este caso se mantuvieron solo los valores porcentuales para facilitar la lectura de la misma. De estos datos podemos observar que se obtiene nuevamente una correcta proporción de los usuarios por los diferentes rangos y una distribución esperada de insignias en base a la función de recompensa elegida, donde las insignias de diseminación son las más relevantes. Además, si bien el periodo de 10 episodios obtuvo una puntuación menor que el de 1 episodio se puede ver que posee una distribución más proporcional entre rangos que se conserva en el periodo siguiente y con una puntuación mayor.

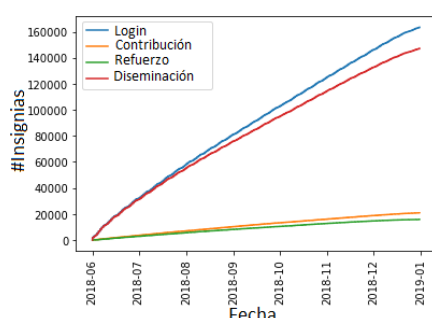


Figura 7.28: Segundo semestre - Configuración por reentrenar 1 episodio. Entrega de insignias a los usuarios a lo largo del tiempo.

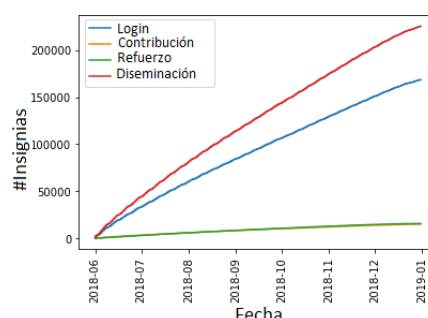


Figura 7.29: Segundo semestre - Configuración por reentrenar 10 episodios. Entrega de insignias a los usuarios a lo largo del tiempo.

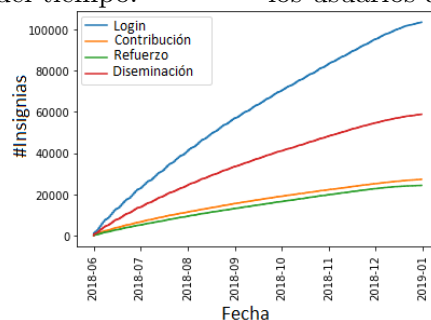


Figura 7.30: Segundo semestre - Configuración por reentrenar 20 episodios. Entrega de insignias a los usuarios a lo largo del tiempo.

7.2.4. Análisis de resultados

En esta sección se presenta un análisis de los resultados obtenidos de la evaluación del enfoque sobre Stack Overflow. Como parte de la evaluación del enfoque se quiso estudiar la efectividad de las configuraciones en periodos posteriores al conocido por el agente. Y posteriormente intentar mejorarla. Para ello se trabajó con el conjunto de datos dividido en dos periodos consecutivos de 6 meses cada uno.

La primera diferencia relevante que se encontró al trabajar con los datos divididos en el primer y segundo semestre se basó en que el nivel actividad y la cantidad de participantes en ambos conjuntos eran distintas. Siendo el conjunto del segundo semestre el que presento un mayor nivel en ambos casos respecto del primero. Esto afecto el rendimiento del agente entrenado

	Rango 1	Rango 2	Rango 3	Rango 4	Rango 5
1 episodio	74.74 %	19.81 %	3.7 %	0.13 %	1.58 %
10 episodios	74.68 %	22.64 %	0.97 %	0.41 %	1.28 %
20 episodios	80.76 %	14.83 %	0.45 %	0.31 %	3.64 %

Tabla 7.17: Rango de jugadores en comunidad de python, segundo semestre del 2018. Por periodo de reentrenamiento.

en el primer semestre al verse enfrentado contra valores fuera de las métricas que este conocía.

Sin embargo, se pudo observar que la configuración obtenida en este primer semestre resulto ser efectiva, aunque no óptima, en la simulación del segundo semestre. Esto se debe a que, si bien las magnitudes son distintas y esto impacta en el agente, las proporciones de actividad y la forma de actividad de los usuarios no variaron de un semestre a otro por lo cual la configuración se mantenía dentro de márgenes aceptables.

Cuando se buscó obtener una configuración que ajuste mejor con la comunidad del segundo semestre se encontró que el agente entrenado previamente tenía un desempeño inferior. Esto se debió a que, como se mencionó previamente, los valores del segundo semestre estaban por fuera de los rangos que el agente era capaz de generalizar. Las fallas del desempeño más allá de la puntuación se pueden observar en la forma de distribución de las insignias, donde las fluctuaciones de dificultad son muy grandes en insignias como la de diseminación. También podemos observar una sobre valorización de las insignias de contribución en esta configuración, mientras se presenta una menor entrega de insignias de refuerzo lo cual limita el avance de los jugadores en los rangos.

Por lo cual para poder mejorar la configuración del segundo semestre se procedió a realizar un reentrenamiento del agente en el nuevo conjunto de datos. Esto permitió observar una mejora de la puntuación dada por la función de recompensa junto con cambios en la forma en que se maneja la dificultad de las insignias. Estos cambios se acentuaron a mayor número de episodios de entrenamiento se generan.

Finalmente, dentro de los casos exitosos de configuración por parte del agente podemos observar una separación más pronunciada en dos grupos de insignias, diseminación y refuerzo contra contribución y login. Si bien entre todos los entrenamientos se conserva dentro de lo posible un ordenamiento de prioridad de las insignias dictaminado por la función de recompensa, en estos casos particulares se acentúa aún más el impacto de la misma. Además, en estos casos podemos observar como el agente maneja la curva de dificultad de insignias como la de diseminación para permitir una adquisición

inicial sencilla brindando acceso así a los primeros rangos. Mientras que con insignias como la de refuerzo la dificultad de la curva se achata en los niveles superiores para ayudar a la conservación del ultimo rango con su característica temporal.

7.2.5. Conclusiones

Este caso de estudio presenta la aplicación del enfoque de RL para optimizar un diseño ludificado para una CCC diferente de la de Wikipedia, en este caso Stack Overflow, con la menor cantidad de cambios posibles. También se introdujo con este fin nociones de la estructura de datos de la comunidad, así como un proceso de extracción y mapeo de datos el cual solo requirió impactar en una única clase del framework. De esta forma se demuestra la flexibilidad del mismo para la adaptación a nuevas comunidades a ludificar de forma óptima.

También se demostró un proceso de adaptación del agente a nuevos conjuntos de datos y evaluaciones de las configuraciones obtenidas por el mismo. Comprobando que, si bien las configuraciones de un periodo previo pueden mantenerse de forma exitosa, en caso de querer optimizar esta configuración se requiere obtener una nueva del agente. En esta situación también se demostró la sensibilidad del agente a los datos del conjunto de datos lo cual será tratado en el próximo caso de estudio cuando se evalúe la integración de agentes distintos interactuando con el framework.

7.3. Integración de nuevos agentes

Este caso de estudio se evalúa la flexibilidad del enfoque y su implementación para poder adaptarse e integrar nuevas estrategias o avances en el área de RL. Para ello se planteará la integración de nuevos agentes de RL al enfoque previamente planteado, observando los cambios requeridos dentro del framework Historical RL.

Estos nuevos agentes fueron tomados de dos frameworks de RL para así comprobar correctamente la flexibilidad del enfoque y del framework en lo respectivo a la interacción con distintos agentes. Estos frameworks así como los agentes seleccionados se presentarán a continuación.

También en base a los resultados del caso de prueba de la Sección 7.2, este caso de estudio busca comprobar el impacto de distintas entradas en lo que respecta a estado y recompensa para los agentes. Con este fin, junto con la simulación y el conjunto de datos seleccionados se presentarán diferentes formatos de los datos de entrada. Luego se explicarán la metodología seleccionadas para la evaluación. Finalmente, se presentarán los resultados y una discusión sobre los mismos.

7.3.1. Selección de agentes de RL

En este caso de estudio se quiere evaluar la capacidad del enfoque utilizando el framework Historical RL para interactuar con diferentes agentes. Dada la presente implementación del framework respetando la interfaz *Env* de GYM, esta integración debería ser sencilla al punto de requerir solo la creación de una instancia de un nuevo agente e indicarle que interactúe con el entorno de nuestra simulación provista por el framework. Para validar esta hipótesis se utilizará un conjunto de agentes tomados de dos librerías de RL, Keras RL2 y Stable Baselines 3. Estas librerías serán introducidas a continuación, así como los agentes seleccionados de las mismas.

Keras RL2

Keras RL2⁴ es la actualización a Tensorflow 2 (presentado en el Capítulo 6) de la librería Keras RL[136]. Esta librería que trabaja sobre la integración de Keras y TensorFlow para brindar implementaciones de algunos algoritmos del estado del arte de Reinforcement Learning (RL). Además, esta librería es compatible con el estándar de GYM respetado por el framework presentado en este trabajo.

Si bien esta librería presenta gran variedad de algoritmos de RL, para esta comparación nos centramos en aquellas que manejan acciones de valores discretos ya que son con las que el simulador actualmente ya venía trabajando. Siendo una acción que maneja valores discretos aquella que recibe el identificador de la acción que se desea realizar.

El primer algoritmo seleccionado es Double DQN[96] la cual es una variante del algoritmo clásico de aprendizaje por refuerzo profundo que introduce redes neuronales al algoritmo q-learning, este es un algoritmo “fuera de la política” o *off policy*. Por otro lado, también se utilizó el algoritmo Deep SARSA[138] el cual es a su vez una implementación del algoritmo SARSA con la aplicación de redes neuronales, este algoritmo a diferencia de DQN es un algoritmo “dentro de la política” o *on policy*. Como ya se mencionó anteriormente se optó por enfoques de aprendizaje por refuerzo profundo dada la naturaleza de la problemática de configuración tratada en este trabajo, ya que nos encontramos con un gran número de estados, variables y acciones que no es seguro que puedan ser abarcadas de forma eficiente por las implementaciones clásicas de estos algoritmos.

Stable Baselines 3

Stable Baseline 3 [137] es la una actualización mayor de Stable Baseline [139] la cual es una librería con implementaciones mejoradas de algoritmos

⁴Repositorio:<https://github.com/wau/keras-rl2>. Visitado el 12/11/20.

de aprendizaje por refuerzo. Esta librería trabaja sobre PyTorch el cual fue previamente introducido en el Capítulo 6.

Esta librería tiene como objetivo facilitar a la comunidad científica y a la industria la replicación, mejora y generación de nuevos proyectos utilizando aprendizaje por refuerzo profundo. Con este objetivo sus algoritmos poseen interfaces comunes, basadas en Baselines[140] de OpenAI y scikit-learn[141, 142].

Al igual que la librería anterior, los algoritmos se seleccionaron entre los que permitían la interacción con acciones de un espacio de acción discreto. Por lo cual los algoritmos seleccionados fueron tres que si bien no se detallaran extensamente se mencionan a continuación.

El algoritmo DQN[143] en su variante original como algoritmo *off-policy*. Por otro lado, como algoritmos *on-policy* tenemos A2C [144], que es una versión sincrónica de A3C que implementan el modelo de actor-critic para el paradigma de RL, y PPO[145] que es un algoritmo que combina ideas de A2C (delegando en múltiples procesos) y TRPO(utilizando regiones de confianza para mejorar al actor).

7.3.2. Metodología de evaluación

Este caso de estudio desea evaluar en conjunto la posibilidad de adaptación del enfoque a nuevas tecnologías de RL, en este caso mediante la integración diferentes agentes, y el desempeño de los agentes en base al formato de los elementos que se utilizan como entrada (estados y recompensas) para que puedan elegir la mejor configuración. Por lo tanto, para esta evaluación se utilizará la CCC de Wikipedia, particularmente su comunidad relacionada con la Wikicup con la cual ya se trabajó en el caso de estudio de la sección 7.1.

Esta elección se basó principalmente por el bajo costo de tiempo ejecución requerido para la evaluación de la simulación comparado con los otros conjuntos de datos utilizados en este trabajo. Por otro lado, ya que se busca estudiar la capacidad de generalización de los agentes en base al tipo de entrada recibida se requirió un conjunto de datos de evaluación, y esta comunidad por su parte facilita la elección del conjunto de datos de validación dado que las competencias de la Wikicup son autocontenidas por año.

Por lo tanto, para estas evaluaciones se usaron los conjuntos de datos pertenecientes a las ediciones 2018 y 2019 de la Wikicup de la Wikipedia en su versión inglesa. El conjunto de datos obtenido para la edición 2018 tiene 549 artículos y 1.522.011 acciones de la estrategia de ludificación. Mientras que la edición 2019 consto de 350 artículos y un total de 1.395.231 acciones de la estrategia de ludificación.

También se obtuvo la lista de participantes con su puesto y puntos de ambas ediciones de la WikiCup, ya que la misma se requiere en la función de recompensa introducida en el caso de estudio de la sección 7.1 (Función 7.3). Esta función de recompensa a su vez es la que permitirá evaluar el desempeño de las configuraciones generadas por el agente, siendo al igual que en el caso anterior que a mayor recompensa mejor desempeño.

El impacto del formato de los datos de entrada para el agente es evaluado con la función de recompensa. Las alternativas de formato para el estado y las recompensas que brindaba el entorno del framework fueron de dos tipos. Por parte del estado, si bien los jugadores por rango ya eran informados de forma porcentual al agente, no era la misma situación para los valores de configuración actuales que se le informaban al agente. Para estos valores como alternativa se agregó al framework la capacidad de normalizar los valores de las configuraciones con el fin de evaluar si estos cambios generan un mejor desempeño en las configuraciones obtenidas.

Esto se logró aplicando una normalización en forma de max-min (Función 7.5) en la cual el valor x es disminuido por el mínimo valor que puede alcanzar esa variable y posteriormente dividido por la resta entre el máximo y el mínimo valor alcanzable por la misma variable. Esto nos permite re-escalar nuestra variable a valores entre 0 y 1 sea cual sea la magnitud de valor original.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (7.5)$$

donde:

$\max(x)$ = retorna el valor máximo en el rango de valores de x .

$\min(x)$ = retorna el valor mínimo en el rango de valores de x .

Como alternativa al formato de la recompensa presentada en la Función 7.3 se ofrece a la planteada en la Función 7.6. Esta función es una versión normalizada mediante el mismo proceso de máximos y mínimos mencionada anteriormente. Particularmente los valores de esta función se adaptarán en base al conjunto de datos que se utilice, permitiendo que los valores se encuentren entre 0 y 1 más allá del conjunto de datos que se esté utilizando. Esto además de permitir evaluar los cambios en el desempeño se espera que sirva de métrica genérica para los puntajes recibidos de datasets con datos de diferentes magnitudes

$$\text{reward}' = \frac{\sum_{i=1}^5 i \times \text{sumPoints}(\text{usersInRank}(i)) - \min(\text{reward})}{\max(\text{reward}) - \min(\text{reward})} \quad (7.6)$$

donde:

$usersInRank(i)$ = retorna el conjunto de usuarios que alcanzaron el rango i en la simulación (Visitante es el rango 1 y Ciudadano Científico Comprometido es el rango 5).

$sumPoints(U)$ = retorna la suma de los puntos obtenidos en la Wikicup por los usuarios en el conjunto U .

$max(reward)$ = retorna el valor calculado como máxima recompensa para un determinado conjunto de datos.

$min(reward)$ = retorna el valor calculado como mínima recompensa para un determinado conjunto de datos.

Con estos nuevos métodos de entrada tanto para variables del estado de la simulación como para la recompensa, se propone una evaluación de los agentes seleccionados para cada combinación de entrada (estado/recompensa), ejecutando sobre las simulaciones de Wikicup 2018 y 2019.

7.3.3. Resultados

Esta sección presenta los resultados de comparar a las configuraciones obtenidas por los 5 agentes sobre distintas configuraciones de entornos. Estas configuraciones consistieron en definir tipo de estado a utilizar (clásico o normalizado), el tipo de recompensa utilizar para entrenar al agente (clásica o normalizada), y finalmente sobre qué conjunto de datos realizar la simulación (Wikicup 2018 o 2019). Si bien solo se puede utilizar una recompensa como métrica para que el agente entrene, los valores de la recompensa tantos normalizados como sin normalizar fueron registrados para facilitar la comparación, legibilidad y posteriores representaciones gráficas de los resultados.

Las simulaciones sobre el conjunto de datos de la Wikicup 2018 con sus 4 variantes por la combinación de tipo de estado y tipo de recompensas se utilizaron para entrenar cada agente durante 100 episodios. Esto generó un total de 20 agentes entrenados. Estos agentes fueron evaluados tanto sobre la misma configuración en la que se los entrenó (Wikicup 2018 con el formato de estado y recompensa respectivo), como sobre la simulación sobre el conjunto de datos de la Wikicup 2019, también con sus respectivas configuraciones de estado y recompensa. Este método de evaluación dio un total de 60 configuraciones diferentes del simulador entre entrenamiento y evaluación.

Si bien la integración de nuevos agentes resultó directa dado que el framework respeta el protocolo de GYM, se desea ver como es el desempeño de estos agentes con el enfoque propuesto en este trabajo. Las Figuras 7.31 y 7.32 muestran los resultados de las mejores configuraciones obtenidas al

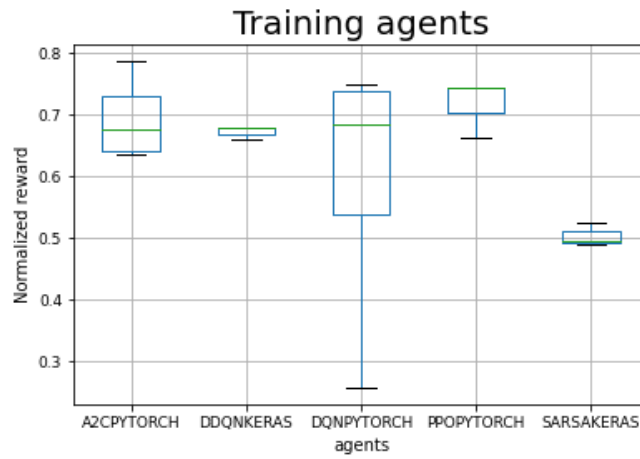


Figura 7.31: Resultados del entrenamiento agrupado por agentes.

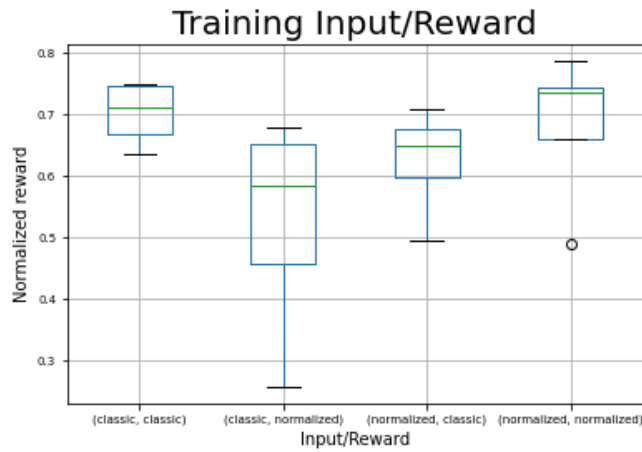


Figura 7.32: Resultados del entrenamiento agrupado por tipo de estado y tipo de recompensa.

evaluar la función de recompensa durante la etapa de entrenamiento en diagramas de caja. Estas evaluaciones se realizaron por cada agente para cada combinación de formato de estado y formato de recompensa.

La Figura 7.31 muestra los resultados agrupados por agente los cuales se distribuyen en el eje x , mientras que en el eje y podemos observar los valores de recompensa en su formato normalizado para facilitar la comparación con resultados en otros conjuntos de datos. Este diagrama nos permite observar por ejemplo que la distribución de recompensas para el agente “SarsaKeras” fue la que obtuvo menor puntaje por la función de recompensa durante el entrenamiento indicando un desempeño bajo por parte del agente. También nos permite observar que para el agente “DQNPYTORCH” se obtuvo la mayor variación en los valores de recompensa, esto está dado por la cobertura de su caja, y con un valor medio cercano al 0.69 indicado por la línea interna de la caja. Esto indica que es uno de los agentes que más volatilidad presenta en su desempeño.

La Figura 7.32 en cambio muestra los resultados agrupados por formato de estado y función de recompensa en formato “estado/recompensa”, estas agrupaciones se disponen sobre el eje x indicando *classic* para valores sin normalizar y *normalized* para valores normalizados. Este gráfico nos permite ver cómo afecta los formatos de los datos en el desempeño de todos los agentes durante el entrenamiento, por ejemplo, podemos ver que el mejor desempeño está dado por el uso de combinaciones del mismo tipo. Esto se observa en que las etiquetas “classic/classic” y “normalized/normalized” poseen las cajas con los valores más altos, siendo particularmente la de estado y recompensa normalizadas la que presenta mejor desempeño en esta etapa.

Las Figuras 7.33 y 7.34 presenta de forma similar los resultados para las evaluaciones en la simulación de la Wikicup 2018 de los agentes ya entrenados. Estos diagramas de cajas se encuentran también agrupados por agente y por la combinación de formato estado/recompensa respectivamente. Si bien los valores de recompensa obtenidos son más bajos que en el entrenamiento esto era esperable dado que solo se dio un episodio (320 acciones de configuración) como oportunidad de evaluación a cada agente. También se puede observar mayor amplitud en el rango de las recompensas probablemente por el mismo motivo.

Finalmente, las Figuras 7.35 y 7.36 continuando con el formato de presentación anterior muestran los resultados para las evaluaciones en la simulación ahora en Wikicup 2019 de los agentes ya entrenados. Nuevamente estos valores agrupados por agente y por la combinación de formatos de estado/recompensa muestran una disminución en el valor de recompensa en este caso dado por la presencia de un entorno desconocido dado por el nuevo conjunto de datos. Sin embargo, este conjunto de datos ya da información sobre la capacidad de generalización de cada agente y según el tipo de estado

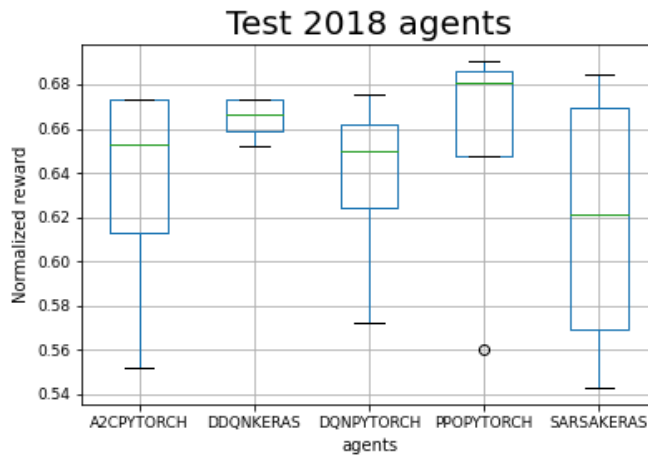


Figura 7.33: Resultados de la evaluación en Wikicup2018 agrupado por agentes.

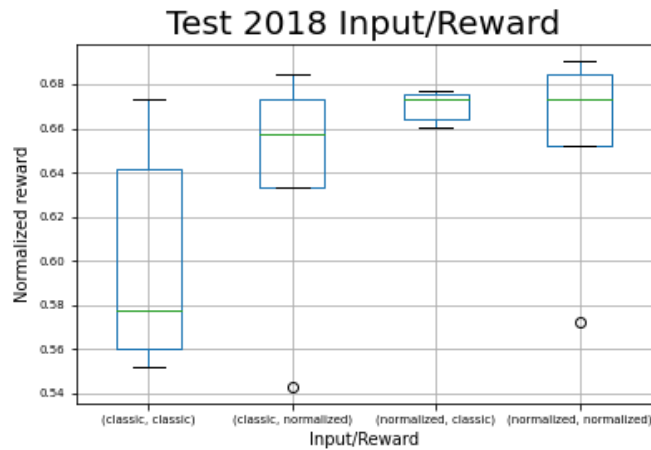


Figura 7.34: Resultados de la evaluación en Wikicup2018 agrupado por tipo de estado y tipo de recompensa.

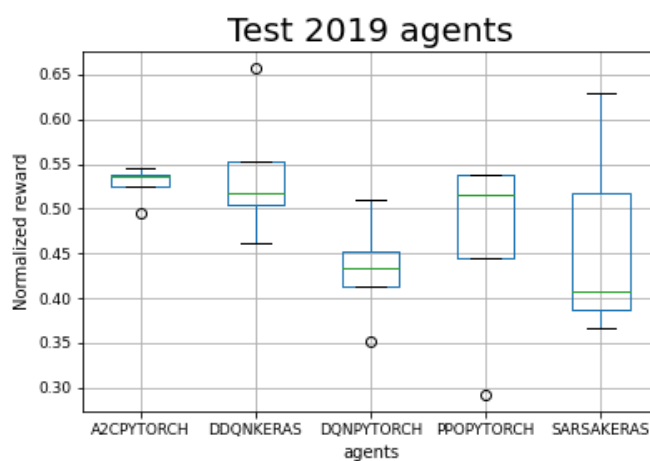


Figura 7.35: Resultados de la evaluación en Wikicup2019 agrupado por agentes.

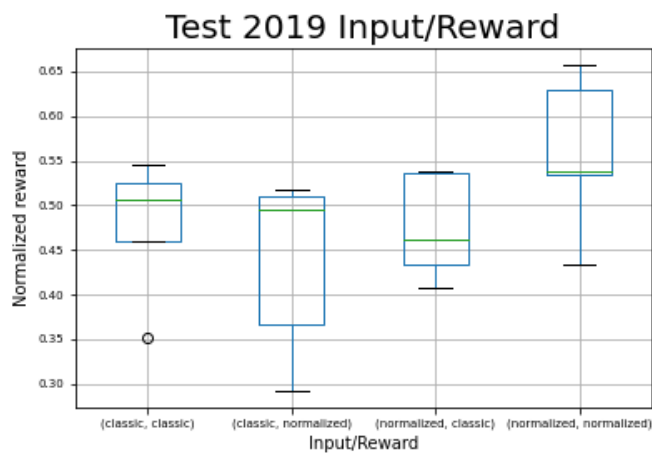


Figura 7.36: Resultados de la evaluación en Wikicup2019 agrupado por tipo de estado y tipo de recompensa.

o recompensa.

7.3.4. Análisis de resultados

En esta sección se presenta un análisis de los resultados obtenidos de la evaluación del enfoque en lo referente a la integración de nuevas tecnologías de RL. Particularmente mediante la integración del framework Historical RL con diferentes agentes de RL y el desempeño de los mismos en base al formato con el que se le comuniquen los datos del simulador al agente. Sobre la integración de los agentes con el framework se puede decir que solo requiere la creación de una instancia de la clase *Historical Enviroment* con la cual el agente podrá interactuar automáticamente gracias a que la misma cumple con el protocolo de GYM.

Por otro lado, los resultados respecto a el desempeño de los agentes basado en el tipo de entrada que brinda el simulador varían en las etapas de entrenamiento y las dos etapas de evaluación. En la etapa de entrenamiento podemos observar que a los agentes les resulta más sencillo trabajar con variables que sigan lineamientos similares ya sean normalizadas o sin normalizar. Sin embargo, en este punto ya podemos ver una mejora sutil en el desempeño con ambas variables normalizadas en comparación con las demás combinaciones.

Luego durante la evaluación se observa una disminución general de las recompensas máximas otorgadas por la función de recompensa ya que los agentes poseen menos oportunidades de buscar configuraciones que en el entrenamiento. Sin embargo, en el caso del conjunto de datos de la Wikicup 2018 podemos observar una mejor generalización por parte de los agentes entrenados con al menos una variable normalizada. Y finalmente, en el conjunto de datos de la Wikicup 2019 que es completamente desconocido para los agentes observamos que el mejor desempeño esta dado al tener ambas variables normalizadas, confirmando por lo tanto que el formato en que se entregan los datos al agente influye en su desempeño y que la normalización en este caso mejora la generalización por parte de los agentes para nuevos conjuntos de datos. Esto a su vez indica que mientras el enfoque pueda adaptarse en formatos a las tecnologías podrá adaptarse a nuevas estrategias de los agentes.

Respecto al desempeño específico de los agentes según su algoritmo podemos decir que los tres mejores fueron PPO de Stable Baselines 3, D-DQN de Keras RL3, y A2C de Stable Baselines 3 nuevamente. Sobre PPO, los gráficos muestran que es el que mejor desempeño tuvo en el entrenamiento y mejor generalización las evaluaciones. Este algoritmo alcanzo sus mayores puntuaciones a partir del episodio 26 y del 46 del entrenamiento en las variantes doblemente normalizadas, y doblemente no normalizadas respec-

tivamente.

Por parte de D-DQN, el algoritmo no presentó el mejor desempeño durante el entrenamiento sin embargo al momento de generalizar tuvo un desempeño superior a otros agentes. Alcanzó sus puntuaciones máximas en los entrenamientos recién a partir del episodio 69 con las variables sin normalizar y del 71 con las variables normalizadas.

Finalmente, A2C, presentó un buen rendimiento durante el entrenamiento y una generalización casi tan buena como la de D-DQN pero con mayor amplitud de valores siendo por lo tanto más inestable. Durante el entrenamiento sus puntuaciones máximas ocurrieron luego del episodio 46 con ambas variables sin normalizar y el episodio 60 con las variables normalizadas. Esto nos permite observar que a excepción a lo ocurrido con el algoritmo PPO en general el entrenamiento con variables normalizadas requiere un proceso más largo, sin embargo, se obtienen agentes con una mejor capacidad de generalización.

7.3.5. Conclusiones

Este caso de estudio presenta un ejemplo de adaptación de nuevas tecnologías al enfoque propuesto en este trabajo mediante la integración de diversos agentes con el framework Historical RL, a partir de lo cual se puede observar que gracias al cumplimiento del estándar propuesto por OpenAI Gym por parte del framework, el mismo genera entornos compatibles para agentes de diversas librerías.

También se demostró la influencia del formato en que se presentan los datos de entrada (estado o recompensa) al agente en lo referente a su desempeño y capacidad de generalización. Para este fin también se presentaron nuevas funcionalidades del framework para permitir normalizar los valores de las configuraciones presentes en el estado y además una versión normalizada de la función de recompensa para la simulación de la Wikicup. Con esto se demostró que, si bien las versiones normalizadas generaban que los agentes requirieran mayor tiempo para poder realizar su entrenamiento, esto concluía generando agentes con mayor capacidad de generalización en conjuntos de datos nuevos como los de la Wikicup 2019.

Finalmente, se presentó un análisis de los agentes seleccionados y su desempeño con respecto a la tarea de configurar una estrategia de ludificación para una CCC. Por este análisis se concluyó que el agente que mejor se desempeñó durante los entrenamientos y obtuvo una mejor capacidad de generalización en nuevos conjuntos de datos fue el algoritmo PPO de Stable Baselines 3.

7.4. Resumen del capítulo

En este capítulo se presentaron tres casos de estudio de la aplicación del enfoque presentado en este trabajo mediante la utilización del framework llamado Historical RL para la configuración de sistemas basados en simulaciones históricas. Estos casos de estudio evaluaron diferentes aspectos del enfoque.

El primer caso de estudio presenta una forma de optimizar un diseño ludificado para recompensar los comportamientos deseados de los usuarios sin perturbar la funcionalidad de una CCC existente, en este caso Wikipedia. Demostrando la flexibilidad del enfoque a la hora de poder trabajar con distintas comunidades u objetivos mediante la posibilidad de definir recompensas.

El segundo caso de estudio presenta la aplicación del enfoque para una CCC nueva, Stack Overflow, solo requiriendo extender una parte del comportamiento del framework referente al proceso de extracción de datos para la nueva comunidad. También se introdujo con este fin nociones de la estructura de datos de la comunidad, así como un proceso de extracción y mapeo de datos. De esta forma demuestra la flexibilidad del enfoque para la adaptación a nuevas comunidades a las cuales poder configurarles una ludificación de forma óptima. Además, evalúa el desempeño del enfoque para obtener configuraciones para periodos de tiempo desconocidos al momento del entrenamiento.

Finalmente, el tercer caso de estudio evalúa la capacidad de adopción de nuevas tecnologías de RL. Para ello, presenta un ejemplo de la integración de diversos agentes de RL con el framework Historical RL, a partir de lo cual se puede observar que gracias al cumplimiento del estándar propuesto por OpenAI Gym por parte del framework, el mismo genera entornos compatibles para agentes de diversas librerías. También dentro de este caso se analiza la influencia en el desempeño de los agentes en base al formato en que el entorno les presenta la información, y también presenta nuevas funcionalidades del framework para alterar los formatos de la información.

A continuación de este capítulo se presentarán las conclusiones generales de este trabajo de tesis en el Capítulo 8 junto con posibles opciones o consideraciones a tener en cuenta para guiar posibles trabajos futuros en esta área de investigación.

Capítulo 8

Conclusiones y trabajo futuro

En este trabajo de tesis se presentó un enfoque para brindar apoyo a la toma de decisiones en el contexto de inserción de una estrategia de ludificación preexistente en una Comunidades de Construcción de Conocimiento, tomando en consideración el contexto de incertidumbre inherente a la forma de trabajo colaborativo en las CCC. Para ello se demostró las dificultades subyacentes de configurar una estrategia de ludificación e integrarla exitosamente en una comunidad existente.

Con estas dificultades en mente, el apoyo a la toma de decisión en este contexto fue realizado utilizando técnicas de aprendizaje por refuerzo. Estas técnicas a su vez se integraron con el framework Historical RL, el cual se desarrolló durante el transcurso de este trabajo con el fin de facilitar el uso de simulaciones basadas en datos históricos para la tarea de configurar ludificaciones en CCC.

8.1. Resumen de contribuciones

En este trabajo se realizaron las siguientes contribuciones:

8.1.1. Un estudio de factibilidad del uso de datos históricos

Para poder realizar una simulación de la integración entre una estrategia ludificación y una CCC existentes primero se requirió validar la existencia de patrones de comportamiento que reflejen el comportamiento de los miembros de la comunidad y con esto la posibilidad de usar datos históricos como entrada para la simulación de la comunidad objetivo. El Capítulo 3 analiza

esta posibilidad utilizando como caso de ejemplo artículos provenientes de la comunidad Wikipedia.

Se trabajo sobre el historial de cambios de dichos artículos analizando las diferencias entre versiones y el uso de wiki-markups por parte de los usuarios. Con esta información se presenta un análisis de patrones de uso de wiki-markups. Los patrones encontrados permitieron considerar que los datos históricos, y particularmente el wiki-markup en caso de Wikipedia, pueden usarse como indicadores de las acciones de los usuarios durante su participación en la comunidad para comprender su comportamiento y posteriormente simularlo.

8.1.2. Una estrategia para la extracción y conversión de datos históricos

En este trabajo se presenta el proceso de extracción y procesamiento de datos históricos de una CCC para su integración en la simulación. Este proceso de extracción varía según la comunidad objetivo, por lo cual, se presentaron ejemplos de este proceso por cada comunidad utilizada.

Para el caso de la comunidad de Wikipedia en el Capítulo 4 se presentan dos herramientas. Wikicrawler para la extracción de historiales de revisiones de artículos de Wikipedia, y artículos relacionados, y Wikigamificator para la extracción de eventos de dichos historiales. Ambas herramientas se usan durante la integración de las sub-comunidades de Wikipedia utilizadas en este trabajo.

La otra comunidad analizada en este trabajo es Stack Overflow. Para esta comunidad en el Capítulo 7 se presentó una estrategia de extracción de datos históricos utilizando BigQuery. Junto con esto se presenta la definición de como estos datos históricos fueron mapeados a acciones de ludificación. Además, ambos enfoques se integran al flujo de datos del framework Historical RL.

8.1.3. Una estrategia de simulación para estrategias de ludificación mediante datos históricos

Como parte de los objetivos de este trabajo se deseaba validar la posibilidad de simular una estrategia ludificación preexistente y su configuración. En el Capítulo 4, se presenta una estrategia para simular la aplicación de una estrategia de ludificación en una comunidad preexistente. Como ludificación se utiliza una implementación en python de Metagame para la cual también se presenta un análisis desde la perspectiva del framework teórico de ludificación MDA.

Mediante este análisis se presentan los elementos necesarios para simular

la ludificación. También se introduce que el uso de una simulación puede evitar resultados disruptivos como los que podrían haberse presentado con los ajustes manuales presentados en este capítulo. Finalmente, esta simulación deja en evidencia de que a pesar de ser factible simular una ludificación, el éxito de la misma depende también en gran medida de la configuración elegida para la misma.

8.1.4. Un enfoque para la obtención de configuraciones óptimas mediante aprendizaje por refuerzo

Se introduce un enfoque para brindar apoyo frente a la problemática de definir configuraciones de una estrategia de ludificación para una comunidad. Este enfoque que trabaja mediante el uso de RL y datos históricos es presentada en el Capítulo 5.

De este enfoque se deriva, en el Capítulo 6, el framework de RL llamado Historical Reinforcement Learning. Este framework facilita la obtención de sugerencias para la configuración de ludificaciones, basándose en simulaciones históricas. Mediante estas simulaciones y el desarrollo de una recompensa como objetivo a obtener con la comunidad, se generan configuraciones que guían hacia el comportamiento deseado a la comunidad manteniendo el comportamiento intrínseco de la comunidad.

Este framework permite la interacción con diferentes agentes de RL gracias a la implementación del estándar establecido por GYM, el cual junto con el resto de las tecnologías utilizadas para la implementación del framework son presentados al comienzo del capítulo. Este enfoque además es generalizable a diferentes CCC sobre las cuales realizar simulaciones.

8.1.5. Casos de uso del enfoque

Para evaluar y validar la efectividad del enfoque, se presentaron en el Capítulo 7, un análisis de tres casos de estudio de la utilización del enfoque mediante la aplicación del framework llamado Historical RL para la configuración de ludificaciones basadas en simulaciones históricas. Estos casos de estudio evaluaron diferentes aspectos del enfoque.

Se demostró la capacidad de generalización del enfoque con el framework a la hora de poder trabajar con distintas comunidades u objetivos mediante la posibilidad de definir recompensas. Así como también para la adaptación a nuevas comunidades a las cuales poder configurarles una estrategia de ludificación de forma óptima.

También se demostró que a partir de los datos históricos del comportamiento de la comunidad también se podía captar y mantener comportamientos relacionados a una estrategia de ludificación distinta a la cual se pretende

configurar. Particularmente esto se observa al adaptar la estrategia de ludificación Metagame al torneo de Wikicup donde se realizó una transferencia de conceptos de ludificación de una estrategia a otra y se presentaron las limitaciones del enfoque en este aspecto.

Otros casos de estudio presentaron un análisis de las configuraciones en distintos periodos de tiempo permitiendo validar la robustez de la misma y la posibilidad de ser posteriormente optimizadas.

Finalmente, se deseaba también que el enfoque sea flexible a la integración de nuevas estrategias y aplicación de futuros avances en el área de RL. Esto se logró gracias a la implementación del estándar propuesto por OpenAI Gym por parte del framework, este estándar permitió la utilización de diversos agentes de RL.

Junto con el análisis de estos agentes también se presenta un análisis de como la configuración del framework y la simulación puede afectar en el desempeño de los agentes en su tarea de encontrar configuraciones, siendo estos cambios en lo referente al formato de los datos comunicados a los agentes.

8.2. Limitaciones

Si bien el enfoque propuesto se encuentra soportado por un framework que facilita la integración entre los distintos elementos del enfoque aún existen ciertas limitaciones en la misma.

Primero en lo referente a la comunidad objetivo, la limitación radica en el conocimiento previo que se debe tener de la misma. Esto puede observarse al momento de definir una función de recompensa que este alineada con los objetivos de la comunidad.

También al momento de obtener datos históricos de la comunidad, se requiere comprender que tipos de acciones pueden realizar los usuarios y el significado en estas acciones para la comunidad. Por otro lado, en relación con los datos históricos, para nuevas comunidades debe recrearse un proceso de extracción y procesamiento de los datos similar a como se muestra en el caso de estudio en Stack Overflow.

Junto con esta limitación de recrear el proceso de extracción, debe considerarse que la definición del mapeo de acciones de la comunidad a la estrategia de ludificación continúa siendo de forma manual. Esto significa que se requieren suficientes conocimientos previos de sobre la estrategia de ludificación y la comunidad para realizar esta asociación entre acciones.

Sobre la simulación en sí, tomando de ejemplo el caso de Metagame, la misma resulta particularmente costosa en tiempo de ejecución. Por este

motivo, en casos sensibles al tiempo de ejecución, el uso de simulaciones podría no ser aceptable.

Además, en caso de querer compararse otra estrategia de ludificación la misma requeriría ser simulada en primera instancia. Por último, estas simulaciones no tienen en cuenta el factor visual el cual puede ser un gran aliciente o un gran problema para el uso de la estrategia de ludificación.

8.3. Trabajo futuro

A partir de este trabajo de tesis, han surgido múltiples líneas de investigación posibles dadas las problemáticas que aquí no fueron exploradas, así como las posibilidades de extensión y continuación de los desarrollos aquí realizados.

Como se mencionó anteriormente en las limitaciones de este enfoque, la configuración de la tabla de mapeo entre acciones de la CCC objetivo y las acciones de la estrategia de ludificación a simular presentan la problemática de que actualmente se realizan de forma manual. Esto genera la necesidad de poseer conocimiento sobre ambas partes y la posible relación entre ellas. Sería de interés poder evitar esta necesidad de conocimiento, así como el proceso manual de crear ese mapeo de información.

Para ello se necesitaría conocer todas las posibles acciones de la comunidad objetivo y de la estrategia de ludificación así como que representan. Esto podría realizarse desde aplicando técnicas de la web semántica con el uso de ontologías hasta el uso de procesamiento del lenguaje natural para definir esas representaciones. Ya con esa representación, quedaría evaluar cómo realizar la posible asociación entre esas acciones para lograr una correcta integración entre la comunidad y la ludificación.

Además, el framework fue desarrollado como tal con el objetivo de poder ser extendido en múltiples aspectos. Uno de los aspectos que ya se consideraron parcialmente en este trabajo en el Capítulo 7, es el desarrollo de nuevas recompensas y estados para transmitir al agente de RL. Sin embargo, aún quedan problemáticas a afrontar este punto, parte de las mismas sería desarrollar un compendio de herramientas generalizadoras que faciliten la normalización de cualquier estado o recompensa creada con el framework. Esto a su vez abre las puertas a considerar la búsqueda de una “meta-configuración” óptima para el sistema de simulación la cual puede seguir lineamientos similares a los introducidos en este trabajo.

También en esta tesis se trabajó sobre una única simulación de una estrategia de ludificación basada en Metagame. Por lo cual, queda abierta la problemática de desarrollar otras simulaciones para estrategias de ludificación preexistentes o nuevas, que sean más adaptables o configurables, sobre

las cuales evaluar también el enfoque y el framework. Esto permitiría evaluar múltiples estrategias de ludificación para estudiar cual se adapta mejor a una comunidad particular. Ese estudio, junto a la posibilidad de aplicar una estrategia de ludificación configurada con este enfoque sobre la versión de usuarios activos de una comunidad, daría lugar a estudiar la adopción por parte de los usuarios y evaluar su evolución.

Finalmente, basándose en el enfoque, pero por fuera de los objetivos de esta tesis. Sería interesante analizar este enfoque de recomendación de configuraciones mediante RL, basándose en simulaciones con datos históricos, sobre la problemática de configurar aplicaciones que no sean pertenecientes al área de ludificación.

Acrónimos

API Application Programming Interface. 46, 48, 84, 116

BO Bayesian Optimization. 43

CCC Comunidades de Construcción de Conocimiento. 1, 2, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 31, 32, 33, 37, 42, 44, 45, 46, 62, 63, 64, 65, 66, 67, 77, 78, 79, 80, 81, 82, 93, 94, 95, 98, 99, 110, 114, 115, 116, 119, 132, 134, 142, 143, 144, 145, 146, 148

CNN Convolutional Neural Network. 40, 41

D-DQN Double DQN. 41, 133, 141, 142

DQN Deep Q Network. 40, 41, 133, 134

DSS Decision Support System. 16, 19, 20, 21, 22

FC Full Conected. 41

GA Genetic Algorithms. 43, 44

GPU Graphics Processing Unit. 85

GYM OpenAI Gym. 2, 82, 83, 85, 87, 92, 133, 141, 142, 143, 146, 147

MDA Models, Dynamics and Aesthetics. 17, 34, 63, 64, 77, 94, 97, 145

MRDM Multirelational Data Mining. 43

RL Reinforcement Learning. 1, 2, 14, 15, 16, 17, 22, 33, 38, 39, 40, 44, 79, 80, 81, 82, 83, 84, 85, 87, 91, 92, 93, 94, 98, 99, 104, 106, 107, 110, 114, 115, 119, 120, 132, 133, 134, 141, 142, 143, 144, 145, 146, 147, 148, 149

SVM Support-vector machine. 44

Glosario

BigQuery BigQuery es un servicio de Google para analítica de datos.. 31, 116, 145

Deep Learning También conocido como aprendizaje profundo, es parte de un conjunto más amplio de métodos de aprendizaje automático basados en asimilar representaciones de datos.. 83

disruptivo Un proceso disruptivo es aquel que produce una ruptura en el desarrollo cotidiano de una actividad.. 2, 79, 81, 146

framework Un framework, entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.. 1, 2, 14, 15, 17, 63, 64, 65, 82, 84, 85, 87, 89, 91, 92, 93, 94, 114, 115, 132, 133, 135, 136, 141, 142, 143, 144, 145, 146, 147, 148, 149

intrusivo Un elemento o proceso intrusivo es aquel por el cual se produce o se percibe una invasión o violación a un elemento o contexto que no le corresponde.. 34, 79

ludificación Es la aplicación de cualquier elemento llevado de juegos o videojuegos a entornos no lúdicos, sin modificar su estructura central.. 1, 2, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 31, 32, 33, 34, 37, 38, 42, 44, 45, 46, 51, 52, 62, 63, 64, 66, 68, 71, 77, 78, 79, 80, 81, 82, 85, 87, 91, 93, 94, 95, 97, 98, 100, 103, 106, 107, 110, 111, 112, 114, 115, 116, 117, 118, 119, 122, 124, 134, 142, 143, 144, 145, 146, 147, 148, 149

ludificación óptima Es la ludificación efectiva para una comunidad que proporciona métodos o principios para ayudar a las personas a lograr sus objetivos.. 13, 33, 42

markups También conocido como lenguaje de marcado es una forma de codificar un documento de texto mediante el uso de etiquetas.. 49

Numpy Es una biblioteca de Python que brinda soporte a cálculos de grandes dimensiones en base a vectores y matrices.. 85

proveniencia La proveniencia o proveniencia de la información es la capacidad de rastrear las fuentes de la información para poder medir la confiabilidad y utilidad de la información para el fin deseado.. 21, 22, 24, 31, 51

script Un script es un documento que contiene instrucciones, escritas en códigos de programación. Estas instrucciones comúnmente no requieren compilación y se utilizan para generar programas relativamente sencillos.. 47, 48, 49

Bibliografía

- [1] B. De Wever, T. Schellens, M. Valcke, and H. Van Keer, “Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review,” *Computers and Education*, vol. 46, no. 1, pp. 6–28, 2006.
- [2] J. R. Saini and A. A. Desai, “A textual analysis of digits used for designing yahoo-group identifiers,” *The IUP Journal of Information Technology*, vol. 6, no. 2, pp. 34–42, 2010.
- [3] J. F. Ortega Soto and F. Ortega, “Wikipedia: A quantitative analysis,” *Doctoral dissertation at University Rey Juan Carlos, Madrid, Spain*, p. 228, 2009.
- [4] G. Stahl, *Group Cognition Computer Support for Building Collaborative Knowledge*. Citeseer, 2006.
- [5] L. P. Robert Jr and D. M. Romero, “The influence of diversity and experience on the effects of crowd size,” *Journal of the Association for Information Science and Technology*, vol. 68, no. 2, pp. 321–332, 2017.
- [6] J. Hamari and J. Koivisto, “Why do people use gamification services?,” *International Journal of Information Management*, vol. 35, no. 4, pp. 419–431, 2015.
- [7] J. Hamari, J. Koivisto, and H. Sarsa, “Does Gamification Work? – A Literature Review of Empirical Studies on Gamification,” in *2014 47th Hawaii International Conference on System Sciences*, pp. 3025–3034, 2014.
- [8] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From game design elements to gamefulness: Defining “gamification”,” in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek ’11, (New York, NY, USA), p. 9–15, Association for Computing Machinery, 2011.
- [9] E. Beza, J. Steinke, J. van Etten, P. Reidsma, C. Fadda, S. Mittra, P. Mathur, and L. Kooistra, “What are the prospects for citizen science

- in agriculture? {Evidence} from three continents on motivation and mobile telephone use of resource-poor farmers,” *PloS One*, vol. 12, no. 5, p. e0175700, 2017.
- [10] A. Domínguez, J. Saenz-De-Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J. J. Martínez-Herráiz, “Gamifying learning experiences: Practical implications and outcomes,” *Computers and Education*, vol. 63, pp. 380–392, 2013.
- [11] B. Burke, “Gamification 2020: What is the future of gamification,” *Gartner, Inc., Nov*, vol. 5, 2012.
- [12] X. Bu, J. Rao, and C.-Z. Xu, “A reinforcement learning approach to online web systems auto-configuration,” in *2009 29th IEEE International Conference on Distributed Computing Systems*, pp. 2–11, IEEE, 2009.
- [13] J. Martin, D. Torres, A. Fernandez, S. Pravisani, and G. Briend, “Using Citizen Science Gamification in Agriculture Collaborative Knowledge Production,” in *Proceedings of the XIX International Conference on Human Computer Interaction - Interacción 2018*, (New York, New York, USA), pp. 1–8, ACM Press, 2018.
- [14] C. J. Devers and R. A. R. Gurung, “Critical Perspective on Gamification in Education,” in *Gamification in Education and Business*, pp. 417–430, Cham: Springer International Publishing, 2015.
- [15] K. Seaborn and D. I. Fels, “Gamification in theory and action: A survey,” *International Journal of Human-Computer Studies*, vol. 74, pp. 14–31, 2015.
- [16] M. K. Geldenhuys, L. Thamsen, K. K. Gontarskay, F. Lorenz, and O. Kao, “Effectively testing system configurations of critical iot analytics pipelines,” in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 4157–4162, Dec 2019.
- [17] M. I. Kellner, R. J. Madachy, and D. M. Raffo, “Software process simulation modeling: why? what? how?,” *Journal of Systems and Software*, vol. 46, no. 2-3, pp. 91–105, 1999.
- [18] F. Lieder and T. Griffiths, “Helping people make better decisions using optimal gamification.,” in *CogSci*, 2016.
- [19] J. Martin, D. Torres, and A. Fernandez, “Patterns of markup use in wikipedia,” in *2017 36th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–9, IEEE, 2017.
- [20] J. Martin, D. Torres, and A. Fernandez, “Optimizing a gamified design through reinforcement learning-a case study in stack overflow,”

- in *Conference on Cloud Computing, Big Data & Emerging Topics*, pp. 89–103, Springer, 2021.
- [21] L. Antonelli, G. Camilleri, C. Challiol, A. Fernández, M. Hozikian, R. Giandini, J. Grigera, A. B. Lliteras, J. Martin, D. Torres, *et al.*, “An extension to scenarios to deal with business cases for the decision-making processes in the agribusiness domain,” in *Agriculture Value Chain-Challenges and Trends in Academia and Industry*, pp. 35–50, Springer, 2021.
- [22] A. Garrido, L. Antonelli, J. Martin, M. Alemany, and J. Mula, “Using lel and scenarios to derive mathematical programming models. application in a fresh tomato packing problem,” *Computers and Electronics in Agriculture*, vol. 170, p. 105242, 2020.
- [23] M. Alemany, A. Estesó, A. Ortiz, J. Hernández, A. Fernández, A. Garrido, J. Martin, S. Liu, G. Zhao, C. Guyon, *et al.*, “A conceptual framework for crop-based agri-food supply chain characterization under uncertainty,” in *Agriculture Value Chain-Challenges and Trends in Academia and Industry*, pp. 19–33, Springer, 2021.
- [24] G. Fick and R. H. Sprague, “Decision support systems : a research perspective,” *IIASA Proceedings*, pp. 23–44, 1980.
- [25] S. Neumann and M. Hadass, “Dss and strategic decisions,” *California Management Review*, vol. 22, no. 3, pp. 77–84, 1980.
- [26] V. L. Sauter, *Decision support systems for business intelligence*. John Wiley & Sons, 2014.
- [27] R. Sharda, D. Delen, E. Turban, J. Aronson, and T. Liang, “Business intelligence and analytics,” *System for Decesion Support*, 2014.
- [28] C. López de Munain, M. Torrent, F. Molina, S. Schanz, A. Sandoval, J. P. Sfligoy, and M. L. Sassaroli, “Sistemas de apoyo a la toma de decisiones,” in *XVI Workshop de Investigadores en Ciencias de la Computación*, 2014.
- [29] J. Schultz, “A history of the promis technology: an effective human interface,” in *A history of personal workstations*, pp. 439–488, Reading, Mass: AMC Press, 1988.
- [30] K. B. Waghlikar, V. Sundararajan, and A. W. Deshpande, “Modeling paradigms for medical diagnostic decision support: a survey and future directions,” *Journal of medical systems*, vol. 36, no. 5, pp. 3029–3049, 2012.
- [31] A. G. Mohapatra and S. K. Lenka, “Neuro-fuzzy-based smart dss for crop specific irrigation control and sms notification generation for pre-

- cision agriculture,” *International Journal of Convergence Computing*, vol. 2, no. 1, pp. 3–22, 2016.
- [32] J. Jones, G. Tsuji, G. Hoogenboom, L. Hunt, P. Thornton, P. Wilkens, D. Imamura, W. Bowen, and U. Singh, “Decision support system for agrotechnology transfer: Dssat v3,” in *Understanding options for agricultural production*, pp. 157–177, Springer, 1998.
- [33] D. Torres, H. Skaf-Molli, P. Molli, and A. Diaz, “Discovering Wikipedia Conventions Using DBpedia Properties,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9507, pp. 115–144, 2016.
- [34] A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, and S. Reiterer, “Toward the next generation of recommender systems: applications and research challenges,” in *Multimedia services in intelligent environments*, pp. 81–98, Springer, 2013.
- [35] M. Zhang, R. Ranjan, S. Nepal, M. Menzel, and A. Haller, “A declarative recommender system for cloud infrastructure services selection,” in *International Conference on Grid Economics and Business Models*, pp. 102–113, Springer, 2012.
- [36] V. L. Sauter, *Decision Support Systems for Business Intelligence*. Wiley, 2011.
- [37] C. Madge, J. Meek, J. Wellens, and T. Hooley, “Facebook , social integration and informal learning at university: ‘It is more for socialising and talking to friends about work than for actually doing work’,” *Learning, Media and Technology*, vol. 34, no. 2, pp. 141–155, 2009.
- [38] O. Hartig, “Provenance information in the web of data.,” *LDOW*, vol. 538, 2009.
- [39] G. Barbier and H. Liu, “Information provenance in social media,” in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pp. 276–283, Springer, 2011.
- [40] F. Orlandi and A. Passant, “Modelling provenance of DBpedia resources using Wikipedia contributions,” *Journal of Web Semantics*, vol. 9, no. 2, pp. 149–164, 2011.
- [41] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. MIT press, 2015.
- [42] M. Alemany, A. Boza, A. Ortiz, and V. S. Fuertes-Miquel, “Configurable dss for uncertainty management by fuzzy sets,” *Procedia Computer Science*, vol. 83, pp. 1019–1024, 2016.

-
- [43] M. Scardamalia and C. Bereiter, “Computer support for knowledge-building communities,” *The journal of the learning sciences*, vol. 3, no. 3, pp. 265–283, 1994.
- [44] R. G. Baraniuk, C. S. Burrus, D. H. Johnson, and D. L. Jones, “Sharing knowledge and building communities in signal processing,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 10–16, 2004.
- [45] J. Moskaliuk, J. Kimmerle, and U. Cress, “Wiki-supported learning and knowledge building: effects of incongruity between knowledge and information,” *Journal of Computer Assisted Learning*, vol. 25, pp. 549–561, 11 2009.
- [46] O. Ferschke, *The quality of content in open online collaboration platforms: Approaches to NLP-supported information quality management in Wikipedia*. PhD thesis, Technische Universität, 2014.
- [47] D. Yang, A. Halfaker, R. Kraut, and E. Hovy, “Edit categories and editor role identification in wikipedia,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pp. 1295–1299, 2016.
- [48] S. Javanmardi and C. Lopes, “Statistical measure of quality in Wikipedia,” *Proceedings of the First Workshop on Social Media Analytics - SOMA ’10*, pp. 132–138, 2010.
- [49] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, “Discovering value from community activity on focused question answering sites: a case study of stack overflow,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 850–858, 2012.
- [50] D. Yang, A. Halfaker, R. Kraut, and E. Hovy, “Who did what: Editor role identification in wikipedia,” in *Tenth International AAAI Conference on Web and Social Media*, 2016.
- [51] T. H. McCormick, R. Ferrell, A. F. Karr, and P. B. Ryan, “Modeling User Reputation in Wikis,” *Science And Technology*, vol. 4, no. 5, pp. 497–511, 2010.
- [52] R. S. R. Geiger and A. Halfaker, “Using edit sessions to measure participation in Wikipedia,” *Proceedings of the 2013 conference on . . .*, no. February 2013, pp. 861–869, 2013.
- [53] A. Pal, S. Chang, and J. A. Konstan, “Evolution of experts in question answering communities,” in *sixth international AAAI conference on weblogs and social media*, 2012.

- [54] A. Halfaker, R. S. Geiger, J. T. Morgan, and J. Riedl, “The Rise and Decline of an Open Collaboration System: How Wikipedia’s Reaction to Popularity Is Causing Its Decline,” *American Behavioral Scientist*, vol. 57, no. 5, pp. 664–688, 2012.
- [55] J. Kiesel, M. Potthast, M. Hagen, and B. Stein, “Spatio-temporal analysis of reverted wikipedia edits,” in *Eleventh International AAAI Conference on Web and Social Media*, 2017.
- [56] P. Fugelstad, P. Dwyer, J. Filson Moses, J. Kim, C. A. Mannino, L. Terveen, and M. Snyder, “What makes users rate (share, tag, edit...)? predicting patterns of participation in online communities,” in *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pp. 969–978, 2012.
- [57] R. Schenkel, F. M. Suchanek, and G. Kasneci, “YAWN: A Semantically Annotated Wikipedia XML Corpus,” *Proceedings of GIFAchtagung für Datenbanksysteme in Business Technologie und Web BTW2007*, vol. 103, pp. 277–291, 2007.
- [58] M. Strube and S. P. Ponzetto, “WikiRelate! Computing semantic relatedness using Wikipedia,” *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 2, p. 1419, 2006.
- [59] S. P. Ponzetto and M. Strube, “Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution,” in *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pp. 192–199, 2006.
- [60] J. E. Gerow, R. Ayyagari, J. B. Thatcher, and P. L. Roth, “Can we have fun@ work? the role of intrinsic motivation for utilitarian systems,” *European Journal of Information Systems*, vol. 22, no. 3, pp. 360–380, 2013.
- [61] H. Sun and P. Zhang, “Causal relationships between perceived enjoyment and perceived ease of use: An alternative approach,” *Journal of the Association for Information Systems*, vol. 7, no. 1, p. 24, 2006.
- [62] J. Hamari and L. Keronen, “Why do people play games? a meta-analysis,” *International Journal of Information Management*, vol. 37, no. 3, pp. 125–141, 2017.
- [63] F. D. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS quarterly*, pp. 319–340, 1989.
- [64] H. Van der Heijden, “User acceptance of hedonic information systems,” *MIS quarterly*, pp. 695–704, 2004.

- [65] J. Koivisto and J. Hamari, “The rise of motivational information systems: A review of gamification research,” *International Journal of Information Management*, vol. 45, pp. 191–210, 2019.
- [66] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.
- [67] P. Buckley, S. Noonan, C. Geary, T. Mackessy, and E. Nagle, “An Empirical Study of Gamification Frameworks,” *Journal of Organizational and End User Computing*, vol. 31, pp. 22–38, jan 2019.
- [68] N. Xi and J. Hamari, “Does gamification satisfy needs? A study on the relationship between gamification features and intrinsic need satisfaction,” *International Journal of Information Management*, vol. 46, pp. 210–221, jun 2019.
- [69] T. Jaguš, I. Botički, and H.-J. So, “Examining competitive, collaborative and adaptive gamification in young learners’ math learning,” *Computers & education*, vol. 125, pp. 444–457, 2018.
- [70] E. Orta and M. Ruiz, “A simulation and gamification approach for IT service management improvement,” in *International Conference on Software Process Improvement and Capability Determination*, pp. 84–97, Springer, 2016.
- [71] A. Mora, D. Riera, C. González, and J. Arnedo-Moreno, “Gamification: a systematic review of design frameworks,” *Journal of Computing in Higher Education*, vol. 29, no. 3, pp. 516–548, 2017.
- [72] J. Harms, C. Wimmer, K. Kappel, and T. Grechenig, “Gamification of online surveys: conceptual foundations and a design process based on the mda framework,” in *Proceedings of the 8th Nordic conference on human-computer interaction: Fun, fast, foundational*, pp. 565–568, 2014.
- [73] B. K. Neeli, “Gamification in the enterprise: Differences from consumer market, implications, and a method to manage them,” in *Gamification in education and business*, pp. 489–511, Springer, 2015.
- [74] K. Robson, K. Plangger, J. H. Kietzmann, I. McCarthy, and L. Pitt, “Is it all a game? understanding the principles of gamification,” *Business horizons*, vol. 58, no. 4, pp. 411–420, 2015.
- [75] U. Ruhi, “Level up your strategy: Towards a descriptive framework for meaningful enterprise gamification,” *Technology Innovation Management Review*, 2015.

- [76] A. C. T. Klock, L. F. Da Cunha, M. F. de Carvalho, B. E. Rosa, A. J. Anton, and I. Gasparini, “Gamification in e-learning systems: A conceptual model to engage students and its application in an adaptive e-learning system,” in *International Conference on Learning and Collaboration Technologies*, pp. 595–607, Springer, 2015.
- [77] A. Mora, P. Zaharias, C. González, and J. Arnedo-Moreno, “Fraggle: a framework for agile gamification of learning experiences,” in *International Conference on Games and Learning Alliance*, pp. 530–539, Springer, 2015.
- [78] R. Hunicke, M. LeBlanc, and R. Zubek, “MDA: A Formal Approach to Game Design and Game Research,” *Workshop on Challenges in Game AI*, pp. 1–4, 2004.
- [79] E. Atighi Lorestani, M. Khalili, *et al.*, “How lighting dynamics create social interactive game,” in *DS 88: Proceedings of the 19th International Conference on Engineering and Product Design Education (E&PDE17), Building Community: Design Education for a Sustainable Future, Oslo, Norway, 7 & 8 September 2017*, pp. 680–685, 2017.
- [80] S. Pravisani, “Metagame,” in *I Workshop de Ciencia Abierta y Ciudadana-Argentina (La Plata, 2017)*, 2017.
- [81] N. R. Prestopnik and J. Tang, “Points, stories, worlds, and diegesis: Comparing player experiences in two citizen science games,” *Computers in Human Behavior*, vol. 52, pp. 492–506, 2015.
- [82] S. Cordero-Brito and J. Mena, “Gamification in the Social Environment,” in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM’18*, (New York, New York, USA), pp. 640–643, ACM Press, 2018.
- [83] D. N. Crowley, J. G. Breslin, P. Corcoran, and K. Young, “Gamification of citizen sensing through mobile social reporting,” in *2012 IEEE International Games Innovation Conference*, pp. 1–5, IEEE, 2012.
- [84] J. P. Cohn, “Citizen science: Can volunteers do real research?,” *BioScience*, vol. 58, no. 3, pp. 192–197, 2008.
- [85] J. Pellicer Marquès, *Wiki y ludificación vinculados a un entorno de diagnóstico para la imagen*. PhD thesis, Universitat Oberta de Catalunya, 2018.
- [86] S. Narayan, J. Orlowitz, J. Morgan, B. M. Hill, and A. Shaw, “The Wikipedia Adventure: Field Evaluation of an Interactive Tutorial for New Users,” in *Proceedings of the 20th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW’17)*, 2017.

-
- [87] L. Alisandra Senabre, *Gamification Quest: Design and Development of a gamification game*. PhD thesis, Universitat Jaume I, 2017.
- [88] T.-Y. Mu, A. Al-Fuqaha, and K. Salah, “Automating the configuration of mapreduce: A reinforcement learning scheme,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
- [89] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*. MIT Press, 1998.
- [90] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [91] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.
- [92] Y. Li, “Deep reinforcement learning: An overview,” *arXiv preprint arXiv:1701.07274*, 2017.
- [93] L.-J. Lin, “Self-improving reactive agents based on reinforcement learning, planning and teaching,” *Machine learning*, vol. 8, no. 3-4, pp. 293–321, 1992.
- [94] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The arcade learning environment: An evaluation platform for general agents,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.
- [95] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, feb 2015.
- [96] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Thirtieth AAAI conference on artificial intelligence*, 2016.
- [97] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” *arXiv preprint arXiv:1511.05952*, 2015.
- [98] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, “Dueling network architectures for deep reinforcement learning,” *arXiv preprint arXiv:1511.06581*, 2015.
- [99] O. Anschel, N. Baram, and N. Shimkin, “Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 176–185, JMLR. org, 2017.

-
- [100] F. S. He, Y. Liu, A. G. Schwing, and J. Peng, “Learning to play in a day: Faster deep reinforcement learning by optimality tightening,” *arXiv preprint arXiv:1611.01606*, 2016.
- [101] B. O’Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih, “Pgc: Combining policy gradient and q-learning,” *arXiv preprint arXiv:1611.01626*, 2016.
- [102] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, “Action-conditional video prediction using deep networks in atari games,” in *Advances in neural information processing systems*, pp. 2863–2871, 2015.
- [103] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, “Deep exploration via bootstrapped dqn,” in *Advances in neural information processing systems*, pp. 4026–4034, 2016.
- [104] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, “Deep q-learning from demonstrations,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [105] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI Gym,” *arXiv preprint arXiv:1606.01540*, 6 2016.
- [106] F. Lieder and T. L. Griffiths, “Helping people make better decisions using optimal gamification,” *CogSci*, 2016.
- [107] S. T. Allen, M. Jankowski, and P. Pathirana, *Storm Applied: Strategies for real-time event processing*. Manning Publications Co., 2015.
- [108] N. Padhy and R. Panigrahi, “Multi relational data mining approaches: A data mining technique,” *arXiv preprint arXiv:1211.3871*, 2012.
- [109] M. Bilal and M. Canini, “Towards automatic parameter tuning of stream processing systems,” in *Proceedings of the 2017 Symposium on Cloud Computing*, pp. 189–200, 2017.
- [110] L. Fischer, S. Gao, and A. Bernstein, “Machines tuning machines: Configuring distributed stream processors with bayesian optimization,” in *2015 IEEE International Conference on Cluster Computing*, pp. 22–31, IEEE, 2015.
- [111] M. Trotter, G. Liu, and T. Wood, “Into the storm: Descrying optimal configurations using genetic algorithms and bayesian optimization,” in *2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W)*, pp. 175–180, IEEE, 2017.

-
- [112] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, “Taking the human out of the loop: A review of bayesian optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [113] P. Jamshidi and G. Casale, “An uncertainty-aware approach to optimal configuration of stream processing systems,” in *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 39–48, IEEE, 2016.
- [114] C. E. Rasmussen and H. Nickisch, “Gaussian processes for machine learning (gpml) toolbox,” *Journal of machine learning research*, vol. 11, no. Nov, pp. 3011–3015, 2010.
- [115] M. Trotter, T. Wood, and J. Hwang, “Forecasting a storm: Divining optimal configurations using genetic algorithms and supervised learning,” in *2019 IEEE International Conference on Autonomic Computing (ICAC)*, pp. 136–146, IEEE, 2019.
- [116] M. D. McKay, R. J. Beckman, and W. J. Conover, “Comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [117] L. M. Vaquero and F. Cuadrado, “Auto-tuning distributed stream processing systems using reinforcement learning,” *arXiv preprint arXiv:1809.05495*, 2018.
- [118] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, “Optimal and autonomous control using reinforcement learning: A survey,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.
- [119] A. Fadhil and A. Villafiorita, “An Adaptive Learning with Gamification & Conversational UIs,” in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17*, (New York, New York, USA), pp. 408–412, ACM Press, 2017.
- [120] A. Knutas, R. van Roy, T. Hynninen, M. Granato, J. Kasurinen, and J. Ikonen, “Profile-Based Algorithm for Personalized Gamification in Computer-Supported Collaborative Learning Environments,” *GHITALY@CHIItaly*, 2017.
- [121] O. Ferschke, T. Zesch, and I. Gurevych, “Wikipedia revision toolkit: efficiently accessing Wikipedia’s edit history,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pp. 97–102, 2011.

-
- [122] H. Dohrn and D. Riehle, “Design and implementation of the swoble wikitext parser: unlocking the structured data of wikipedia,” in *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pp. 72–81, 2011.
- [123] N. Zumel and J. Mount, *Practical data science with R*. Manning Publications Co., 2014.
- [124] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. I. Perona, “An extensive comparative study of cluster validity indices,” *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, 2013.
- [125] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [126] L. M. Sheikh, B. Tanveer, M. A. Hamdani, S. Mustafa, and A. Hamdani, “Interesting Measures for Mining Association Rules,” *Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International*, pp. 641–644, 2004.
- [127] S. Rafaeli, T. Hayat, and Y. Ariel, “Knowledge building and motivations in wikipedia: Participation as “ba”,” in *Cyberculture and new media*, pp. 51–67, Brill Rodopi, 2009.
- [128] D. Geiger, M. Rosemann, E. Felt, and M. Schader, “Crowdsourcing Information Systems - Definition, Typology, and Design,” *ICIS*, 2012.
- [129] C. D. Sciascio, D. Strohmaier, M. Errecalde, and E. Veas, “Interactive Quality Analytics of User-generated Content,” *ACM Transactions on Interactive Intelligent Systems*, vol. 9, pp. 1–42, mar 2019.
- [130] S. Velichety, S. Ram, and J. Bockstedt, “Quality Assessment of Peer-Produced Content in Knowledge Repositories using Development and Coordination Activities,” *Journal of Management Information Systems*, vol. 36, pp. 478–512, apr 2019.
- [131] B. Morschheuser, A. Maedche, and D. Walter, “Designing cooperative gamification: conceptualization and prototypical implementation,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 2410–2421, 2017.
- [132] D. Torres, P. Molli, H. Skaf-Molli, and A. Diaz, “Improving wikipedia with dbpedia,” in *Proceedings of the 21st International Conference on World Wide Web*, pp. 1107–1112, ACM, 2012.
- [133] D. Torres, H. Skaf-Molli, P. Molli, and A. Díaz, “Discovering wikipedia conventions using DBpedia properties,” in *International Workshop on*

- Semantic Web Collaborative Spaces*, pp. 115–144, Springer International Publishing, 2013.
- [134] J. Lehmann, M. Lalmas, E. Yom-Tov, and G. Dupret, “Models of user engagement,” in *International Conference on User Modeling, Adaptation, and Personalization*, pp. 164–175, Springer, 2012.
- [135] J. Radoff, *Game on: Energize your business with social media games*. John Wiley & Sons, 2011.
- [136] M. Plappert, “keras-rl.” <https://github.com/keras-rl/keras-rl>, 2016.
- [137] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, “Stable baselines3.” <https://github.com/DLR-RM/stable-baselines3>, 2019.
- [138] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 2011.
- [139] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, “Stable baselines.” <https://github.com/hill-a/stable-baselines>, 2018.
- [140] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, “Openai baselines.” <https://github.com/openai/baselines>, 2017.
- [141] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [142] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.
- [143] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” 2013.
- [144] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” 2016.

- [145] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017.