FACULTAD DE INFORMÁTICA | UNIVERSIDAD NACIONAL DE LA PLATA

TESINA DE LICENCIATURA

Titulo: Desarrollo de Buenas Prácticas a partir de la Gamificación
Autores: Vannasaeng, Franco – Rodríguez, Facundo
Director: Esponda, Silvia
Codirector: Pasini, Ariel
Asesor profesional: -
Carrera: Licenciatura en Informática – Licenciatura en Sistemas

Resumen

El trabajo plantea la realización de una componente de accesible inserción en herramientas de Gestión de Proyectos, donde se propone un juego con la finalidad de incentivar a los integrantes del grupo desarrollador de software a seguir las buenas prácticas descritas en la norma ISO 29110, dando recompensas de algún tipo.

El desarrollo del componente propuesto se realizará utilizando mecánicas y dinámicas de juego para adquirir hábitos y alcanzar objetivos, lo que se conoce como Gamificación.

El juego, propuesto en este proyecto, es una manera de acercar el conocimiento de una forma diferente, en donde el aspecto lúdico resulta esencial en situaciones de aprendizaje.

En conclusión, el trabajo tiene como objetivo la incentivación, concientización y el aprendizaje del uso de las mejores prácticas del desarrollo de software a través de una aplicación lúdica.

Palabras Claves

Gestión de Proyecto - PyMEs - ISO 29110 – Buenas Prácticas - Roles - Gamificación - Mecánicas de juego - Herramientas de Gestión - Trello - Android

Conclusiones

Se realizó un estudio de la norma ISO 29110 y del concepto de Gamificación.
Se realizó un análisis de diferentes herramientas de Gestión de Proyectos y se seleccionó Trello para la integración con Flip-Flop.
Se desarrolló una componente lúdica utilizando las mecánicas y dinámicas de juego para incentivar la utilización de buenas prácticas, en el desarrollo de software.

Trabajos Realizados

Desarrollo de una aplicación nativa para el sistema operativo Android, llamada Flip-Flop, orientada a grupos de desarrollo de software de Pequeñas y Medianas Empresas (PyMEs), que incentiva la aplicación de buenas prácticas, propuestas en la norma ISO 29110.

Trabajos Futuros

- *Flip-Flop Multiplataforma: desarrollar una versión Web, como así también versiones para iOS y Windows Mobile.*
- *Funciones adicionales para el Líder Técnico: brindar funcionalidades que permitan analizar el desempeño de los jugadores.*
- *Configuración inicial automatizada: permitir crear un tablero en Trello válido para Flip-Flop utilizando su API.*

Fecha de la presentación: JUNIO 2016

INDICE

Capítulo 1 - Introducción

1.1 Objetivos y desarrollo propuesto

Para la presente tesina se propone desarrollar una componente, llamada Flip-Flop, que utilice métodos lúdicos para ser aplicada en herramientas de Gestión, orientada a grupos de desarrollos de software de Pequeñas y Medianas Empresas (PyMEs), que incentive la aplicación de buenas prácticas, propuestas en la norma ISO 29110 [1.1].
Se definen los siguientes objetivos específicos:

1. Estudiar la norma ISO 29110, en particular las prácticas/actividades propuestas para el Proceso de Implementación de Software.
2. Investigar el concepto Gamificación definido como el uso de técnicas y dinámicas propias de los juegos y el ocio en actividades no recreativas. Estudiar sus mecánicas y dinámicas de juego para ser utilizadas en la componente a realizar.
3. Explorar las herramientas de Gestión de Proyectos disponibles para seleccionar la más conveniente de uso para realizar la comunicación con la componente en cuestión.
4. Desarrollar una componente utilizando las mecánicas y dinámicas del juego para incentivar la utilización de buenas prácticas, en el desarrollo de software.

1.2 Motivación

La ISO 29110 es una norma creada para PyMEs dedicadas al desarrollo de software, que permite desarrollar proyectos utilizando guías de gestión de proyectos y desarrollo de software, incrementando así las probabilidades de éxito en función de costo, tiempo y alcance, ayudando a las empresas a percibir y adecuarse a los cambios tecnológicos.

La implementación de esta norma aumenta la competitividad y optimización para desarrollar un proyecto de software, con mayor control en cada etapa. Impacta directamente en la calidad de software mejorando la imagen de la empresa y reduce riesgos en el desarrollo.

Es un hecho habitual que, por cuestiones de tiempo, fallas en la comunicación, pocas reuniones con clientes, y otras tantas situaciones comunes en la mayoría de equipos de desarrollo de software, el hecho de seguir las mejores prácticas, no siempre es posible o, en otros casos, simplemente se carece del conocimiento de las mismas y de sus beneficios. Por esta razón se ve la necesidad de lograr una herramienta que incentive e inculque a sus usuarios sobre el uso de las mejores prácticas. Para lograr esto la idea de incorporar un juego con bases en la Gamificación nos resulta ideal.

El objetivo general de la Gamificación es influir y motivar a los usuarios a conseguir que logren la adquisición de hábitos y alcanzar objetivos. Para ello, se incentiva al jugador a participar, compartir e interactuar en alguna actividad de forma individual o con la comunidad. Una Gamificación eficaz, dinámica y con una rica experiencia puede ser utilizada para llevar a cabo una variedad de objetivos en todo tipo de ámbitos y sectores.

El trabajo plantea la realización de una componente de accesible inserción en herramientas de Gestión de Proyectos [1.2], donde se propone un juego con la finalidad de incentivar a los integrantes del grupo desarrollador de software a seguir las buenas prácticas descritas en la norma ISO 29110, dando recompensas de algún tipo.

La norma ofrece procesos de gestión y de implementación de proyectos cuyo propósito es establecer y llevar a cabo de forma sistemática las tareas del proceso de implementación de software, lo que permite cumplir con los objetivos del proyecto con la calidad, tiempo y costos esperados; y la realización sistemática de análisis, diseño, actividades de construcción, integración y pruebas de productos nuevos o modificados de software de acuerdo a la especificación de requisitos.

Para el desarrollo del componente propuesto se emplearán mecánicas y dinámicas de juego para adquirir hábitos y alcanzar objetivos, lo que se conoce como Gamificación [1.3]. Dicho concepto hace referencia a la aplicación de mecánicas de juego [1.4] en entornos no lúdicos, es decir, distintas acciones, comportamientos, técnicas y mecanismos de control que se utilizan para convertir en juego una actividad. Se trata de los aspectos que, en conjunto, crean una experiencia atractiva y fácil de adhesión para el jugador.

La idea de plantear el desarrollo de un juego, como el que se propone en este proyecto, es una manera de acercar el conocimiento de una forma diferente, en donde el aspecto lúdico resulta esencial en situaciones de aprendizaje.

En conclusión, el trabajo tendrá como objetivo la incentivación, concientización y el aprendizaje respecto del uso de las mejores prácticas del desarrollo de software a través de una aplicación lúdica.

Se comienza el *Capítulo 2* describiendo los conceptos de Calidad de Proceso y Gestión de Proyecto para luego describir la norma ISO 29110 incluyendo las prácticas/actividades propuestas y los roles que se definen para las mismas. En el *Capítulo 3* se investiga el concepto de Gamificación y se describen las mecánicas de juego que se implementan en la componente lúdica propuesta. A partir del *Capítulo 4* se describe Flip-Flop incluyendo las interfaces más importantes del juego y una completa documentación de las funcionalidades desarrolladas. También se detallan las condiciones y componentes

necesarios para poder utilizar Flip-Flop resaltando principalmente, la herramienta de gestión elegida, Trello [1.5]. Se continúa en el *Capítulo 5* con los detalles de implementación. Aquí se hacen presentes las diferentes tecnologías y servicios utilizados durante el desarrollo y que forman parte de componente final. Se abordan temas que abarcan desde los lenguajes de programación utilizados, la persistencia de los datos, la comunicación entre las aplicaciones y la integración con los servicios de juego de Google[1.6] utilizados para obtener una experiencia de juego que incentive a los jugadores. En el *Capítulo 6* se realiza un informe sobre el caso de estudio llevado a cabo. La finalidad del mismo fue probar el juego en diferentes dispositivos y que los participantes no tengan dificultades para poder interactuar con el juego, que fácilmente capten el objetivo y que puedan jugarlo más allá de los puntajes obtenidos. Por último, en el *Capítulo 7*, se enumeran una serie de mejoras/extensiones que los tesistas consideran apropiadas para ser realizadas en trabajos futuros. Se finaliza con una conclusión en el *Capítulo 8* acerca de la presente tesina.

Capítulo 2 - Descripción de Calidad de Proceso y Gestión de Proyecto

La mejora de procesos en una organización se define como un conjunto de tareas llevadas adelante para obtener productos o servicios de mejor calidad a partir de la revisión y adaptación de sus procesos.

Si se ve a la organización como una empresa prestadora de servicios, el concepto de calidad está directamente relacionado a la obtención de la satisfacción de las necesidades del cliente. En consecuencia, el nivel de calidad de la organización estará ligado al nivel de calidad con que se preste el servicio. Cuanto más ordenadas y predecibles sean las organizaciones para la prestación de sus servicios, mayores serán sus niveles de calidad. Para lograr este efecto de previsibilidad, las organizaciones fueron definiendo los procesos de cada proyecto. Cuando estos procesos se hicieron extensibles, se convirtieron en estándares y la utilización de los mismos garantiza la calidad de los productos o servicios prestados [2.1].

Las organizaciones prestadoras de servicios de software, apuntan a la predictibilidad con la definición de las etapas del ciclo de vida y estableciendo los procesos necesarios para que cada etapa sea realizada según lo planificado.

Estos procesos deben ser lo suficientemente ordenados para obtener la confiabilidad en las estimaciones y además ser lo suficientemente flexibles para adaptarse rápidamente a los cambios que, particularmente, la industria del software requiere. Este es el gran desafío que hoy enfrenta este tipo de organizaciones.

En los últimos años y en busca de esta previsibilidad, que garantiza la calidad de los productos y servicios desarrollados, diferentes organismos relacionados con la industria del software han construido modelos y normas certificables.

2.1 Modelos para PyMEs

La aplicación de los modelos para PyMEs requiere una importante inversión, que se será recuperada a largo plazo, por lo que muchas veces se las considera como un gasto y no una inversión, sin percibir a la calidad como un valor en sí mismo que hará prosperar a la organización. En el caso de las PyMEs, la posibilidad de utilizar este tipo de instrumentos es mucho más dificultosa por el tamaño reducido de los grupos de trabajo, por los problemas de definición de roles dentro de estos grupos, por las responsabilidades poco precisas y por los recursos restringidos con que cuentan este tipo de organización. Debido a esto, aparecen modelos directamente orientados a este tipo de organizaciones como por ejemplo: MoProSoft [2.12] en México o MPS BR [2.13] en Brasil, y COMPETISOFT como un proyecto Iberoamericano [2.14] y la más reciente ISO 29110 que se establece como estándar en las Mejoras de Procesos para PyMEs.

2.2 La norma ISO 29110

La norma ISO 29110 "Perfiles de ciclo de vida para pequeñas organizaciones" tiene por objeto resolver los problemas señalados anteriormente y aborda las necesidades específicas de las pequeñas empresas para hacer frente a los problemas de pobre adopción de estándares de pequeñas empresas. Los requisitos básicos de un proceso de desarrollo de software son que debe adaptarse a las necesidades del proyecto y contribuir al éxito del mismo. La característica central de las entidades a las que apunta la norma ISO 29110 es el tamaño, sin embargo, hay otros aspectos y características de las pequeñas empresas que pueden afectar el proceso de desarrollo, tales como: modelos de negocio (comercial, contratación, desarrollo interno, etc.); los factores situacionales (como la criticidad, ambiente de incertidumbre, etc.); y niveles de riesgo.
El núcleo de esta norma es la Guía de Gestión e Ingeniería (ISO 29110-5 [2.15]) que se centra en la Gestión de Proyectos e Implementación de Software.

2.2.1 Proceso de Gestión de Proyectos

Dentro de la norma ISO 29110, el objetivo del proceso de Gestión de Proyectos es establecer y llevar a cabo de manera sistemática las tareas del proceso de implementación de software, lo que permite cumplir con los objetivos del proyecto con la calidad, tiempo y costos esperados. El proceso de Gestión de Proyectos está destinado a las PyMEs que deseen establecer procesos para implementar cualquier enfoque de desarrollo o metodología (desarrollo basado en pruebas, desarrollo ágil, desarrollo evolutivo, desarrollo incremental, etc.) basados en la organización o en las necesidades del proyecto. La *Imagen 2.1* muestra el flujo de información entre las actividades del Proceso de Gestión de Proyectos, incluyendo los elementos de trabajo más relevantes y su relación.

<u>Objetivos del proceso de Gestión de Proyecto</u>

1. El *Plan de Proyecto* para la ejecución del proyecto se desarrolla de acuerdo con la *Declaración de Trabajo* y es revisado y aceptado por el cliente. Se estiman las tareas y los recursos necesarios para completar el trabajo.
2. El avance del proyecto es monitoreado contra el *Plan del Proyecto* y se registra en el *Registro de Estado de Progreso*. Se hacen correcciones para remediar problemas y desviaciones del plan cuando los objetivos del proyecto no son alcanzados.
   El cierre del proyecto se lleva a cabo para obtener la aceptación del cliente y se documenta en el *Registro de Aceptación*.

3. Las *Solicitudes de Cambio* se abordan a través de su recepción y análisis. Los cambios en los requisitos de software son evaluados por el costo, tiempo e impacto técnico.
4. Se llevan a cabo reuniones de revisión con el equipo de trabajo y el cliente. Los acuerdos son registrados y monitoreados.
5. Los riesgos se identifican en el desarrollo y durante la realización del proyecto.
6. Un software de control de versiones es desarrollado, donde se identifican y definen los elementos de *Configuración de Software*. Las versiones de los elementos son controladas y puestas a disposición del cliente y de equipo de trabajo.
7. El aseguramiento de la calidad de software se realiza para dar garantías de que los productos y los procesos de trabajo cumplen con la especificación del *Plan de Proyecto* y los requisitos del proyecto.

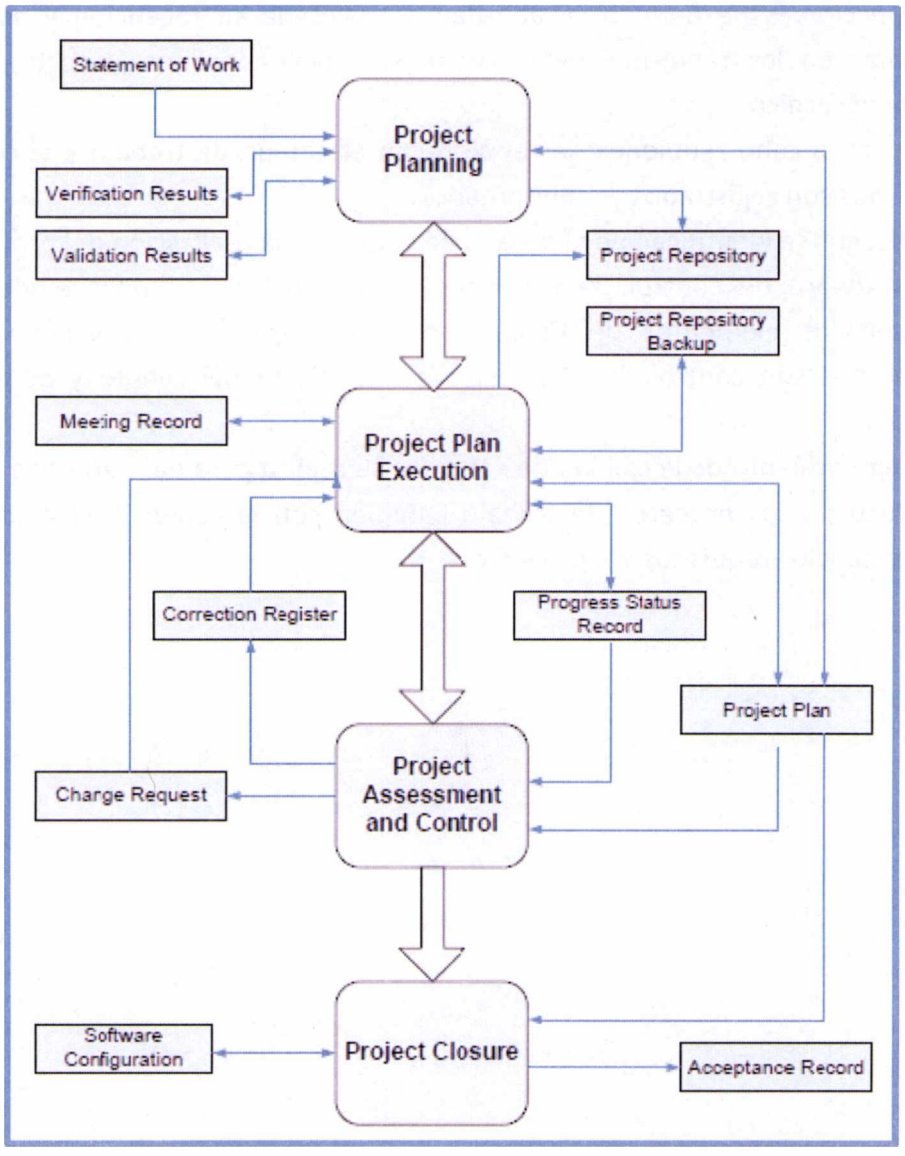

Imagen 2.1. Diagrama del proceso de Gestión de Proyectos.

Actividades del proceso de Gestión de Proyecto

PM.1 Planificación del Proyecto

El objetivo principal de este proceso es producir y comunicar planes eficaces y viables de proyectos. Este proceso determina el ámbito de la gestión de proyectos y actividades técnicas, identifica las salidas de procesos, tareas y entregables del proyecto, establece horarios para la conducta de tareas del proyecto, incluidos los criterios de rendimiento y los recursos necesarios para llevar a cabo las tareas del proyecto.

Rol	Lista de tareas
PM **TL**	Revisión de la *Declaración del Trabajo*
PM **CUS**	Definir con el cliente las instrucciones de entrega de cada uno de los entregables especificados en la *Declaración de Trabajo*.
PM **TL**	Identificar las tareas concretas que se realizarán con el fin de producir los entregables y los *Componentes de Software* identificados en la *Declaración de Trabajo*. Incluir tareas en el proceso de SI, junto con la verificación, validación y revisiones con clientes y el equipo de trabajo de tareas para asegurar la calidad de los productos del trabajo. Identificar las tareas para llevar a cabo las instrucciones de entrega. Documentar las tareas.
PM **TL**	Establecer la duración estimada para realizar cada tarea.
PM **TL**	Identificar y documentar los recursos: humanos, materiales, equipos y herramientas, normas, incluida la formación necesaria del equipo de trabajo para llevar a cabo el proyecto. Incluir en el calendario las fechas en que se necesitan recursos y capacitación.
PM **TL**	Establecer la composición del Equipo de Trabajo asignando roles y responsabilidades de acuerdo con los recursos.
PM **TL**	Asignar fechas de inicio y finalización estimada para cada una de las tareas con el fin de crear la Lista de las tareas del proyecto, teniendo en cuenta la asignación de recursos, la secuencia y la dependencia de las tareas.
PM	Calcular y documentar esfuerzo y costo estimado del proyecto.
PM **TL**	Identificar y documentar los riesgos que pueden afectar al proyecto.
PM **TL**	Documentar la estrategia de control de versiones en el *Plan de Proyecto*.
PM	Generar el *Plan de Proyecto* integrando los elementos previamente identificados y documentados.
PM **TL**	Incluir la descripción del producto, alcance, objetivos y resultados en el *Plan de Proyecto*.
PM **TL**	Verificar y obtener la aprobación del *Plan de Proyecto*. Verificar que todos los elementos del *Plan de Proyecto* son viables y coherentes. Los resultados encontrados se documentan en los *Resultados de la Verificación* y se realizan correcciones hasta que el documento es aprobado por PM.
PM **TL**	Revisar y aceptar el *Plan de Proyecto*. Revisiones y aceptación con el cliente del *Plan de Proyecto*, asegurándose de que los elementos del *Plan de Proyecto* coinciden con la *Declaración de Trabajo*.
PM **TL**	Establecer el *Repositorio del Proyecto* utilizando la estrategia de control de versiones.

PM.2 Plan de Ejecución del Proyecto

Implementar las tareas laborales reales del proyecto de acuerdo con el plan del proyecto. Lo ideal sería que cuando el plan del proyecto se ha acordado y comunicado a todos los miembros de los equipos, el trabajo del desarrollo del producto, que es el objeto del proyecto, debe comenzar.

Rol	Lista de Tareas
PM **TL** **WT**	Supervisar la ejecución del *Plan del Proyecto* y registrar los datos actuales en el *Registro de Estado de Progreso*.
PM **TL**	Analizar y evaluar los costos, plazos e impacto técnico de las *Solicitudes de Cambio*. La *Solicitud de Cambio* puede ser iniciada externamente por el cliente o internamente por el equipo de trabajo. Actualizar el *Plan de Proyecto*, si el cambio aceptado no afecta a los acuerdos con los clientes. Las *Solicitudes de Cambio*, que afectan a esos acuerdos, deben ser negociadas por ambas partes
PM **TL** **WT**	Llevar a cabo reuniones de revisión con el equipo de trabajo, identificar los problemas, revisar el estado de riesgo y realizar un seguimiento de los mismos hasta su cierre.
PM **CUS** **TL** **WT**	Llevar a cabo reuniones de revisión con los clientes, registrar los acuerdos y realizar un seguimiento de los mismos hasta su cierre.
PM	Realizar una *Copia de Seguridad del Repositorio del Proyecto* de acuerdo con la estrategia de control de versiones.
PM	Realizar la recuperación del *Repositorio del Proyecto* utilizando la *Copia de Seguridad del Repositorio del Proyecto*, si es necesario.

PM.3 Evaluación y Control de Proyecto

El propósito es determinar el estado del proyecto y asegurar que el proyecto se lleva a cabo de acuerdo a los planes y tiempos, dentro de los presupuestos proyectados y que satisface los objetivos técnicos. Este proceso incluye la reorientación de las actividades del proyecto para corregir las desviaciones detectadas y variaciones de otra gestión de proyectos o procesos técnicos. La redirección puede incluir re-planificación, según corresponda.

Rol	Lista de Tareas
PM **TL** **WT**	Evaluar el progreso del proyecto en relación con el *Plan de Proyecto*, comparando: - Tareas reales contra tareas previstas - Los resultados reales contra los objetivos establecidos del proyecto - La asignación de recursos real con los recursos previstos - Costo real contra estimaciones presupuestarias - Hora efectiva comparada con la programación prevista - Riesgo real contra los previamente identificados
PM **TL** **WT**	Establecer acciones para corregir las desviaciones o problemas y riesgos identificados en relación con el cumplimiento del plan, documentarlas en el *Registro de Corrección* y realizar el seguimiento de los mismos hasta su cierre.
PM **TL** **WT**	Identificar cambios en los requisitos y/o *Plan de Proyecto* para abordar las principales desviaciones, posibles riesgos o problemas relacionados con el cumplimiento del plan, documentarlos en las *Solicitudes de Cambio* y realizar el seguimiento de los mismos hasta su cierre.

PM.4 Cierre del proyecto

Implica la liberación de los productos finales para el cliente, la entrega de la documentación del proyecto a la empresa, terminación de los contratos de los proveedores, la liberación de los recursos del proyecto y comunicar el cierre del proyecto a todos los interesados. A menudo, un último paso es proceder a la revisión de la implementación del anuncio para identificar el nivel de éxito del proyecto y tener en cuenta las lecciones aprendidas para futuros proyectos.

Rol	Lista de Tareas
PM **CUS**	Formalizar la terminación del proyecto de acuerdo a las instrucciones de entrega establecidas en el *Plan de Proyecto*, proporcionar soporte y obtener el *Registro de Aceptación* firmado.
PM	Actualizar el *Repositorio del Proyecto*.

2.2.2 Proceso de Implementación de Software

El propósito del proceso de implementación de software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas para productos de software nuevos o modificados de acuerdo a los requerimientos especificados. Esta parte de la norma ISO 29110 está destinada a ser utilizada por PyMEs para establecer procesos para implementar cualquier enfoque de desarrollo o metodología que incluye, por ejemplo, desarrollo ágil, evolutivo, incremental, basado en pruebas, etc. basado en la organización de las PyMEs o necesidades del proyecto.
La *Imagen 2.2* muestra el flujo de información entre las actividades del Proceso de

Implementación de Software, incluyendo los elementos de trabajo más relevantes y su relación.

Objetivos del proceso de Implementación de Software

1. Las tareas de las actividades se llevan a cabo a través de la realización del *Plan de Proyecto* actual.
2. Los requerimientos de software son definidos y analizados para la corrección y capacidad de prueba, aprobados por el cliente y son comunicados.
3. Se desarrolla la arquitectura y diseño detallado del software. Se describen los *Componentes de Software* y las interfaces internas y externas de los mismos. La consistencia y trazabilidad de los requisitos de software son establecidos.
4. Se producen los *Componentes de Software* definidos por el diseño. Las unidades de prueba son definidas y realizadas para verificar la consistencia de los requerimientos y el diseño. La trazabilidad de los requerimientos y el diseño se han establecido.
5. El software se produce realizando la integración de los *Componentes de Software* y se verifica usando *Casos y Procedimientos de Prueba*. Los resultados son registrados en el *Informe de Prueba*. Los defectos se corrigen y la consistencia y trazabilidad del diseño del software son establecidos.
6. Una *Configuración de Software*, que cumpla con la *Especificación de los Requerimientos* según lo acordado con el cliente, que incluye la documentación de usuario, operación y mantenimiento, está integrada y almacenada en el repositorio del proyecto. Las necesidades de cambios en la configuración del software se detectan y se inician las *Solicitudes de Cambio* relacionadas.
7. Tareas de verificación y validación de todos los productos de trabajo necesarios se realizan usando los criterios definidos para lograr la consistencia entre los productos de entrada y salida en cada actividad. Los defectos son identificados y corregidos; los registros son almacenados en los *Resultados de Verificación/Validación*.

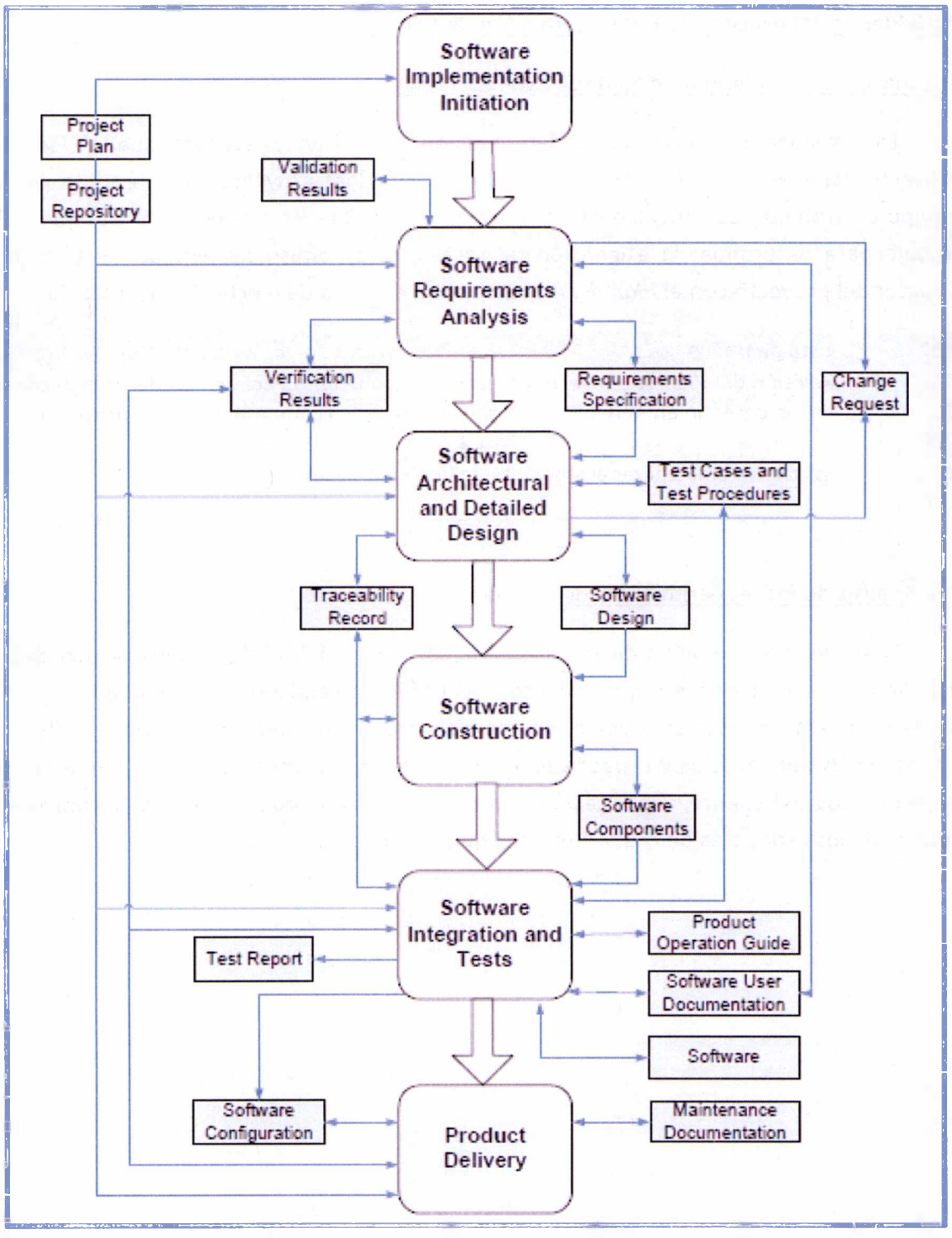

Imagen 2.2 .Diagrama del proceso de Implementación de Software.

Actividades del proceso de Implementación de Software

SI.1 Inicio de la Implementación de Software

La actividad de inicio de implementación de software asegura que el *Plan de Proyecto* establecido en la actividad de planificación del proyecto es entendido por el equipo de trabajo. La actividad ofrece la revisión del plan de proyecto por el equipo de trabajo para determinar la asignación de tareas, compromiso del equipo de trabajo y director del proyecto con el *Plan del Proyecto*, y un entorno de ejecución establecido.

Rol	Lista de tareas
PM **TL** **WT**	Revisión del *Plan de Proyecto* actual con los miembros del equipo de trabajo con el fin de lograr un entendimiento común y obtener su compromiso con el proyecto.
TL **WT**	Establecer o actualizar el ambiente de implementación.

SI.2 Análisis de los requerimientos de software

La actividad de análisis de requisitos de software analiza los requisitos acordados del cliente y establece los requerimientos del proyecto validados. La actividad ofrece la revisión del *Plan de Proyecto* por el equipo de trabajo para determinar la asignación de tareas, elicitación, análisis y especificación de los requerimientos del cliente, acuerdo de los requisitos del cliente, verificación y validación de los requerimientos, y control de versión de los productos de requerimientos de software.

Rol	Lista de tareas
TL **WT**	Asignar tareas a los miembros del equipo de acuerdo con su función, basado en el *Plan de Proyecto* actual.
AN **CUS**	Documentar o actualizar la *Especificación de Requerimientos*. Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) con el fin de obtener nuevos requisitos. Analizar los requerimientos identificados para determinar el alcance y la viabilidad. Generar o actualizar la *Especificación de Requerimientos*.
AN **TL**	Verificar y obtener la aprobación de la *Especificación de Requisitos*. Verificar la exactitud y la capacidad de prueba de la *Especificación de Requerimientos* y la consistencia con la descripción del producto. Además, revisar que los requerimientos estén completos, sin ambigüedades y que no sean contradictorios. Los resultados encontrados se documentan en los *Resultados de la Verificación* y las correcciones son hechas hasta que el documento es aprobado por AN. Si se necesitan cambios significativos, se inicia una *Solicitud de Cambio*.
CUS **AN**	Validar y obtener la aprobación de la *Especificación de Requerimientos*. Validar que la *Especificación de Requerimientos* satisface las expectativas, incluyendo la usabilidad de la interfaz de usuario. Los resultados encontrados son documentados en la *Validación de Resultados* y las correcciones son hechas hasta que el documento es aprobado por CUS.
AN	Documentar la versión preliminar de la *Documentación de Software del Usuario* o actualizar el manual, si es apropiado.
AN **TL**	Verificar y obtener la aprobación de la *Documentación de Software del Usuario*, si es necesario. Verificar la consistencia de la *Documentación de Software del Usuario* con la *Especificación de Requerimientos*. Los resultados encontrados son documentados en los *Resultados de la Verificación* y las correcciones son hechas hasta que el documento es aprobado por AN. Si cambios significantes son necesarios, se inicia una *Solicitud de Cambio*.
TL	Incorporar la *Especificación de Requisitos* y la *Documentación de Software del Usuario* en la configuración de software.

SI.3 Diseño arquitectónico y detallado del software

La actividad de diseño arquitectónico y detallado de software transforma los requisitos del software al sistema de diseño arquitectónico y detallado del software. Esta actividad ofrece, entre otras, las siguientes tareas: revisión del plan del proyecto por el equipo de trabajo para determinar la asignación de tareas; diseñar la arquitectura del software, los componentes del software y las interfaces asociadas; revisión de la especificación de requisitos por el equipo de trabajo; y verificación de casos y procedimientos de prueba para las pruebas de integración.

Rol	Lista de tareas
TL **AN** **DES**	Asignar tareas a los miembros del equipo de trabajo relacionados con su función de acuerdo al *Plan de Proyecto* actual.
AN **DES**	Entender la *Especificación de Requerimientos*.
AN **DES**	Documentar o actualizar el *Diseño de Software*. Analizar la *Especificación de Software* para generar el diseño arquitectónico, su disposición en los subsistemas y *Componentes de Software* que definen las interfaces internas y externas. Describir en detalle, la apariencia y el comportamiento de la interfaz, basada en la *Especificación de Requerimientos* de forma que los recursos para su implementación puedan ser previstos. Proporcionar el detalle de los *Componentes del Software* y sus interfaces para permitir la construcción de una manera evidente. Generar o actualizar el *Registro de Trazabilidad*.
AN **DES**	Verificar y obtener la aprobación del *Diseño de Software*. Verificar la corrección de la documentación del *Diseño del Software*, su viabilidad y consistencia con su *Especificación de Requisitos*. Verificar que el *Registro de Trazabilidad* contiene las relaciones adecuadas entre los requisitos y los elementos de *Diseño del Software*. Los resultados encontrados son documentados en los *Resultados de la Verificación* y las correcciones son hechas hasta que el documento es aprobado por DES. Si son necesarios cambios significativos, se inicia una *Solicitud de Cambio*.
DES	Establecer o actualizar los *Casos y Procedimientos de Prueba* para las pruebas de integración basadas en la *Especificación de Requerimientos* y *Diseño del Software*. El cliente proporcionará datos de prueba, si es necesario.
DES **AN**	Verificar y obtener la aprobación de los *Casos y Procedimientos de Prueba*. Verificar la consistencia entre la *Especificación de Requisitos*, *Diseño de Software* y *Casos y Procedimientos de Prueba*. Los resultados encontrados son documentados en los *Resultados de la Verificación* y las correcciones son hechas hasta que el documento es aprobado por AN.
DES	Actualizar el *Registro de Trazabilidad* incorporando *Casos y Procedimientos de Prueba*.
TL	Incorporar el *Diseño de Software* y el *Registro de Trazabilidad* en la *Configuración del Software*. Incorporar *Casos y Procedimientos de Prueba* en el *Repositorio del Proyecto*.

SI.4 Construcción del software

La actividad de construcción del software desarrolla el código y datos del software a partir del *Diseño del Software*. Esta actividad proporciona una revisión del plan del proyecto por el equipo de trabajo para determinar la asignación de tareas, revisión del diseño del software por el equipo de trabajo para determinar la secuencia de construcción

de software, codificar los componentes de software y aplicar pruebas de unidad, y realizar una trazabilidad entre los *Componentes del Software* y *Diseño del Software*.

Rol	Lista de tareas
TL **PR**	Asignar tareas a los miembros del equipo de trabajo relacionados con su rol, de acuerdo con el *Plan de Proyecto* actual.
PR	Entender el *Diseño del Software*.
PR	Construir o actualizar los *Componentes de Software* basados en las partes detalladas del *Diseño del Software*.
PR	Diseñar o actualizar los casos de prueba unitarios y aplicarlos para verificar que los *Componentes de Software* implementan las partes detalladas en el *Diseño del Software*.
PR	Corregir los errores encontrados hasta que se logre pasar con éxito las pruebas de unidad (alcanzar los criterios de salida).
PR	Actualizar el *Registro de Trazabilidad* incorporando los *Componentes de Software* construidos o modificados.
TL	Incorporar los *Componentes del Software* y el *Registro de Trazabilidad* en la *Configuración del Software*.

SI.5 Integración del Software y Pruebas

Esta actividad asegura que los *Componentes de Software* integrados cumplen los requisitos de software.

Rol	Lista de tareas
TL **PR**	Asignar tareas a los miembros del equipo de trabajo acordes con su rol, de acuerdo con el *Plan de Proyecto* actual.
PR	Comprender los *Casos y Procedimientos de Prueba*. Establecer o actualizar el entorno de prueba.
PR	Se integran los *Componentes de Software* y se actualizan los *Casos y Procedimientos de Prueba* para realizar las pruebas de integración.
PR **CUS**	Realizar pruebas de software utilizando *Casos y Procedimientos de Prueba* de integración y documentar los resultados en el *Informe de Pruebas*.
PR	Corregir los errores encontrados y realizar pruebas de regresión hasta que se alcanza el nivel deseado.
PR	Actualizar el *Registro de Trazabilidad*.
AN	Redactar la *Documentación de Software del Usuario* o actualizar la actual.
AN **CUS**	Verificar y obtener la aprobación de la *Documentación de Software del Usuario*.
TL	Incorporar los Casos y Procedimientos de Prueba, Software, Registro de Trazabilidad, Informe de Prueba, y la Documentación de Software del Usuario en la Configuración de Software como parte de la línea de base.

SI.6 Entrega del producto

La actividad de entrega del producto contiene el producto de software integrado para el cliente. La actividad ofrece: revisión del equipo de trabajo del *Plan de Proyecto* para determinar la asignación de tareas; *Documentación de Mantenimiento* verificada; y la entrega del producto de software y documentación aplicable de acuerdo con las instrucciones de entrega.

Rol	Lista de Tareas
TL **WT**	Asignar tareas a los miembros del equipo de trabajo acordes con su rol, de acuerdo con el *Plan de Proyecto* actual.
DES	Entender la *Configuración de Software*.
DES	Redactar la *Documentación de Mantenimiento* o actualizar la actual.
DES **TL**	Verificar y obtener la aprobación de la *Documentación de Mantenimiento*. Verificar la coherencia de la *Documentación de Mantenimiento* con la *Configuración de Software*. Los resultados encontrados se documentan en los *Resultados de la Verificación* y las correcciones se hacen hasta que el documento es aprobado por el TL.
TL	Incorporar la *Documentación de Mantenimiento* como línea de base para la configuración del software.
TL	Realizar la entrega de acuerdo a las instrucciones de entrega.

2.2.3 Descripción de los roles

Rol	Abreviación	Características
Analista	**AN**	Conocimiento y experiencia para elicitar, especificar y analizar los requerimientos. Conocimiento en diseñar la interfaz de usuario y los criterios ergonómicos. Conocimiento de las técnicas de revisión. Conocimiento de las técnicas de edición. Experiencia en el desarrollo y mantenimiento de software.
Cliente	**CUS**	Conocimiento de los procesos del cliente y capacidad para explicar los requerimientos del cliente. El cliente (representante) debe tener autoridad de aprobar los requerimientos y sus cambios. Conocimiento y experiencia en el dominio de la aplicación.
Diseñador	**DES**	Conocimiento y experiencia en los componentes de software y diseño arquitectónico. Conocimiento en técnicas de revisión. Conocimiento y experiencia en la planificación y ejecución de las pruebas de integración. Conocimiento en técnicas de edición. Conocimiento en el desarrollo y mantenimiento del software.
Programador	**PR**	Conocimiento y/o experiencia en programación, integración y pruebas de unidad. Conocimiento en técnicas de revisión. Conocimiento en técnicas de edición. Experiencia en desarrollo y mantenimiento del software.
Líder de proyecto	**PM**	Capacidad de liderazgo con experiencia en la toma de decisiones, la planificación, la gestión del personal, la delegación y supervisión, las finanzas y el desarrollo de software.
Líder técnico	**TL**	Conocimiento y experiencia en el dominio de procesos de software.
Equipo de trabajo	**WT**	Conocimiento y experiencia de acuerdo a los roles del proyecto: TL, AN, DES, y/o PR. El conocimiento sobre las normas usadas por el cliente o por las PyMEs.

Capítulo 3 – Gamificación

3.1 ¿Qué es Gamificación?

El término Gamificación posee múltiples interpretaciones. Se puede pensar que Gamificación es hacer juegos para anunciar productos o servicios de forma explícita. Por otro lado, también puede definirse como la creación de mundos virtuales en 3D que impulsan el cambio de comportamiento o proporcionan un método para la formación de los usuarios en los sistemas complejos. Sin embargo, la definición que mejor se ajusta al término es la siguiente:

> *El uso de técnicas propias de los juegos y las mecánicas de juego para atraer a los usuarios y resolver problemas.*

Este marco para la comprensión de Gamificación es potente y flexible, ya que puede ser aplicado fácilmente a cualquier problema que pueda ser resuelto impulsando un cambio en la motivación y en el comportamiento humano [3.1].

3.2 Bases de la Gamificación

3.2.1 Todo tiene el potencial de ser un juego

Se puede pensar en una gran cantidad de cosas en la vida que simplemente no son divertidas. Sin embargo, cuatro de los juegos más populares en la última década incluyen actividades tales como cultivos de plantación (Farm Ville [3.2]), atender un restaurante (Diner Dash [3.3]), cambiar el pañal de un bebé (Diaper Dash [3.4]), y arreglar el pelo y uñas de otras personas (Sally's Salon [3.5]).

Otro juego online de alta calificación pone a sus jugadores en uno de los puestos de trabajo más estresantes en nuestra sociedad: control del tráfico aéreo. En el juego Flight Control [3.6], se espera que los jugadores puedan guiar los aviones de forma segura a una pista sin matar a cualquiera de los cientos de pasajeros a bordo.

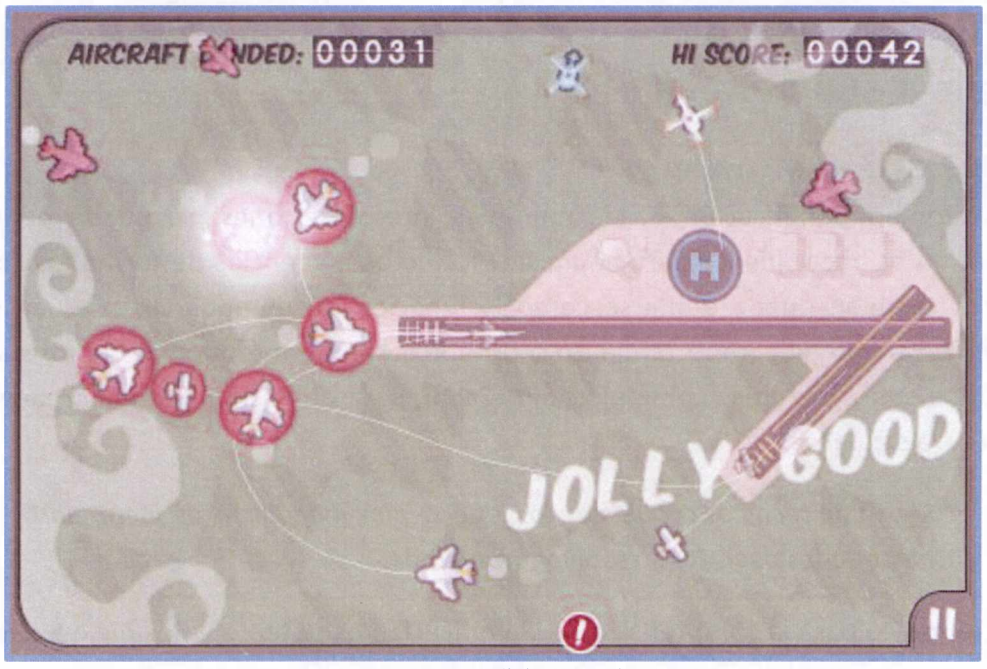

Imagen 3.1. Flight Control

Los ejemplos anteriormente mencionados demuestran que es la mecánica de un juego lo que lo hace divertido y no el tema del mismo.

3.2.2 SAPS

SAPS [3.7] es un acrónimo que significa estado (**S**tatus), acceso (**A**ccess), poder (**P**ower), y regalos (**S**tuff). En pocas palabras, es un sistema de recompensas. La sigla se hizo popular por Gabe Zichermann [3.8], quien explica que la gente valora el estado por sobre los regalos.

Estado

Es la posición relativa de un individuo en relación con los demás, sobre todo en un grupo social. Los beneficios relacionados al estado y las recompensas dan a los jugadores ciertas prioridades en un sistema de clasificación definido. Es importante destacar que este sistema de clasificación no tiene que basarse en el mundo real en absoluto. Algunos ejemplos de elementos de estado incluyen insignias y tablas de clasificación.

- Insignias
  Las insignias son un elemento de estado conocido. Se les puede dar salida virtual o físicamente. Sin embargo, deben ser visibles para otros jugadores en el juego; de lo contrario, su significado y valoración es limitada.

- Niveles y tablas de clasificación
  Los niveles y tablas de clasificación son otra manera de indicar que un jugador tiene más o menos estado o logros de un determinado juego.

Acceso

En lugar de ofrecer a los mejores clientes descuentos o regalos, lo que se brinda es acceso temprano. Otras formas de proporcionar acceso como recompensa a sus jugadores pueden incluir el almuerzo con un CEO, prioridad o asientos VIP, o las primeras citas posibles.

Imagen 3.2. Gilt Noir[3.9] da a sus principales compradores 15 minutos de acceso temprano en ventas online, así como un paquete de bienvenida.

Poder

Adjudicar poder a los jugadores ofrece un mínimo de control sobre otros jugadores en el juego. Por ejemplo, a un buen jugador se le podría dar el rol de moderador en un foro. La mayoría de los foros ofrecen con éxito puestos de poder para que sus jugadores compitan a diario.

Regalos

Si se poseen grandes artículos para regalar y los jugadores están esperando recibir artículos gratuitos, los regalos pueden ser un fuerte incentivo. Una vez que el artículo ha sido regalado, el incentivo para jugar ha terminado. En otras palabras, los regalos sólo son buenos hasta que se redimen, que es la longitud exacta de tiempo que sus jugadores van a participar en el juego.

¿Cómo valoran los jugadores el estado, el acceso, el poder, y los regalos? No saben con exactitud el precio de esos beneficios, por lo que en general, tienden a sobrevalorarlos. Por ejemplo, al evaluar la importancia de no tener que esperar en una fila, la mayoría de las personas sobrevaloran su tiempo ahorrado. Un diseñador de Gamificación debe entender estos valores, y el precio es casi siempre más barato que regalar cosas.

3.3 La motivación del jugador

El jugador es la base de la Gamificación. En cualquier sistema de juego, la motivación del jugador en última instancia conduce el resultado. Por lo tanto, la comprensión de la motivación del jugador es fundamental para la construcción de un sistema de Gamificación exitoso.

3.3.1 ¿Por qué la gente juega?

Una teoría de por qué la gente está motivada a jugar juegos sostiene que hay cuatro razones subyacentes, que pueden ser vistas juntas o por separado como motivadores individuales:

- Para el dominio/maestría/superioridad
- Para liberarse del estrés
- Para divertirse
- Para socializar

Complementariamente, Nicole Lazzaro [3.10] define cuatro tipos de diversión para tratar de entender las motivaciones de los diferentes jugadores [3.11]. Las describiremos a continuación.

<u>Diversión difícil</u>

Para muchos jugadores superar obstáculos es por lo que juegan. Este tipo de diversión crea una emoción mediante la estructuración de la experiencia hacia la persecución de un objetivo. El desafío se centra en la atención y en recompensas sobre el progreso para crear emociones relacionadas al triunfo personal. Inspira la creatividad del desarrollo y aplicación de estrategias. Se premia al jugador con información sobre el progreso y el éxito. Jugadores que disfrutan de la diversión difícil dicen que les gusta:

- Jugar para ver lo buenos que son en realidad.
- Jugar para ganar el juego.
- Tener múltiples objetivos.

- Necesitar una estrategia en lugar de suerte.

Juegos de este tipo ofrecen desafíos cautivantes con una variedad de estrategias. Ellos equilibran la dificultad del juego con la habilidad del jugador.

Diversión fácil

Otros jugadores se centran en el puro placer de experimentar las actividades del juego. La Diversión Fácil da importancia a la atención del jugador en lugar de una condición de victoria. La clave de inmersión despierta en el jugador una sensación de curiosidad. Se incita al jugador a considerar opciones y encontrar más información. La ambigüedad, incompletitud, y los detalles se combinan para crear un mundo vivo. Jugadores de este tipo de diversión afirman que les gusta:

- Explorar nuevos mundos con personas intrigantes.
- La emoción y la aventura.
- Ver lo que sucede en la historia, incluso si tienen que dar un paseo.
- Sensación de cómo ellos y sus personajes son uno.
- Escuchar el sonido de barajar las cartas.

Estímulos ricos y ambigüedad, así como los detalles hacen que el jugador haga una pausa con asombro y curiosidad.

Cambios de estado

Los jugadores argumentan que cómo un juego hace que se sientan en su interior es una de las principales razones por las que juegan. Disfrutan de los cambios en su estado emocional durante y después de jugar. Este aspecto de la experiencia del jugador es la forma en que la percepción, el comportamiento y el pensamiento se combinan en un contexto social para producir emociones y otras sensaciones internas, mayormente las de agitación y alivio. Los jugadores utilizan esta diversión para pasar de un estado mental a otro, o sentir algo diferente. Algunas de razones por las que los jugadores dicen que les gusta jugar son:

- Sentirse mejor.
- Evitar el aburrimiento.
- Ser mejor en algo que importe.

Diversión social

Muchos jugadores sostienen que disfrutan principalmente jugar con otros dentro o fuera del juego. Los participantes juegan juegos para poder pasar tiempo con sus amigos. Los jugadores que utilizan estos juegos los ven como mecanismos para la interacción social. Los jugadores que disfrutan de la interacción con otras personas dicen que:

- La gente es adictiva, no el juego.
- Quieren una excusa para invitar a sus amigos.
- No les gusta jugar, pero es una forma divertida de pasar el tiempo con sus amigos.
- No juegan, pero es divertido ver.

3.3.2 Tipos de jugadores

Cuanto más se sabe acerca de los jugadores, más fácil es diseñar una experiencia para impulsar el comportamiento de los mismos en la forma deseada. El trabajo [3.12] realizado por Richard Bartle [3.13] en la comprensión de los tipos de jugadores es un buen comienzo para entender a los jugadores. Bartle identificó cuatro tipos de jugadores que describiremos a continuación.

Exploradores

A los exploradores les gusta explorar el mundo, no sólo su geografía sino también los detalles más finos de la mecánica del juego. Estos jugadores pueden llegar a conocer cómo funciona el juego y se comportan mejor que los propios creadores del juego. Conocen todas las mecánicas, atajos, trucos, y problemas técnicos que hay que saber en el juego y prosperan en descubrir más.

Asesinos

A los asesinos les gusta provocar y causar drama y/o imponerlas sobre otros jugadores en el ámbito de la aplicación provisto por el mundo virtual.

Triunfadores

Los triunfadores son competitivos y disfrutan superando retos difíciles si son fijados por el juego o por ellos mismos. Cuanto más difícil sea la meta, más recompensados tienden a sentirse.

Socializadores

Los socializadores son a menudo más interesados en tener relaciones con los demás jugadores que jugar el juego en sí. Ellos ayudan a difundir el conocimiento y los sentimientos, y con frecuencia están involucrados en el aspecto de la comunidad del juego.

Un jugador puede tener características de los cuatro tipos al mismo tiempo [3.14]. Para la persona promedio, el desglose podría ser algo como esto:

- 80% socializador
- 50% explorador
- 40% triunfador
- 20% asesino

Si las puntuaciones fueran mutuamente excluyentes y un jugador pudiera sólo ser de un tipo, aproximadamente el 75% de las personas serían probablemente socializadores.

3.4 Mecánicas de juego

Las mecánicas de un juego se componen de una serie de herramientas que, cuando se usan correctamente, prometen dar una respuesta significativa a los jugadores [3.15]. Se describirán algunos de los elementos fundamentales de las mecánicas de los juegos, que guiarán el desarrollo de la componente lúdica propuesta.

3.4.1 Puntos

Los puntos son importantes, independientemente de si su acumulación es compartida entre los jugadores, o incluso entre el diseñador y el jugador. Son un requisito absoluto para todos los sistemas de juego. Es imprescindible valorar y rastrear cada movimiento que los jugadores hacen, incluso si esos resultados no son visibles a ellos. De esta manera, se puede ver cómo los jugadores están interactuando con el sistema y hacer los ajustes adecuados.

Ejemplos de puntos en el mundo real

Puntuación de videojuego

Una puntuación mucho más evidente es la que se presenta en casi cualquier videojuego. La puntuación se encuentra siempre en la esquina de la pantalla, dejando que el jugador conozca lo cerca o lejos que está del siguiente nivel, de otros jugadores, y

finalmente de ganar el juego. Pocos sistemas en la vida real mantienen la puntuación tan omnipresente como los videojuegos.

Puntuación en Redes sociales

Cuando Facebook [3.16] se desarrolló, nada indicaba que el número de amigos de un usuario tenga función alguna. Del mismo modo, el número de seguidores o menciones en Twitter [3.17] nunca fue señalado explícitamente como un "puntaje". Pero ambos lo son. La mayoría de los jugadores pueden nombrar la cantidad de amigos y seguidores que tienen en cualquier red social. Es más, es probable que pueda nombrar que entre sus amigos tiene un número inusualmente alto de amigos o seguidores. Se toma un inventario de todo tipo, simplemente porque Facebook y Twitter le dan un lugar destacado a la "puntuación" en sus páginas.

Sistemas de Puntuación

En la Gamificación, se puede aprovechar uno de los cinco diseños de puntos para formar la base de nuestra experiencia. En algunos casos, el sistema de puntos será abierto, directo y altamente motivador. En otros diseños, se pueden utilizar diferentes tipos de puntos para alcanzar los objetivos. En otros casos, los puntos quedan en un segundo plano frente a otras mecánicas, sirviendo principalmente al diseñador. Los diseños de puntos incluyen los siguientes:

- Puntos de experiencia
- Puntos canjeables
- Puntos de habilidad
- Puntos de karma
- Puntos de reputación

Puntos de experiencia

De los cinco tipos de sistemas de puntos, los más importantes son los puntos de experiencia (XP). Todo lo que un jugador hace dentro del sistema incrementará sus XP y, en general, los XP nunca se pierden y no pueden ser redimidos. Mediante la asignación de XP para cada actividad en el sistema, el diseñador alinea sus objetivos de comportamiento con el jugador de una manera a largo plazo. En algunos sistemas, los XP pueden caducar. La caducidad puede servir al propósito importante de "restablecer" el juego para nivelar el campo de juego. Sin embargo, tal vez aún más importante, los XP no tienen límite. Un jugador continúa ganando XP mientras más juega el juego. Ese es el poder de los XP.

Puntos canjeables

El segundo sistema de puntos se compone de los puntos canjeables (RP). A diferencia de los XP, los RP pueden fluctuar. La expectativa para la mayoría de la gente es que estos puntos son utilizables dentro del sistema en el cambio de las cosas. Ellos son ganados y cobrados. El plazo para este bucle en juegos sociales y los programas de fidelización es "ganar y quemar", lo que indica claramente el propósito de un sistema de RP.

Los RP generalmente forman la base de una economía virtual, y con frecuencia se dan nombres como monedas, dólares, dinero en efectivo, etc. Como cualquier economía, se tendrá que supervisar, administrar y modificar los flujos de capital para asegurarse de que todo funciona a la perfección, así como para evitar la inflación o la deflación masiva.

Puntos de habilidad

El tercer sistema de puntos se llama sistema de puntos de habilidad. Los puntos de habilidad son asignados a actividades específicas dentro del juego y son tangentes a ambos XP y RP. Son un conjunto de puntos de bonificación que le permiten a un jugador ganar experiencia/recompensas por actividades específicas. Mediante la asignación de puntos de habilidad a una actividad, dirigimos al jugador a completar algunas tareas y objetivos claves.

Puntos de karma

Los puntos de karma son un sistema único que rara vez aparece en los juegos clásicos. El único propósito del karma es dar puntos a los jugadores. Es decir, los jugadores no obtienen beneficios al mantener sus puntos de karma, sólo por compartirlos. A menudo, los puntos de karma se dan como parte de una rutina regular, o para comprobar el comportamiento, por ejemplo: se ganan 3 puntos de karma por cada cheque mensual. El objetivo principal de puntos de karma es crear un camino de comportamiento para el altruismo y la recompensa de los usuarios.

Puntos de reputación

Finalmente, los puntos de reputación conforman el sistema de puntos más complejo. Cada vez que un sistema requiere confianza entre dos o más partes que no se puede garantizar ni gestionar de forma explícita, un sistema de reputación es la clave. Su propósito es actuar como un intermediario para la confianza.

Cómo utilizar los sistemas de puntuación

Economías virtuales

El poder de una economía virtual es que permite a un diseñador atraer una gran cantidad de dinero y controlar la forma en que se utiliza. Así es como la mayoría de las economías se han diseñado en el mundo virtual. Por ejemplo, FarmVille cuenta con un sistema de cambio en un solo sentido. Un jugador sólo puede poner dinero en el juego, y como no hay recompensas del mundo real para canjearlo, todo queda en el juego.

Doble moneda

FarmVille implementa lo que se llama una doble moneda. Ha creado dos monedas dentro del sistema del juego: dinero en efectivo y monedas. Cada una se utiliza para diferentes tipos de artículos dentro del juego, como se muestra en la *Imagen 3.3*. Los beneficios de una doble moneda son múltiples. Por ejemplo, una doble moneda puede permitir configurar completamente diferentes valores sobre los artículos dentro de la economía, al mismo tiempo que el control de la oferta monetaria de entrada. Esto significa que se puede variar más fácilmente los valores para las diferentes actividades sin tener que inflar/desinflar todo al mismo tiempo.

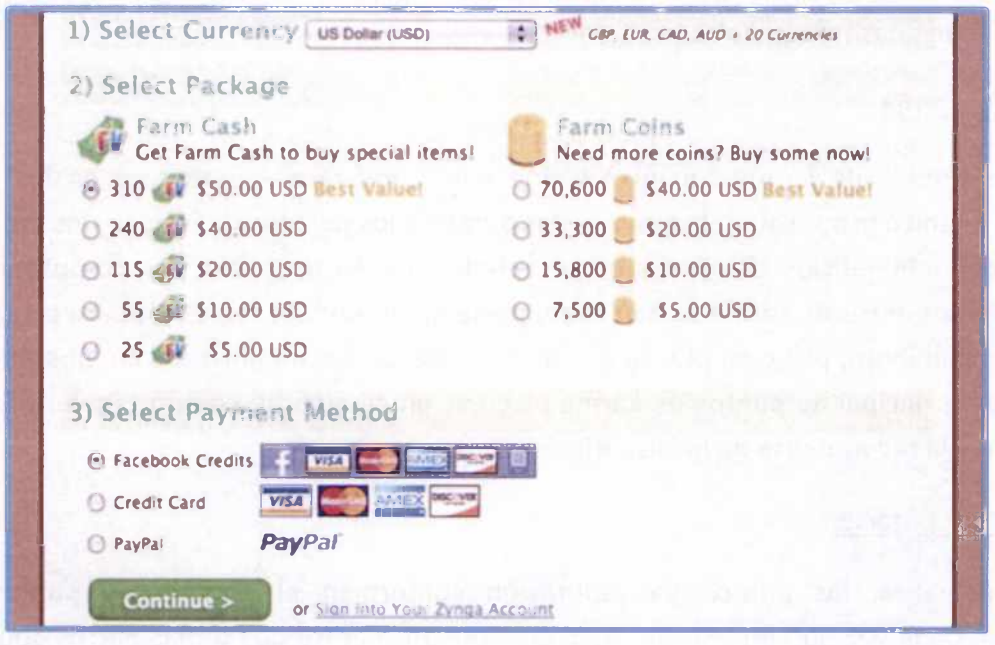

Imagen 3.3. Doble moneda de FarmVille: dinero en efectivo para artículos especiales, monedas de todo lo demás. Se utilizan tipos de conversión complejos para ofuscar la convertibilidad a los usuarios.

3.4.2 Niveles

En la mayoría de los juegos, los niveles indican el progreso, aunque ya no son tan exclusivos en este papel como lo eran antes. Los niveles sirven como un marcador para los jugadores para saber dónde se encuentran en una experiencia de juego a través del tiempo.

En el diseño del juego, el nivel de dificultad no es lineal. En otras palabras, no se necesitan 100 puntos para llegar al nivel uno, 200 para el nivel dos, 300 para el nivel tres, y así sucesivamente. La dificultad aumenta en una forma curvilínea. El juego podría comenzar a ser más complejo, pero como la mayoría de los diseños de niveles, la dificultad aumenta exponencialmente a través de cada nivel y luego disminuye con el tiempo; un ejemplo de esto se muestra en la *Imagen 3.4*.

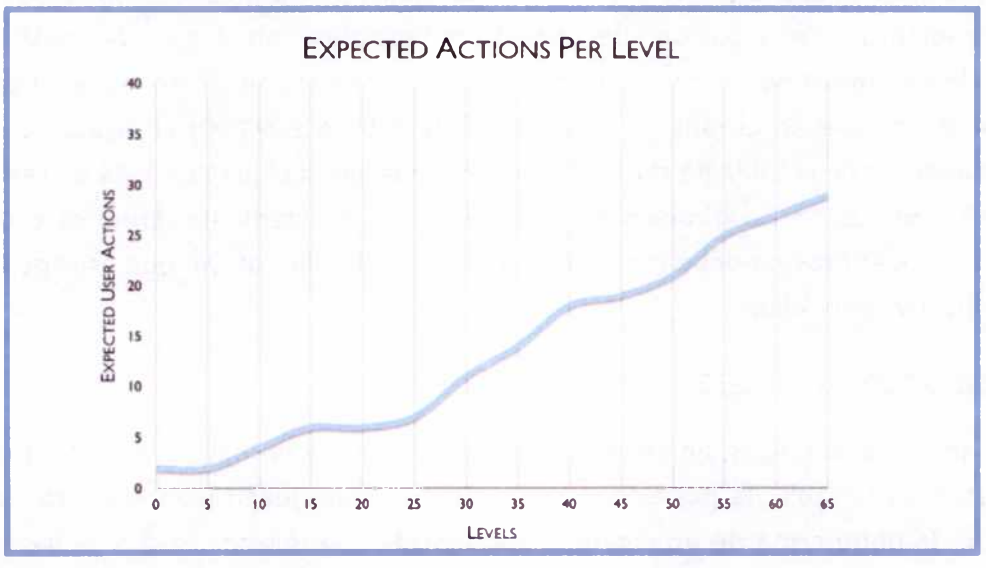

Imagen 3.4. Nivel de complejidad. Aunque esta progresión es diferente en cada juego, el concepto básico muestra que la progresión a través de niveles no es lineal ni exponencial.

<u>El aumento de la dificultad</u>

Actualmente, los sistemas de juegos poseen la característica de ser extensos. Los diseños comienzan a partir de niveles muy simples y se desplazan progresivamente hacia el complejo. En algunos sistemas, los niveles definen la dificultad o el elemento principal del juego. El equilibrio de niveles es tan complejo como la construcción del juego, y debe ser probado y vuelto a probar incluso cuando jugadores están en el juego.

Barra de progreso

Las barras de progreso trabajan en conjunto con los niveles, sirviendo como guía del progreso basado en el porcentaje para un jugador.

3.4.3 Tablas de clasificación

El propósito de una tabla de clasificación es hacer comparaciones simples. De forma predeterminada, se presenta una lista ordenada con una puntuación al lado de cada nombre. Existen dos tipos de tablas de clasificación utilizadas en gran medida en la actualidad.

La tabla de clasificación sin desincentivo

En la era de Facebook y redes sociales, las tablas de clasificación son en su mayoría los instrumentos de creación de incentivos sociales, en lugar de desincentivo. Esto se logra simplemente tomando al jugador y poniéndolo justo en el medio. Sin importar donde cae en el ranking (si es en número 81 o 200.000) el jugador se ve a sí mismo en medio de la tabla de clasificación. Sin embargo, si el jugador está en realidad en el top 10 o top 20, la clasificación debe reflejar esto directamente. En el caso de estos jugadores, la clasificación debe mostrarles su clasificación literal, ya que es probable que sea significativa para ellos.

La tabla de posiciones infinita

En una sala de juegos, no hay demasiadas maneras de permitir a todos los jugadores que existan en la tabla de posiciones de un determinado juego para siempre. En algún momento, la puntuación de un jugador será vencida y la misma caerá o se llegará a un nuevo record y habrá que esperar hasta que alguien finalmente lo supere. Pero existen maneras de controlar las tablas de clasificación de tal manera que ningún jugador desaparezca o se atasque.

Muchos juegos actuales permiten a sus jugadores ver la tabla de clasificación de diversas maneras: a nivel local, social, y en el mundo (*Imagen 3.5*). No hay ninguna razón para que los jugadores no pueden examinar minuciosamente su clasificación como quieran. Las tablas de clasificación también se pueden visualizar con vistas personalizadas para el jugador. Flight Control tiene una tabla de clasificación que muestra otros jugadores al mismo nivel y clasificados en orden de proximidad. Este tipo de clasificación de varias capas ayuda a manejar un juego con millones de jugadores.

Imagen 3.5. El juego Boom Beach[3.18] brinda una tabla de clasificación multidimensional que permite vistas flexibles de progreso y de comparación.

3.4.4 Insignias

Las insignias han estado presentes durante mucho tiempo. La gente desea insignias por todo tipo de razones. Para muchas personas, coleccionarlas es valioso. Otros jugadores disfrutan de las sensaciones de sorpresa o placer cuando una insignia inesperada aparece en un juego. Una insignia bien diseñada también puede ser convincente por razones puramente estéticas. Para los diseñadores de juegos, las insignias son una excelente manera de fomentar la promoción social de sus productos y servicios. Las insignias también marcan el cumplimiento de las metas y el progreso constante en el juego dentro del sistema.

En algunos diseños, las insignias pueden reemplazar los niveles como señales de progreso efectivas. Por ejemplo, Foursquare [3.19] utiliza la acumulación de insignias para demarcar niveles en lugar de tener un sistema de nivelación separado. Como resultado de la naturaleza social, de colección, y visual de insignias, un número creciente de sistemas de juegos están siguiendo el ejemplo de Foursquare.

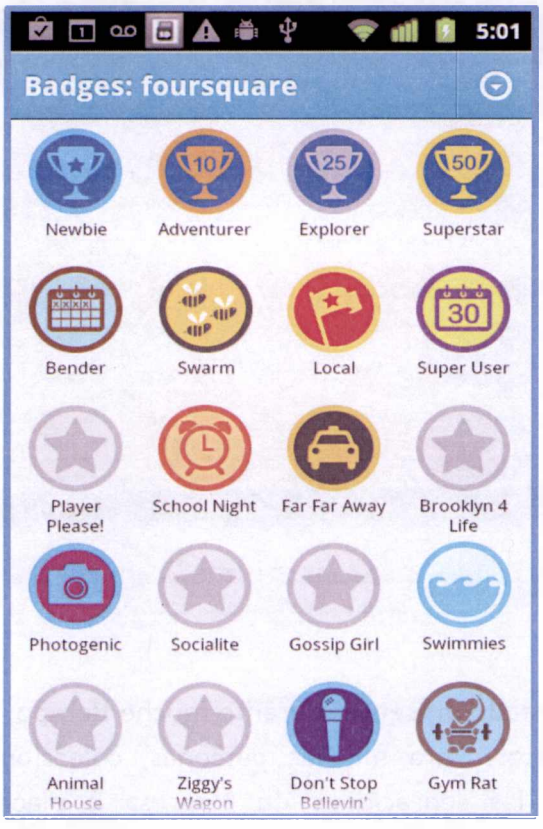

Imagen 3.6. Sala de insignias de Foursquare.

En FarmVille, las cintas sirven como el sistema de insignias (*Imagen 3.7*). Sin embargo, a diferencia de las insignias en Foursquare, estas cintas tienen diferentes niveles para cada objetivo, y actúan en estrecha coordinación con desafíos. Este sistema revela los desafíos más claramente al jugador. De esta manera, el jugador sabe lo que tiene que hacer para avanzar en el juego.

Imagen 3.7. Las cintas de FarmVille son un sistema de insignias que combina desafíos previsibles con logros.

3.4.5 Desafíos y Misiones

Los desafíos y misiones dan a los jugadores indicaciones para saber qué hacer en el mundo del juego. Algunas personas entran en el juego sin tener idea de sus objetivos o unidades fundamentales. El uso de los desafíos y misiones es una opción del sistema que puede añadir profundidad y significado para el jugador. La idea es asegurar que siempre los jugadores tienen un desafío que tomar. Los jugadores deben ser capaces de entrar en la experiencia y siempre tener algo interesante e importante para lograr. Algunos jugadores jugarán desafío tras desafío en secuencia, tratando de vencer al juego tanto como sea posible. Otros simplemente probaran uno según sea necesario para mantener el interés.

También se pueden crear desafíos cooperativos que dependen de una comunidad de jugadores. Estos son el tipo más difícil de diseñar. Al principio del proceso de diseño de un sistema de juego, es mejor diseñar un juego para un solo jugador que puede luego evolucionar al juego cooperativo. Obviamente, diseños cooperativos son socialmente más potentes. También se pueden diseñar experiencias de un solo jugador en un contexto

grupal, donde los jugadores actúan solos, pero sus logros se asocian a un grupo, o los resultados son compartidos con un grupo.

Capítulo 4 – Presentación de Flip-Flop

4.1 ¿Qué es Flip-Flop?

Flip-Flop, de ahora en adelante FF, es una aplicación lúdica desarrollada para el sistema operativo Android [4.1]. Está pensada para ser aplicada en herramientas de gestión, orientada a grupos de desarrollos de software de Pequeñas y Medianas Empresas (PyMEs), que incentive la aplicación de buenas prácticas, propuestas en la norma ISO 29110.

Para el desarrollo de la presente tesina se ha elegido Trello como la herramienta de gestión con la que trabajará FF.

FF entrega premios a los jugadores en base al comportamiento de los mismos, reflejado en Trello. Para lograr esto, FF tiene la capacidad de realizar las siguientes funciones:

1. Conectarse con Trello para obtener toda la información necesaria de jugadores, tareas y actualizaciones en las mismas.
2. Realizar el cálculo y entrega de puntos a los jugadores en base a la información previamente recabada.
3. Brindar una interface donde los jugadores pueden ver sus logros e interactuar con otros jugadores.
4. Registrar todas las tareas realizadas por los jugadores y datos propios del juego con fines estadísticos que sirvan para crear un perfil de jugador. Este perfil podría ser utilizado para ver las fortalezas y/o debilidades de cada jugador.

A continuación se explicará cómo jugar y se describirán las características más importantes del juego. Los detalles de implementación, herramientas y tecnologías utilizadas se verán en detalle en el *Capítulo 5*.

4.2 Antes de jugar

4.2.1 Prácticas seleccionadas

En FF se trabaja con un subconjunto de las actividades de la ISO 29110. Se eligieron las siguientes actividades del proceso de Implementación de Software:

- SI.4 Construcción del software
- SI.5 Integración del Software y Pruebas
- SI.6 Entrega del producto

La elección de estas actividades se debe a que los tesistas se encuentran más familiarizados con las mismas y sus roles; pero no hay ninguna limitación en FF para utilizar todas las prácticas descriptas en la norma.

4.2.2 Roles

Como resultado de la selección de las actividades se obtiene un subconjunto de roles y son, lógicamente, los involucrados en las actividades anteriormente listadas:

- TL: Líder Técnico
- PR: Programador
- AN: Analista
- DES: Diseñador

4.2.3 Herramienta de gestión elegida

Trello es una herramienta web de gestión de proyectos. Fue creada en el año 2011 por Fog Creek Software y en julio de 2014 se dividió formando su propia compañía. Actualmente ofrece un servicio básico en forma gratuita y algunas alternativas pagas. Se basa en el método Kanban [4.2] para gestión de proyectos, con tarjetas que viajan por diferentes listas en función de su estado. Así, se suele tener una lista de cosas por hacer (pendientes), que se están haciendo (en proceso) y hechas (terminadas).

La organización básica de Trello es bastante simple: tableros en los que se pueden crear diferentes listas y, dentro de estas listas, tarjetas. Estas tarjetas van cambiando de lista conforme se van realizando o pasan por diferentes fases. Las tarjetas soportan cualquier tipo de documentación, imágenes, vídeos, listas, comentarios, etc.; por lo que sirven también como instrumento de comunicación en grupos de trabajo, acerca de un determinado tema. Estas tarjetas pueden asignarse a un miembro del grupo, ponerles fecha de entrega, añadirles etiquetas de colores o stickers (para distinguir en un golpe de vista la categoría de la tarjeta, o si ya está aprobada), etc. Pueden moverse de una lista a otra arrastrando las mismas. Cuando se quieren limpiar las tareas realizadas, sólo hay que archivarlas y desaparecen. En cualquier momento se pueden ver todos los elementos archivados, y hacer búsquedas. También aplicar diferentes filtros sobre las tarjetas existentes para encontrar una o un grupo.

Su elección como herramienta de gestión para la presente tesina reside principalmente en una característica: su API [4.3]. Trello proporciona una API web RESTful [4.4], donde cada tipo de recurso (por ejemplo, una tarjeta, un tablero o un miembro) tiene un URI [4.5] que se puede interactuar. No solo esto presenta una gran ventaja, sino que nos permite trabajar con paradigmas y arquitecturas de programación modernas.

Adicionalmente la presencia de una API de este estilo trae asociada una documentación completa, fundamental para el desarrollo de una componente como FF.

Funciones de Trello y FF, su aplicación.

Tablero y listas

Un tablero de Trello es un juego de FF. Dentro del mismo deben existir 3 (tres) listas:

- "POR HACER": en esta lista se deben ubicar las tarjetas que pueden comenzar a realizarse.
- "EN PROGRESO": debe contener las tarjetas en las que los jugadores se encuentran trabajando.
- "TERMINADO": las tarjetas finalizadas se deben ubicar en esta lista.

A modo de referencia, en la *Imagen 4.1* se muestra el tablero utilizado para el desarrollo y pruebas de FF.

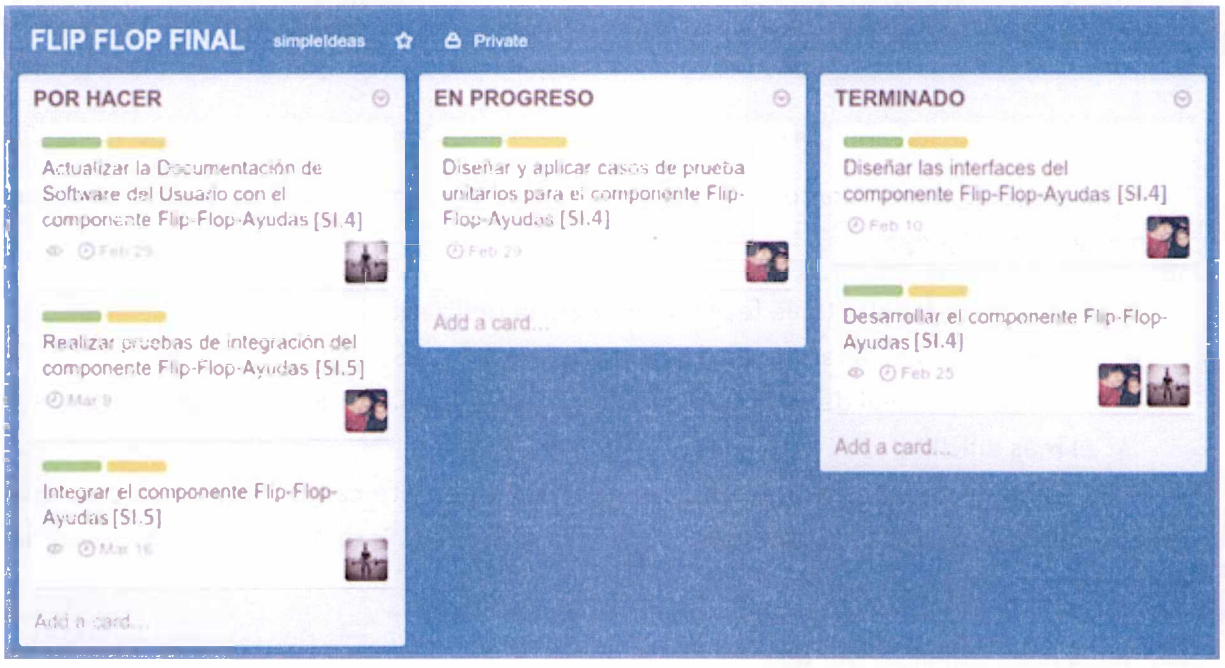

Imagen 4.1. Tablero de Trello

Archivos adjuntos para documentar las tareas

Trello permite adjuntar múltiples archivos a una tarea. Como se ve en la *Imagen 4.2*, brinda la posibilidad de agregar archivos adjuntos desde una gran variedad de fuentes: Google Drive [4.6], DropBox [4.7], Box [4.8], OneDrive [4.9], copiando el enlace o, incluso, permite subir el archivo desde el dispositivo que utilice el usuario.

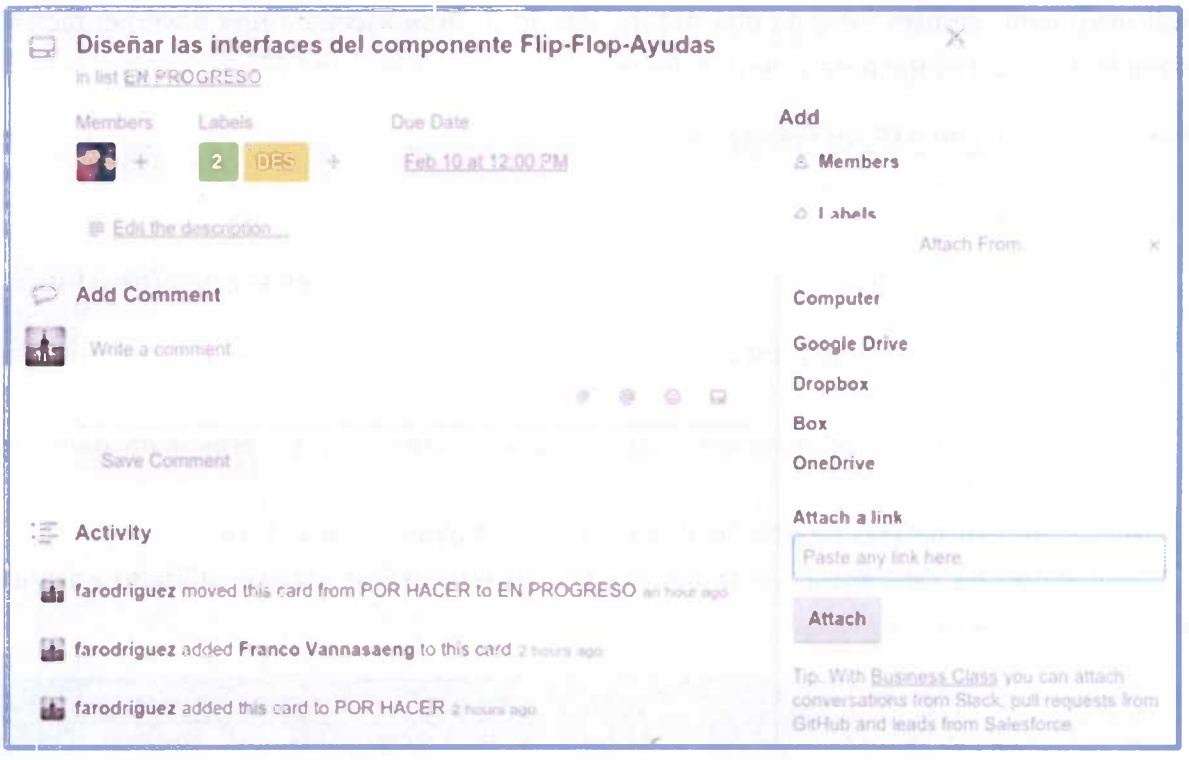

Imagen 4.2. Archivos adjuntos en las tarjetas

Las tarjetas deben contener como mínimo los siguientes datos cargados para que sean válidas en el juego:

- Fecha de vencimiento: la fecha límite para la realización de la tarjeta.
- Rol: el rol al que la tarjeta está orientada (AN, PR, DES, TL).
- Dificultad: el nivel de dificultad de la tarjeta. De 1 a 5 siendo 1 el nivel más fácil y 5 el más difícil.

La fecha de vencimiento es un dato que Trello permite cargar cuando se crea una tarjeta, pero no ocurre lo mismo para el rol y la dificultad. Para estos datos se utilizarán etiquetas.

<u>Etiquetas para definir dificultad y rol de las tareas</u>

Una etiqueta es una forma de categorizar una tarjeta por tipo, representado por un color y el término o, para las etiquetas incoloras, solamente un término. Hay 10 (diez) colores de etiquetas para elegir en Trello, y una opción de etiqueta incolora. Una tarjeta puede tener múltiples etiquetas.

Se define para el correcto uso de FF el siguiente esquema de etiquetas:

- Etiquetas de color verde
  Se presentan 5 (cinco) etiquetas verdes indicando el nivel de dificultad de la tarjeta. Las mismas se nombran de "1" a "5", siendo "1" el nivel más fácil y "5" el más difícil.
- Etiquetas de color amarillo
  Se presentan 4 (cuatro) etiquetas amarillas que indican el rol al que está dirigida la tarjeta. Se nombran "AN", "DES", "PR" y "TL" indicando el rol (ver *Capítulo 2)*.

En la *Imagen 4.3* se presenta la interface brindada por Trello para crear o editar las tarjetas de FF. Allí se incluyen las etiquetas descritas.

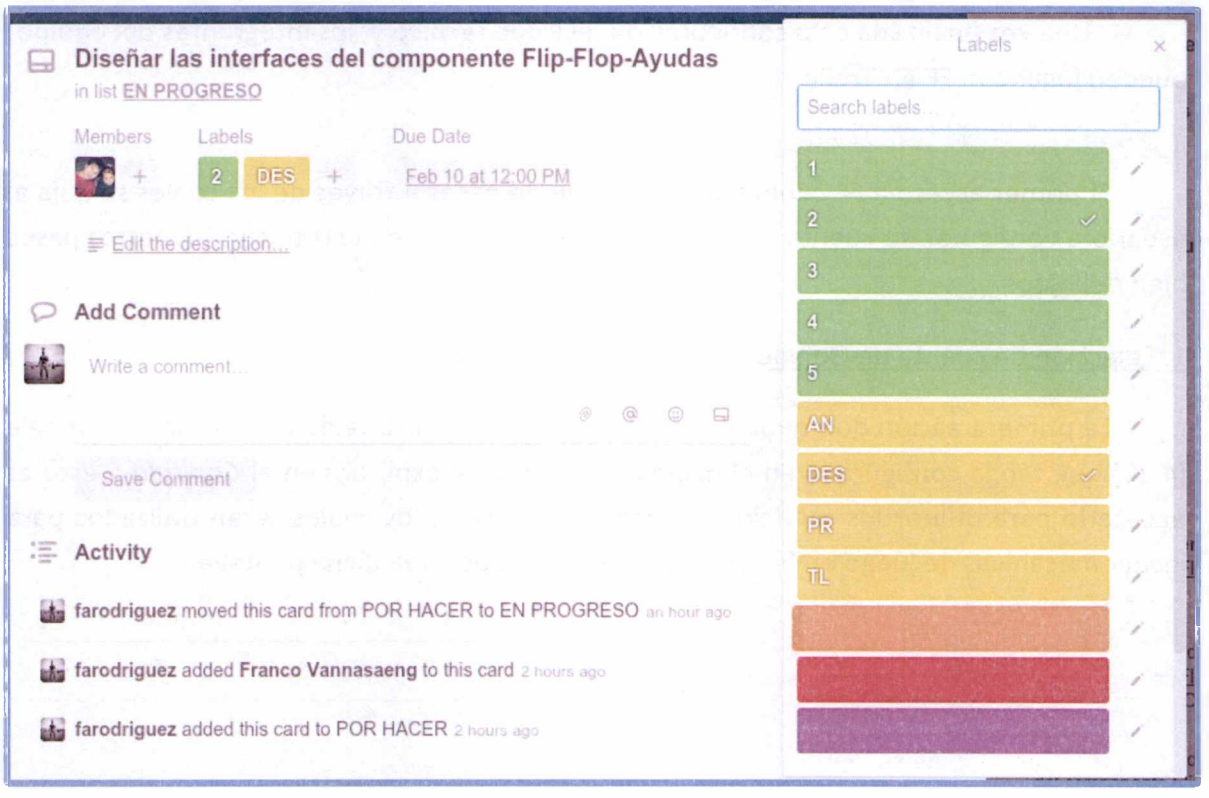

Imagen 4.3. Etiquetas de dificultad y rol

4.4 ¿Cómo jugar?

4.4.1 Configuración inicial

El Líder Técnico es el encargado de configurar inicialmente Trello para trabajar con FF. En primer lugar debe crear un tablero con las listas mencionadas en la sección anterior y las etiquetas correspondientes a los roles y las dificultades de las tareas. Una vez que el tablero ha sido creado se debe continuar con los siguientes pasos:

- Crear tareas correspondientes a las mejores prácticas e ir ubicándolas en la lista "POR HACER". Estas son las tareas que los jugadores podrán asignarse más adelante.
- Agregar a los miembros de su equipo al tablero recientemente creado en Trello.

Una vez finalizada esta configuración, el Líder Técnico y los integrantes del equipo pueden jugar con FF en Trello.

4.4.2 Ingresando por primera vez

El primer acceso a FF cuenta con una serie de pasos a través de los cuales se guía al usuario a configurar su cuenta y perfil en el juego. Se pueden identificar 4 (cuatro) pasos bien definidos:

<u>1. Selección de cuenta de Google</u>

La primera acción que debe hacer el usuario es elegir una de las cuentas de Google [4.10] que tenga configurada en el dispositivo. Como se explicará en el *Capítulo 5* esto es necesario para utilizar los servicios de juegos de Google, los cuales serán utilizados para incluir mecánicas de juego en FF. En la *Imagen 4.4* se observa dicha pantalla.

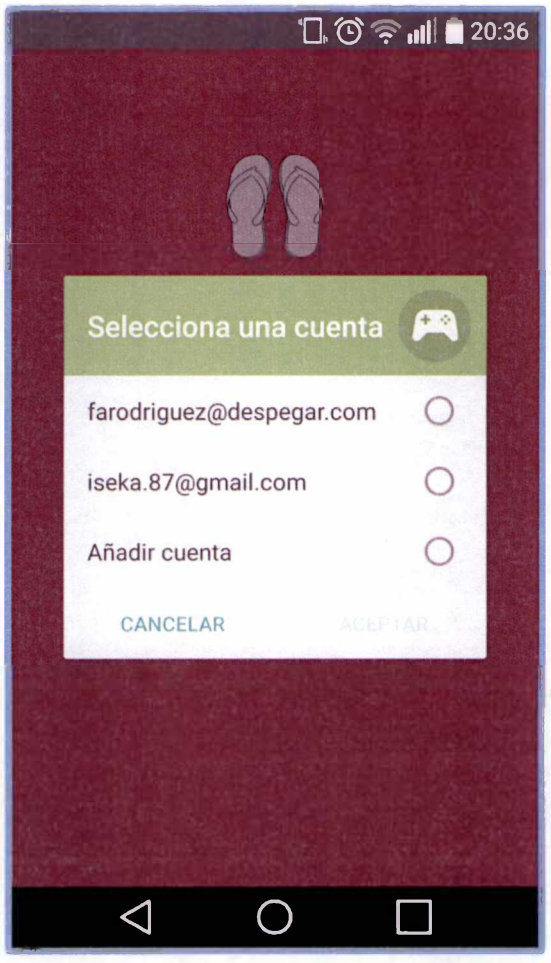

Imagen 4.4. Selección de cuenta de Google.

2. Selección de cuenta de Trello

El segundo paso implica que el usuario se loguee a Trello y acepte los permisos para que FF pueda leer sus datos y actualizar la información de sus tarjetas cuando sea necesario. La *Imagen 4.5* muestra una de las interfaces en la que el usuario debe iniciar sesión en Trello.

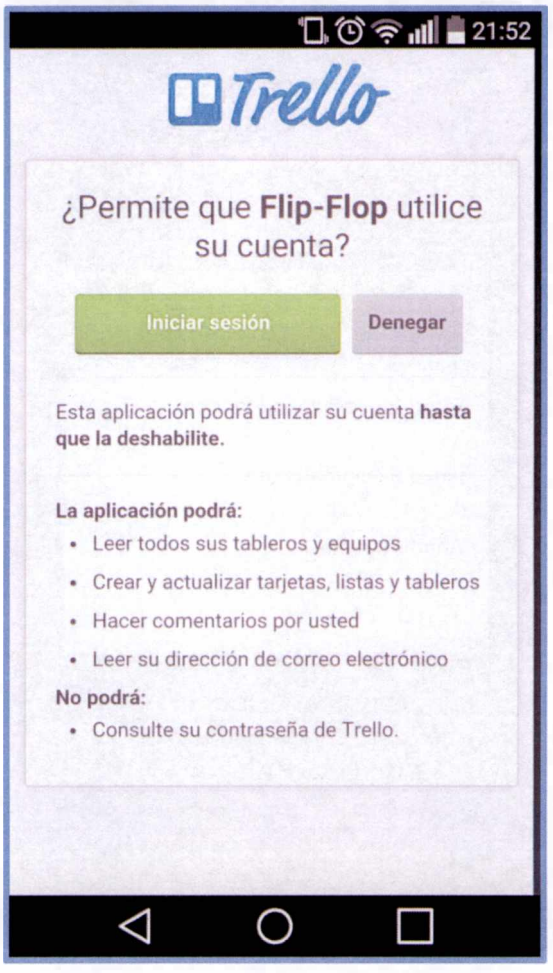

Imagen 4.5. Selección de cuenta de Trello

3. Selección de tablero

Luego de permitir a FF el acceso a su información, se presentan al usuario todos los tableros de Trello a los que pertenece para que elija uno de ellos. El tablero elegido quedará asociado al jugador y será con el que juegue en FF. El mismo debe ser válido para el juego, es decir, debe cumplir con los requisitos que se mencionaron al comienzo de esta sección. El Líder Técnico es responsable de informar el nombre del tablero a los integrantes de su equipo para que los mismos puedan elegirlo en esta pantalla, como se observa en la *Imagen 4.6*.

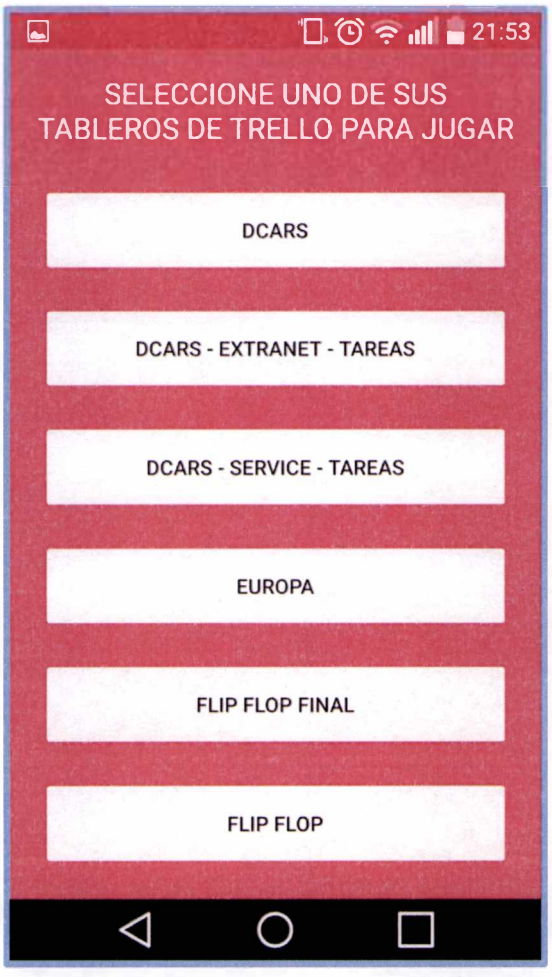

Imagen 4.6. Selección de tablero de Trello.

4. Selección de rol

El último paso es el de selección de rol. Al igual que en el paso anterior, se le presentan al usuario los roles para que seleccione con el que va a jugar. El rol seleccionado debe representar su rol real en el tablero que eligió anteriormente. En la *Imagen 4.7* se pueden observar los roles incluidos en FF.

Imagen 4.7. Selección de rol.

Nota: esta secuencia de pasos solo se realiza una única vez. Para el desarrollo de la presente tesina no se tiene en cuenta que los jugadores puedan jugar muchos juegos al mismo tiempo. Por cuestiones de tiempo de desarrollo un jugador puede jugar solo en un juego a la vez.

4.4.3 Perfil

Luego de llevar a cabo la configuración inicial y en los siguientes accesos al juego, el perfil del jugador podrá verse en la pantalla. Aquí se muestra un resumen de la actividad del jugador, su estado actual y es el punto de partida de todas las acciones que el jugador puede realizar en FF. En la *Imagen 4.8* podemos ver la información presentada. Se muestra el puntaje total alcanzado hasta el momento en el juego junto a información obtenida desde Trello: foto, nombre, correo electrónico y la cantidad de tarjetas que posee en cada una de las listas. También 3 secciones llamadas "ESTADÍSTICAS", "MIS TAREAS" y "SOLICITUDES DE AYUDA". Las mismas se describirán más adelante en este mismo capítulo.

Imagen 4.8. Perfil del jugador en el primer ingreso

4.4.4 Desarrollo del juego

El Líder Técnico del equipo de desarrollo es el encargado de suministrar tarjetas a Trello. Crea tarjetas y las coloca en la lista "POR HACER", dejándolas visibles a los jugadores. Cada una de estas tarjetas está relacionada a una buena práctica de la ISO 29110 y debe ser válida para el juego.

Por otro lado, los jugadores se asignarán las tareas que deseen realizar. No hay restricciones en la cantidad de tareas ni en el tipo de las mismas que puede elegir un jugador. De esta forma cada jugador tiene la posibilidad de trabajar en *n* tareas y debe actualizar el estado de las mismas a medida que las vaya resolviendo. Es un requisito que cuando se finalice una tarjeta la misma quede situada en la lista "TERMINADO".

Todos los viernes a última hora se realiza la entrega de puntos. Se analizará el trabajo de cada jugador y se entregan los puntos, medallas y/o premios correspondientes. Para esto se tienen en cuenta las tarjetas que se encuentren en la lista "TERMINADO" al momento del cálculo; por esta razón es fundamental mantener Trello actualizado. De esta forma queda finalizada la competencia semanal correspondiente, de donde se obtendrán los respectivos ganadores y perdedores. Se denomina a cada competencia semanal como una partida de FF.

Hasta este momento se ha descrito el flujo básico del juego sin considerar las funcionalidades que hacen de FF una herramienta divertida para los jugadores. A continuación se explicarán las funcionalidades desarrolladas para incluir dinámicas y mecánicas de juego en FF. También se explicará en detalle el cálculo de puntos.

4.5 Gamificación en Flip-Flop

4.5.1 Solicitudes de ayuda

Durante el desarrollo de una tarea, muchas veces se necesita ayuda de otro integrante del equipo de desarrollo para resolver alguna situación particular. Para poder representar esta situación en FF, y también para tener un seguimiento de las acciones de los jugadores, se brinda la funcionalidad de "Solicitar ayuda". Cuando un jugador necesita ayuda para la resolución de una tarea puede mandar una solicitud al equipo.

¿Cómo se crea una solicitud de ayuda? En la sección "MIS TAREAS" se listan las tareas que el jugador posee en la lista "EN PROGRESO" de Trello. Desde cada una de ellas se puede crear una solicitud desde la opción "Solicitar ayuda" ubicada en la parte inferior derecha de la tarea *(Imagen 4.9)*. Al resto de los jugadores del equipo se les enviará una notificación informando acerca de esta solicitud.

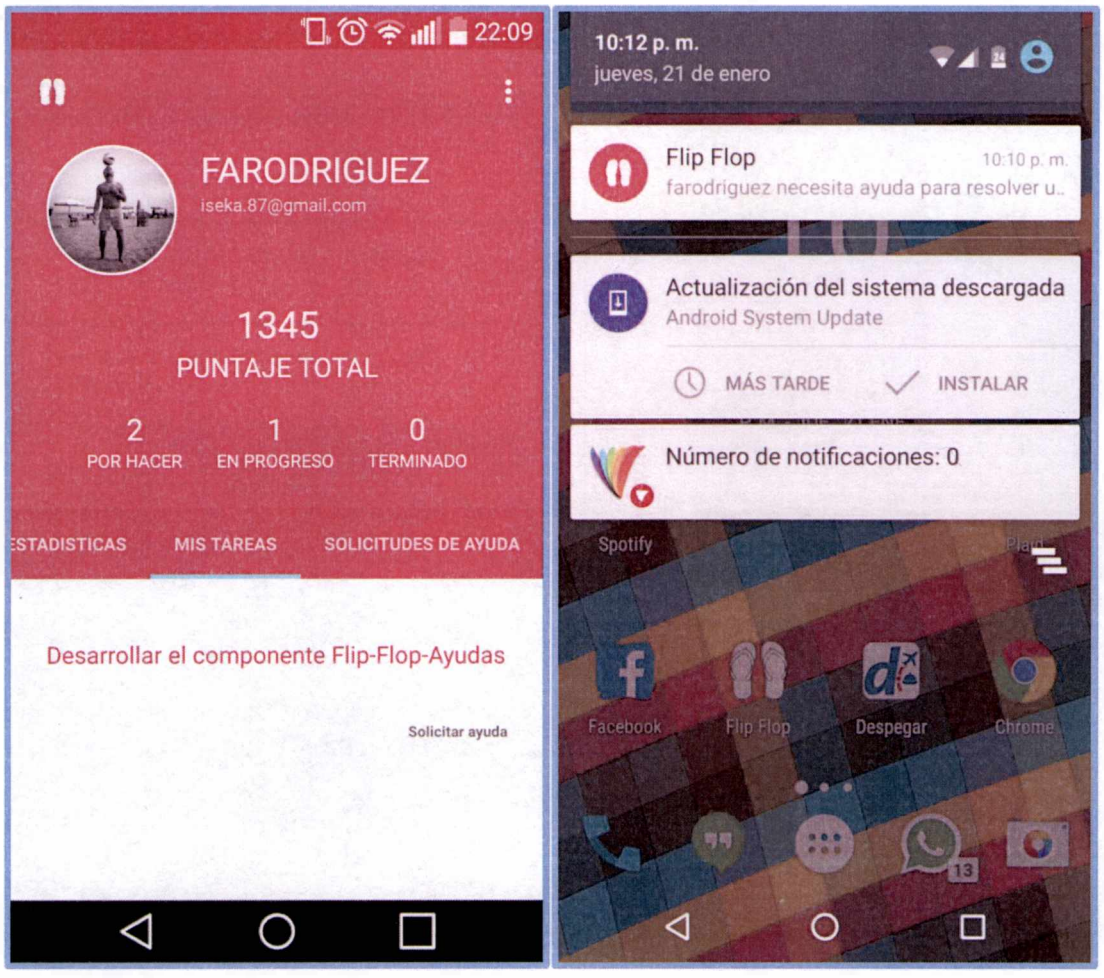

Imagen 4.9. Creación de una solicitud de ayuda (izq.); y notificación luego de haber creado un pedido (der.)

¿Cómo se responde a una solicitud de ayuda? En la sección "SOLICITUDES DE AYUDA" se listan todas las solicitudes de ayuda actuales del equipo de desarrollo. Cada solicitud contiene la tarea relacionada, el jugador que creó la solicitud y el tiempo transcurrido desde la creación de la misma. En la parte inferior derecha se presenta la opción "Ayudar" desde donde los jugadores pueden responder a una solicitud particular *(Imagen 4.10)*. Cuando un jugador responde a una solicitud el resultado es el siguiente:

- Se envía una notificación al emisor de la solicitud informando el jugador que ha decidido ayudarlo (ayudante) para que puedan avanzar juntos en la resolución de la tarea lo antes posible.
- En la tarjeta de Trello correspondiente se agrega al ayudante como un miembro más. En consecuencia el ayudante tendrá una tarjeta más en la sección "MIS TAREAS" pero con la etiqueta "AYUDANDO". De esta forma cada jugador puede ver fácilmente cuáles son sus tareas y en cuales está como ayudante.

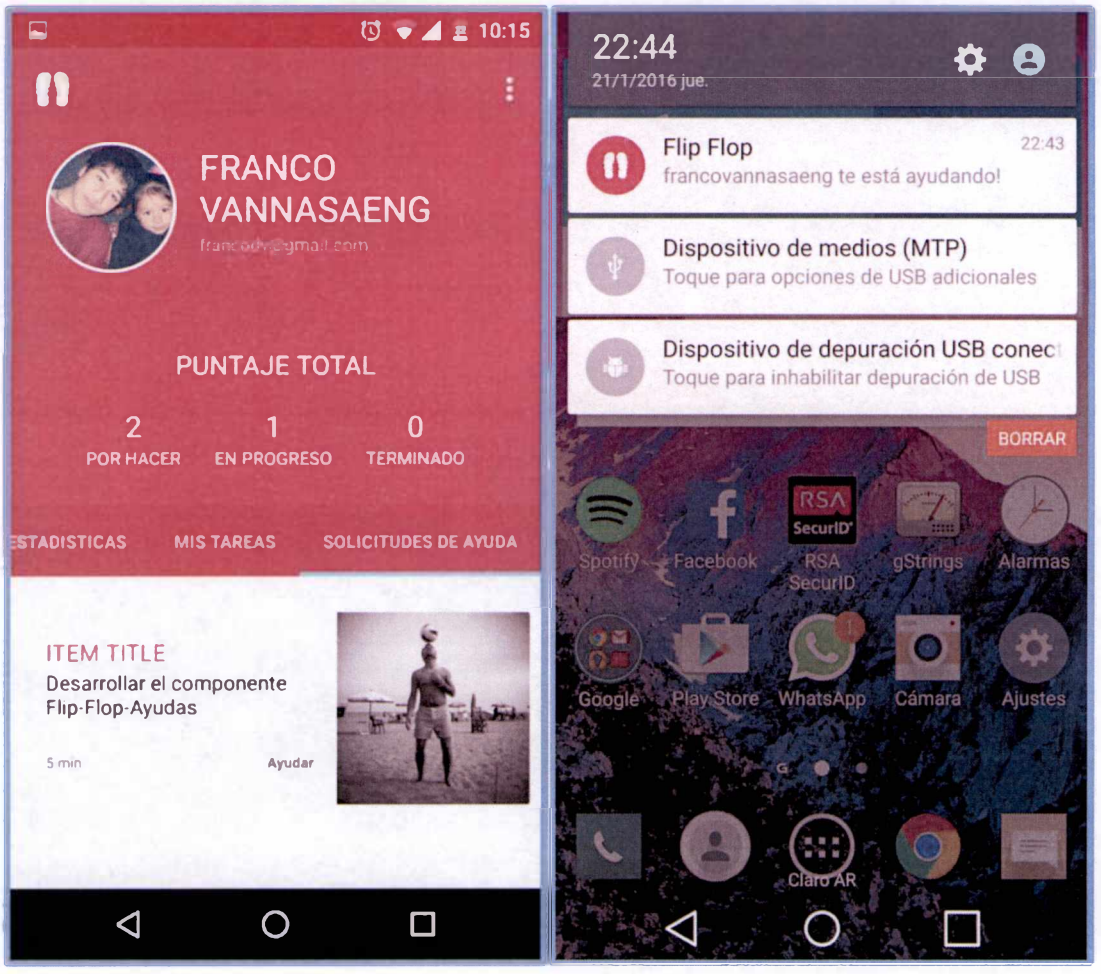

Imagen 4.10. Respuesta a una solicitud de ayuda (izq.); y notificación luego de ayudar (der.)

Para incentivar el uso de la funcionalidad se tendrán en cuenta estas ayudas en el cálculo de puntos. El ayudante ganará un porcentaje de los puntos obtenidos por la realización de dicha tarea.

Los jugadores no pueden hacer uso de esta función ilimitadamente. Los mismos comienzan el juego con 3 pedidos de ayuda a su disposición, pero una vez usados se debe cumplir una condición que está directamente asociada a la actividad del jugador para poder seguir usando la funcionalidad. La condición hace referencia a la relación entre la cantidad de veces que en su equipo se enviaron solicitudes de ayuda y la cantidad de veces que el jugador ha sido ayudante en dichas solicitudes.

Se define de la siguiente manera;

siendo:

n el número de solicitudes de ayuda que se enviaron en un equipo;
j el número de jugadores del equipo; y
a el número de veces que el jugador brindó ayuda;

el jugador puede solicitar ayuda si y sólo si **a >= n / (j *2)**

Para entenderlo mejor se explica con un ejemplo. En un equipo de 4 jugadores, para que un jugador pueda pedir ayuda (exceptuando las 3 veces que se dan al inicio del juego) debe haber sido ayudante en 1 de cada 8 pedidos de ayuda. Es decir, si hubo 17 pedidos de ayuda un jugador puede usar la funcionalidad si brindó ayuda en al menos 2 oportunidades. De lo contrario no puede hacer uso de la misma. En la siguiente funcionalidad se explicará otra forma de obtener pedidos de ayuda.

En el *Capítulo 3* se describieron las mecánicas de juego que se iban a tener en cuenta en el desarrollo de FF. Esta funcionalidad fue pensada y diseñada como un desafío cooperativo y/o misión que depende del jugador que pidió ayuda y el ayudante; y en donde el resultado (en este caso los puntos obtenidos por la realización de la tarea) es , en cierta forma, compartido.

4.5.2 Documentación de la tarea

Una cuestión no menos importante en los equipos de desarrollo es la falta de documentación. Por diversas cuestiones relacionadas con tiempo, falta de costumbre o simplemente no saber los beneficios de tener una documentación actualizada y completa; la misma queda, muchas veces, relegada.

Por este motivo en FF se implementó un sistema de recompensas basado en la documentación de las tareas. Un jugador puede documentar la tarea que desee a través de Trello. Una tarea puede tener múltiples archivos adjuntos, pero esto no quiere decir que sean archivos que el jugador adjuntó como documentación. En este listado pueden figurar archivos necesarios para la resolución de la tarea como imágenes, documentos compartidos, o cualquier material que le sea útil al jugador responsable de dicha tarea. Para poder distinguir estos archivos del archivo de documentación que un jugador puede cargar es necesario respetar un prefijo en el nombre del mismo. De esta forma, cuando un jugador desee adjuntar documentación el nombre del archivo debe contener el prefijo "DOC:". FF se encargará de identificarlos y premiar a los jugadores.

Como se mencionó anteriormente esta funcionalidad puede verse como un sistema de recompensas. Específicamente, se diseñó como un sistema de puntos de habilidad (ver *Capítulo 3*) en donde el jugador consigue recompensas por realizar actividades específicas dentro del juego. A continuación detallamos las recompensas que se obtiene por documentar una tarea:

- En primer lugar; el jugador ganará un puntaje extra. Se define como el 20% más de los puntos que esa tarea tenga asociados.
- En segundo lugar; se establece una relación con la funcionalidad de "SOLICITUDES DE AYUDA". Cada 5 (cinco) tareas documentadas el jugador ganará un pedido de ayuda.

4.5.3 Estadísticas

Entre las capacidades presentes en FF se destacó la de registrar todas las acciones de los jugadores para poder analizarlas y encontrar posibilidades de mejoras e incentivos para los mismos. Una vez diseñadas, se pusieron a prueba las funcionalidades antes descritas, de las que se recabaron datos con los que se comenzó a definir la forma de procesar y mostrar dicha información. Aquí surgió la idea de crear y calcular métricas acerca del comportamiento de los jugadores y luego presentarlas como estadísticas. Esto influye en la competitividad del juego y, por ende, es una fuente de motivación para algunos jugadores. También brinda una herramienta para que ellos mismos puedan saber qué características deben mejorar y en cuales se desempeñan satisfactoriamente.

Por otro lado, y con el objetivo de valorar el desempeño de los jugadores, estas métricas son de gran utilidad para el Líder Técnico. A través del análisis de las mismas podría tomar acciones para corregir y/o mejorar el desempeño de su equipo de trabajo. Incluso se podrían tener en cuenta las mismas a la hora de realizar cambios de categorías, cambios de roles o revisiones salariales.

Se describen a continuación las métricas creadas en FF *(Imagen 4.11)*.

Métricas

Documentador

Esta métrica es bastante simple y se relaciona directamente con la funcionalidad de documentación de una tarea. Se define la misma como el porcentaje de tareas documentadas de un jugador.

Versatilidad

Se mencionó que las tarjetas de Trello debían contener como mínimo una serie de datos para que sean válidas en el juego. Uno de los datos es el rol al que está dirigida la tarea. También se describió que los jugadores se asignan tareas libremente. De esta forma, un jugador puede asignarse tareas de otros roles si asi lo desea. Para detectar y analizar este comportamiento se crea la métrica llamada versatilidad y se define como el porcentaje de tareas de otros roles que un jugador ha realizado.

Complementariamente, esta métrica influye en el puntaje de la tarea. Al igual que la documentación, la versatilidad otorga un puntaje extra. De esta manera, las tareas resueltas que sean de un rol distinto al del jugador responsable de la misma brindan un 25% más de los puntos asociados a la misma.

Esta métrica puede ser de gran utilidad para que el Líder Técnico detecte a los jugadores que realizan tareas de diferentes roles y pueda así llevar a cabo o proponer, por ejemplo, un cambio de rol para el jugador. También es un medio para que los jugadores demuestren su capacidad de realizar tareas que requieren otras aptitudes.

Puntualidad

Otro de los datos requerido en las tareas de Trello es la fecha de vencimiento de la tarea. Esta fecha, si bien puede no ser estricta, refleja una estimación que previamente el Líder Técnico debe haber realizado como parte de la actividad de Planificación del Proyecto (ver *Capítulo 2*).

Para reconocer el esfuerzo y responsabilidad de aquellos jugadores que realizan las tareas dentro del tiempo estimado, se presenta esta métrica que indica el porcentaje de tareas que un jugador finalizó antes de la fecha de vencimiento de la misma. Con el mismo objetivo se penalizará a aquellos jugadores que finalicen la tarea fuera de fecha. Para estos casos se restará el 30% de los puntos asociados a la tarea.

Independencia

Se define como la independencia de un jugador el porcentaje de tareas que ha realizado sin ayuda. Estos valores se calculan con la información que FF almacena sobre el uso de la funcionalidad "SOLICITUDES DE AYUDA". Por ejemplo, un jugador que nunca ha solicitado ayuda para resolver sus tareas posee 100% de independencia.

Cooperación

La Cooperación, al igual que la Independencia, se calcula a partir del uso de "SOLICITUDES DE AYUDA". Como se explicó en dicha funcionalidad los jugadores pueden enviar solicitudes de ayuda que luego algún jugador del equipo puede responder y comenzar a ayudar. En este contexto, la Cooperación indica en qué medida un jugador ayuda a sus compañeros. Esto se calcula como el porcentaje de tareas en las que un jugador figura como ayudante, del total de solicitudes de ayuda que hubo en el equipo al que pertenece.

Esta métrica promueve el uso de dicha funcionalidad y pretende impulsar un aumento del compañerismo dentro del equipo.

Superación

Se definió Puntualidad en relación a la fecha de vencimiento de la tarea y versatilidad en relación al rol al que está dirigida la misma. El tercer dato obligatorio para la validez de una tarea es el nivel de dificultad de la misma. Este nivel va de 1 a 5 e influye, como se explicará más adelante, en los puntos de la tarea. Por el momento, conocer la existencia del nivel es suficiente para comprender la presente métrica.

Se define la Superación como el promedio de dificultad de las tareas que un jugador resuelve. Para presentar la métrica como porcentaje, se transforma el promedio a un porcentaje y se lo muestra como las otras métricas.

Este simple cálculo dice mucho acerca de un jugador. Aquel jugador que tenga un nivel de superación alto es debido a que realiza tareas dificultosas. Esta métrica utilizada en conjunto con la puntualidad podría servir para detectar futuros cambios de seniority de un desarrollador, por poner solo un ejemplo.

Se han presentado hasta aquí 6 (seis) métricas que describen las estadísticas de los jugadores y, que para llevar a cabo el cálculo de las mismas, utilizan la información almacenada en FF.

Imagen 4.11. Estadísticas

4.5.4 Sistema de puntos

Como se explicó en el *Capítulo 3,* los puntos son un requisito absoluto en un sistema de juego y existen diferentes sistemas de puntuación. En FF los puntos son un elemento fundamental. Como se describió anteriormente, el objetivo de los jugadores es realizar tareas y estas tareas les entregarán puntos. A esto se agregan todas las funciones diseñadas que pueden utilizarse para obtener aún más puntos. Se explicará cómo se calculan los mismos, incluyendo todas las variables que intervienen en dicho cálculo.

Al finalizar cada partida se suman los puntos de todas las tarjetas finalizadas de cada jugador y se incrementa el puntaje total del mismo con el resultado de esta sumatoria. Se define para cada tarea un puntaje básico que hace referencia a los puntos que esa tarea tiene asociados sin tener en cuenta otras variables. El puntaje básico, entonces, está definido por el nivel de dificultad de la siguiente manera:

Nivel de dificultad	Puntaje básico
1	10
2	20
3	30
4	40
5	50

Luego de obtener el puntaje básico, se aplican las siguientes reglas que pueden modificar dicho puntaje:

- Roles involucrados
  Si la tarea es de un rol diferente al rol del jugador se adiciona un **25%** del puntaje básico.

- Fecha de finalización
  Si la tarea se finaliza antes de la fecha estimada se adiciona un **20%** del puntaje básico. En caso contrario, en el que la tarea se finaliza vencida, se penaliza al jugador restando un **40%** del puntaje básico.

- Documentación de la tarea
  Si la tarea presenta documentación se adiciona un **20%** del puntaje básico.

- Solicitudes de ayuda
  Los jugadores que figuran como ayudantes recibirán un **30%** del puntaje básico de la tarea en la que ayudaron.

Se explica con un ejemplo. Un jugador de rol analista finaliza una tarea de rol programador fuera de tiempo con documentación y habiendo sido ayudado. La tarea tiene nivel de dificultad 4. Se obtienen los siguientes valores:

- Puntaje básico: +40
- Versatilidad: +10
- Puntualidad: -16
- Documentación: +8
- Solicitudes de ayuda: +12 (para el ayudante)
- **Total: 42 puntos**

Se puede observar cómo algunas funcionalidades tienen más peso que otras a la hora del cálculo del puntaje.

Por último, se debe mencionar que esta modalidad de cálculo de puntaje es conocida como Puntos de experiencia. Todas las acciones que los jugadores hacen en el juego tienden a incrementar sus puntos de experiencia. Por otro lado, los mismos no tienen límite ni caducan, es decir, seguirán aumentando a medida que el jugador juegue.

4.5.5 Insignias/Logros

Los Logros pueden ser una gran manera de aumentar la participación de usuarios dentro del juego. Se pueden implementar Logros en el juego para animar a los jugadores a experimentar con características que normalmente no utilizarían. También pueden ser una forma divertida para que los jugadores comparen el progreso con otros y participen en una sana y alegre competencia.

Luego de realizar el cálculo de puntos semanalmente se procede al desbloqueo de Logros. Con esto se hace referencia a que en base a los datos de los jugadores, que luego de cada partida se actualizan, los mismos pueden obtener Logros. Para el diseño e implementación de esta funcionalidad, a diferencia de las descritas hasta el momento, se utilizaron los servicios de Google Play Game. A través de los mismos, se pueden integrar mecánicas de juegos a FF (ver *Capítulo 5*).

A continuación se detallan todos los logros presentes en FF.

Nombre	Descripción
GANADOR	Gana 5 partidas.
EXPERTO	Gana 10 partidas.
REY	Gana 20 partidas.
DOCUMENTADOR	Documenta 5 tareas.
BIBLIOTECARIO	Documenta 10 tareas.
PUNTUAL	Finaliza 5 tareas a tiempo.
PRECISO	Finaliza 10 tareas a tiempo.
INDEPENDIENTE	Finaliza 5 tareas sin ayuda.
AUTODIDACTA	Finaliza 10 tareas sin ayuda.
VERSATIL	Finaliza 5 tareas de otros roles.
MAGO	Finaliza 10 tareas de otros roles.
COOPERADOR	Brinda ayuda en 5 tareas.
SOCORRISTA	Brinda ayuda en 10 tareas.
MAESTRO	Obtiene 500 puntos de experiencia
PROFESOR	Obtiene 1000 puntos de experiencia
GENIO	Obtiene 2000 puntos de experiencia

Adicionalmente, para aumentar la participación de los jugadores, se entregan Logros cuando por primera vez se realiza una acción específica en el juego.

Nombre	Descripción
PRIMERA TAREA DOCUMENTADA	Documenta 1 tarea
AYUDANTE	Ayuda en 1 tarea.
PRODUCTIVO	Finaliza 1 tarea.

Por último, y no menos importante, se obtendrá el respectivo Ganador y Perdedor de cada partida. Para estos dos casos se enviará una notificación personalizada al ganador y perdedor respectivamente.

Los jugadores podrán consultar sus logros a través de la opción "LOGROS" del menú, ubicado en la esquina superior derecha de la pantalla de FF. Allí se listan todos los Logros, los ya desbloqueados y los que aún no se han obtenido *(Imagen 4.12)*.

Imagen 4.12. Logros

4.5.6 Tabla de clasificación

Como se mencionó en el *Capítulo 3,* el estado es la posición relativa de un individuo en relación con los demás. Las tablas de clasificación son una manera de indicar el estado. Al igual que para la implementación de Logros, se utilizan los servicios de Google Play Game para incluir una tabla de clasificación en el juego.

Desde el mismo menú desde donde se accede a la sección "LOGROS" es posible ingresar a "TABLA DE CLASIFICACIÓN". Allí se presenta la Tabla de Clasificación ordenada por los puntos de experiencia de los jugadores *(Imagen 4.13)*. Esta tabla es actualizada al finalizar cada partida.

Se indica de una forma clara la posición del jugador, e incluso se presentan opciones para filtrar los resultados históricos o de la última semana si se quisiera ver el desempeño de la última partida, por ejemplo.

Imagen 4.13. Tabla de clasificación

Capítulo 5 – Implementación de Flip-Flop

5.1. Arquitectura

Flip-Flop consta de una serie de componentes relacionados con funciones claramente definidas que en conjunto forman la arquitectura de la aplicación. Se desarrollaron en paralelo dos componentes: Flip-Flop Manager y Flip-Flop Mobile. Flip-Flop Manager es una componente de backend destinada a brindar servicios a Flip-Flop Mobile, a la persistencia de los datos, y a consumir los servicios de Trello. Como se describió en el *Capítulo 4,* Trello es la herramienta de gestión elegida para el desarrollo de la presente tesina y la fuente de datos que FF utiliza para su funcionamiento. Por otro lado, la persistencia se realiza sobre una base de datos NoSQL [5.1] llamada MongoDB [5.2]. Flip-Flop Mobile es una aplicación nativa para el sistema operativo Android. Consume los servicios brindados por Flip-Flop Manager y utiliza los servicios de Google para incluir algunas mecánicas de juego.

En la *Imagen 5.1* se muestra la arquitectura de FF. La idea es presentar de una forma clara todos los componentes que fueron utilizados y desarrollados para lograr la construcción de FF. También mostrar las relaciones entre los mismos a través de las flechas.

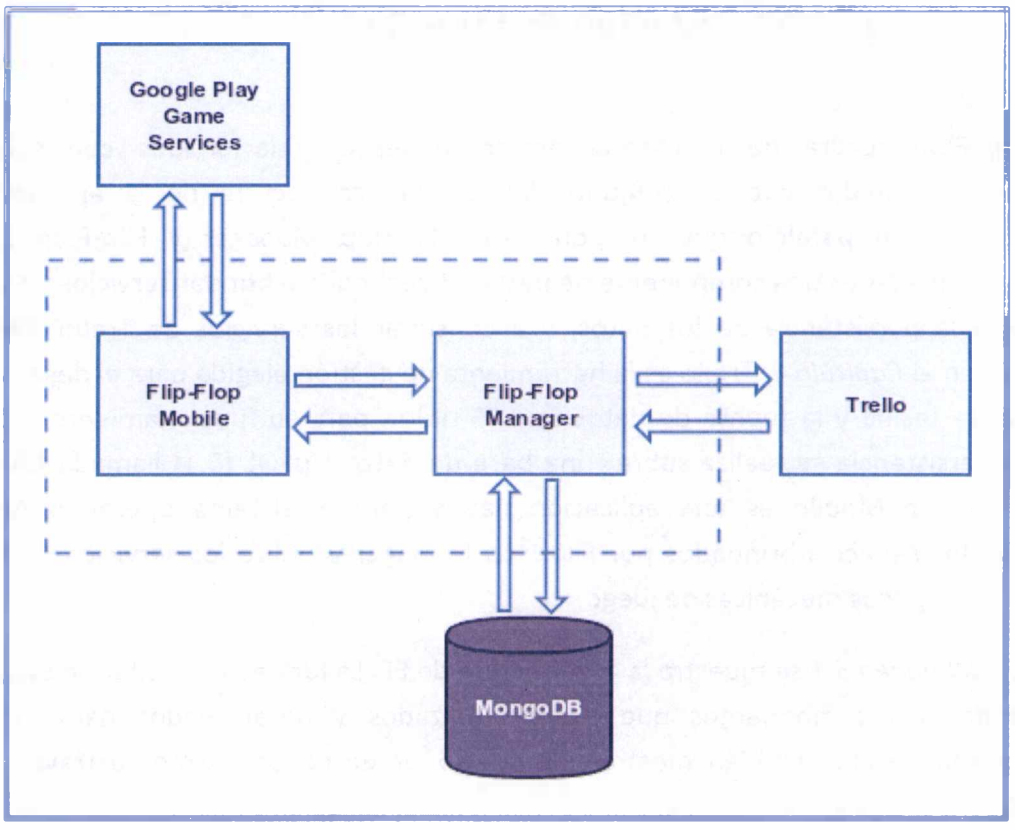

Imagen 5.1. Arquitectura de FF.

Los 5 (cinco) componentes son los siguientes:

- **Trello**: herramienta de gestión de proyectos.
- **Google Play Game Services**: servicios utilizados para incluir mecánicas de juego.
- **MongoDB**: base de datos NoSQL para la persistencia de la información.
- **Flip-Flop Manager**: aplicación Java que posee, entre sus principales funciones, la integración con Trello, el cálculo de puntos y los servicios utilizados por Flip-Flop Mobile.
- **Flip-Flop Mobile**: aplicación Android utilizada por los jugadores.

Las líneas punteadas marcan la separación entre los componentes desarrollados completamente por los tesistas (dentro de las líneas); y las herramientas y servicios que se utilizaron (fuera de las líneas).

5.1.1 Comunicación entre aplicaciones

Antes de entrar en los detalles de cada uno de los componentes anteriormente mencionados, se describe una característica importante de la comunicación y construcción de las aplicaciones y que es importante resaltar.

Tanto Trello como Flip-Flop Manager son sistemas RESTful, es decir, siguen los principios REST [5.3]. Ambos sistemas proveen interfaces que utilizan directamente HTTP [5.4] para obtener datos o indicar la ejecución de operaciones sobre ellos en formato JSON [5.5].

Podemos enumerar los siguientes diseños fundamentales claves a la hora de construir un sistema REST:

- Un protocolo cliente/servidor sin estado.
  Cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes.
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información.
  HTTP define un conjunto de operaciones (POST, GET, PUT, DELETE, entre otras). Con frecuencia estas operaciones se equiparan a las operaciones CRUD [5.6] en bases de datos que se requieren para la persistencia de datos.
- Una sintaxis universal para identificar los recursos.
  En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- Uso de hipermedia [5.7].
  Es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

5.2 MongoDB

MongoDB es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto. Forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (MongoDB llama ese formato BSON [5.8]), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

Permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de

datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta.

MongoDB ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad. Brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria. La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel y flexibilidad operativa.

Si se analiza el conjunto de datos con los que trabaja FF y cómo los utiliza, se encuentran las siguientes características:

- Datos de diferentes fuentes: los datos provenientes de cada uno de los dispositivos que utilicen los jugados; los datos provenientes de Trello; y todos aquellos datos producto del procesamiento y lógica del juego.
- FF realiza el seguimiento de todos los jugadores, esto es, tiene los datos históricos de toda la actividad de los jugadores que luego son utilizados en el juego y que nunca se borrarán.

Frente a este conjunto de datos y características de la aplicación, una base de datos NoSQL como MongoDB se ajusta mejor a este modelo que una base de datos relacional. Adicionalmente, si se quisieran utilizar estrategias de Big Data [5.9] en un futuro para estudiar o predecir comportamientos de los jugadores MongoDB ofrece una perfecta integración con herramientas para el procesamiento de este tipo de datos.

MongoLab

Para la base de datos de FF se creó un servidor de MongoDB en MongoLab [5.10]. MongoLab es un servicio de base de datos en la nube totalmente gestionado para bases de datos MongoDB. Incluye copias de seguridad, monitoreo y alerta. Actualmente gestiona más de 200.000 bases de datos a través de tres principales proveedores de la nube en 18 centros de datos en todo el mundo.

Se utiliza el servicio gratuito el cual ofrece 500 MB de almacenamiento, tamaño suficiente para la presente tesina. Ofrece otros planes de almacenamiento pagos.

5.3 Flip-Flop Manager

Flip-Flop Manager es una aplicación web RESTful diseñada como un componente de backend. Es una aplicación Java [5.11] desarrollada con el framework Spring Boot [5.12] y provee los servicios necesarios para el completo funcionamiento del juego.

Integración con Trello

Una de las funciones principales es consumir la API de Trello. Para esto fue necesario realizar un mapeo del modelo de Trello, para luego poder acceder a la información necesaria referida a tableros, tarjetas, listas y usuarios. Esta integración fue una de las primeras actividades desarrolladas, ya que las funciones que se crearon para el juego dependen íntegramente de las posibilidades que la API de Trello ofrece. Por este motivo el análisis y prueba de la misma fue una de las primeras tareas llevadas a cabo.

Persistencia en MongoDB

La configuración y conexión a la base de datos también reside en esta aplicación. Se utilizó el módulo Spring Data MongoDB [5.13] el cual ofrece integración con la base de datos. Como se describió, MongoDB es una base orientada a documentos. Los mismos se agrupan en colecciones.

Para el desarrollo del juego se definieron 4 (cuatro) colecciones de documentos:

- Games
  Se persisten los juegos que se vayan creando en FF con alguna información adicional. Por ejemplo, se guarda el identificador del tablero de Trello para los posteriores accesos y búsqueda de la información.
- Players
  En esta colección se persisten los datos necesarios de los jugadores. La mayoría de la información se obtiene de Trello, pero otros datos son propios del contexto de FF. Por ejemplo, se persiste el rol y el tablero con el que un jugador ingresa al juego. También se guarda una clave de Trello necesaria para acceder a los datos de cada jugador.
- Help_requests
  Aquí queda persistida la información asociada a las "SOLICITUDES DE AYUDA". Se guarda la tarea asociada al pedido de ayuda, qué jugador solicitó ayuda, quién lo ayudó, y los puntos que ese pedido de ayuda entregó al ayudante.
- Card_info
  Esta colección tiene el histórico de tareas que los jugadores resuelven. Para cada jugador se guardan todas las tareas que resolvió con la información completa para el posterior análisis de las mismas.

Cálculo de puntos

Puede verse como una de las principales funciones de la aplicación. Toda la lógica que se explicó en el capítulo anterior para el cálculo de puntos reside en Flip-Flop Manager. Posee la configuración y programación para que todos los viernes a las 18:00 horas se realice el cálculo de puntos. Más en detalle, el cálculo de puntos tiene asociadas las siguientes funciones:

1. Obtener de Trello el listado de tareas de la lista "FINALIZADO" de cada jugador.
2. Por cada una de las tareas, calcular el puntaje que gana el jugador.
3. Guardar en MongoDB la información procesada.
4. Notificar a Flip-Flop Mobile acerca de los nuevos puntajes.

Servicios para Flip-Flop Mobile

Se brindan servicios RESTful para ser consumidos desde Flip-Flop Mobile. Se incluyó la integración de Swagger [5.14] para presentar una API documentada. Esto facilitó el desarrollo de la componente que consume dichos servicios *(Imagen 5.2)*.

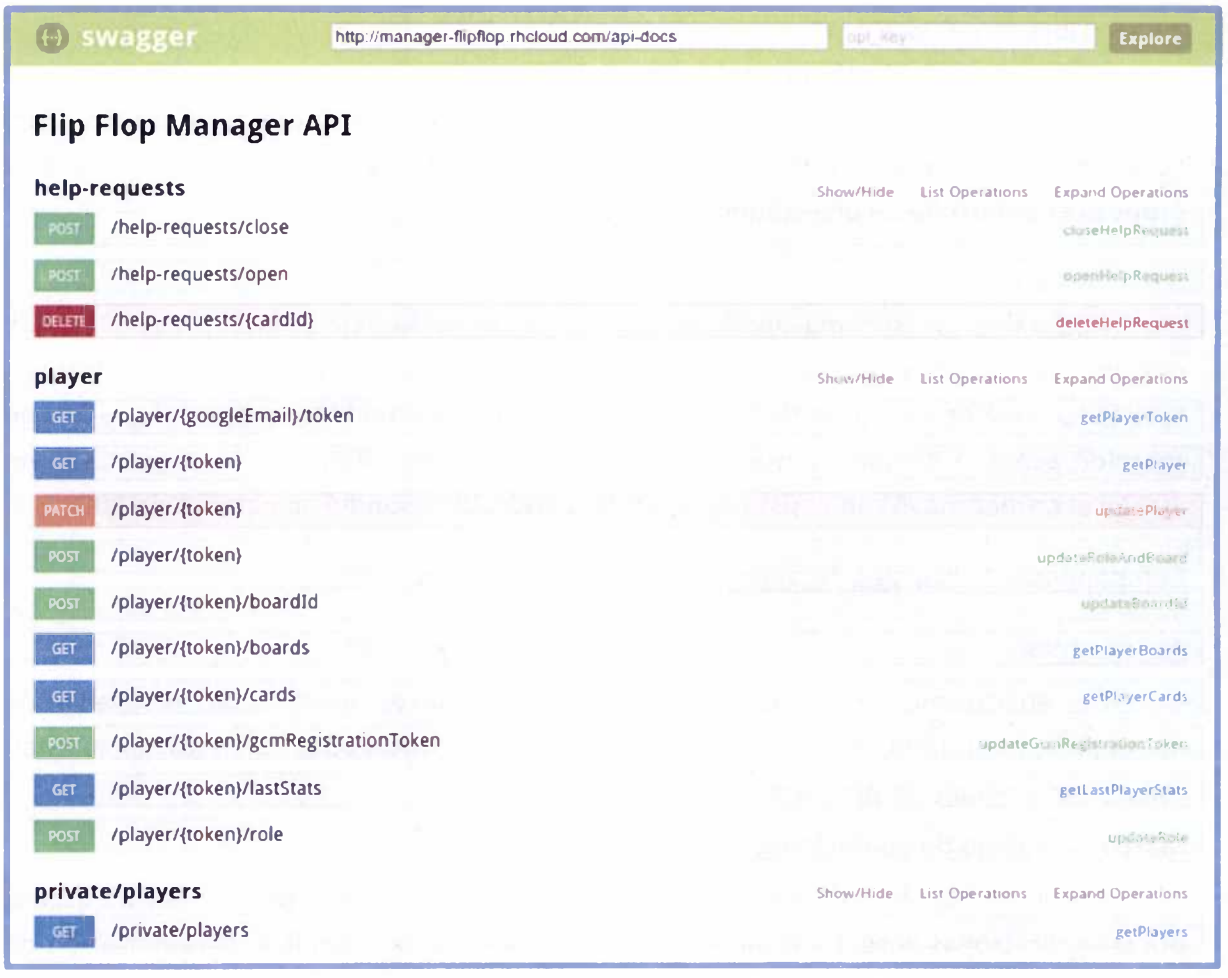

Imagen 5.2. Documentación de la API con Swagger

5.4 Flip-Flop Mobile

Flip-Flop Mobile es una aplicación nativa desarrollada para el sistema operativo Android. Una aplicación nativa se desarrolla con herramientas específicas que otorga el proveedor del sistema operativo para dispositivos móviles tales como el kit de desarrollo de software para la plataforma iOS [5.15] de Apple [5.16], las herramientas de desarrollo para Android de Google, o el .NET Compact Framework [5.17] de Microsoft [5.18].

Estas herramientas están optimizadas para una plataforma y permiten desarrollar aplicaciones centradas en la mejor experiencia de usuario, el rendimiento y funcionalidad, debido a que aprovechan características específicas del hardware ya que tienen acceso directo a las características nativas del dispositivo. Adicionalmente, utilizan sus propios lenguajes de programación:

- JAVA (Android)

- Objective-C [5.19] (iOS)
- Visual C++ [5.20] (Windows Mobile [5.21])

Estas aplicaciones son más fáciles de utilizar porque es a lo que los usuarios están acostumbrados y sus animaciones son más fluidas. Estas son las razones porque las que se propuso el desarrollo de una componente nativa.

5.4.1 Android

Android es un sistema operativo basado en el núcleo Linux [5.22]. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas o tabléfonos. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 y el primer móvil con el sistema operativo Android se vendió en octubre de 2008.

Componentes principales del sistema operativo de Android:

Aplicaciones

Las aplicaciones base incluyen un cliente de correo electrónico, programa de mensajería, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

Marco de trabajo de aplicaciones

Los desarrolladores tienen acceso completo a las mismas APIs del framework usadas por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades. Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.

Bibliotecas

Android incluye un conjunto de bibliotecas usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: bibliotecas de medios, bibliotecas de gráficos, entre otras.

Núcleo Linux

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software. Las aplicaciones se desarrollan habitualmente, como es el caso de FF, en el lenguaje Java con Android Software Development Kit [5.23] (Android SDK).

Todas las aplicaciones están comprimidas en formato APK [5.24] y se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos. Un archivo con extensión .apk (Application Package File) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR [5.25] de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android.

5.4.2 Google Play Store

Google Play Store [5.26] es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google. Esta plataforma permite a los usuarios navegar y descargar aplicaciones (desarrolladas mediante Android SDK), juegos, música, libros, revistas y películas. Las aplicaciones se encuentran disponibles de forma gratuita, así como también con costo. Pueden ser descargadas directamente desde un dispositivo con Android a través de la aplicación móvil Play Store.

Se utiliza Google Play Store para la distribución de Flip-Flop Manager de forma gratuita.

5.4.3 Google Play Game Services

Se describió anteriormente que para las funcionalidades de "LOGROS" y "TABLA DE CLASIFICACIÓN" se utilizaron los servicios de Google Play Game. Estos servicios ayudan a hacer los juegos más sociables, con logros, tablas de clasificación y con opciones de multijugador; así como también más robustos, almacenando la configuración y las partidas guardadas en la nube. Están disponibles en Android, y muchos en iOS o cualquier otro dispositivo conectado. Al construir un juego utilizando estos servicios de Google es posible centrarse en lo que es más importante: la creación de grandes experiencias de juego para los usuarios. Con los servicios se pueden incorporar:

- Logros
  Aumentan la participación y promueven diferentes estilos de juego.
- Tablas de clasificación social y pública
  Utilizan los círculos de Google para rastrear las puntuaciones altas entre todos los amigos y el mundo respectivamente.
- Almacenamiento en la nube
  Proporcionan una API sencilla y ágil para el almacenamiento de partidas y configuraciones.
- Multijugador en tiempo real
  Proporcionan servicios para facilitar la adición de juego cooperativo o competitivo en los dispositivos Android.

Capítulo 6 - Caso de estudio

6.1 Expectativas sobre la prueba

Las expectativas de las pruebas realizadas eran, en primer lugar, que los participantes no tengan dificultades para poder interactuar con el juego; que fácilmente capten el objetivo; y que puedan jugarlo más allá de los puntajes obtenidos. Por otro lado, se esperaba que quienes lo jueguen puedan comprender la importancia de las buenas prácticas y que para esto, se sientan atraídos por el juego y asuman como un desafío propio mejorar sus habilidades como jugadores.

6.2 ¿Cómo se trabajó?

La prueba fue llevada a cabo durante una semana en la empresa en la que los tesistas trabajan, Despegar.com [6.1]. Se seleccionó un equipo de desarrollo específico compuesto por: un Líder Técnico, un Analista, un Diseñador y dos Programadores. Se creó un tablero de Trello especialmente para la prueba y se les explicó a los integrantes del equipo el objetivo de FF.

6.3 Sobre el uso de FF

Todos los jugadores del equipo de desarrollo se mostraron entusiasmados por instalar la aplicación y asombrados por el concepto de FF: una componente lúdica que puede emplearse en el contexto laboral.

El proceso de logueo lo pudieron realizar sin ningún tipo de asistencia. En cambio, para utilizar la función de "SOLICITUDES DE AYUDA" se les mostró con un ejemplo cómo utilizar la misma.

Luego comenzaron a explorar las diferentes funciones y compararse entre ellos cuando lograron desbloquear algunos logros. Después de una semana de competencia, los tesistas pudieron observar cómo los jugadores estaban atentos al tablero de Trello. En particular, a las fechas de vencimiento de sus tareas para poder realizarlas a tiempo. Por otro lado, los jugadores valoraron que FF actualice las tarjetas de Trello cuando un integrante brindaba ayuda a otro. Otro punto de gran interés fueron los logros definidos en el juego.

En conclusión, los jugadores pudieron jugar a lo largo de una semana FF sin ningún tipo de problemas, se sintieron atraídos por el juego y se interesaron por cumplir las tareas que se asignaban. En consecuencia, se deduce que los objetivos planteados para esta prueba fueron cumplidos y se considera una experiencia exitosa.

Imagen 6.1. Instalación de FF en diversos dispositivos Android.

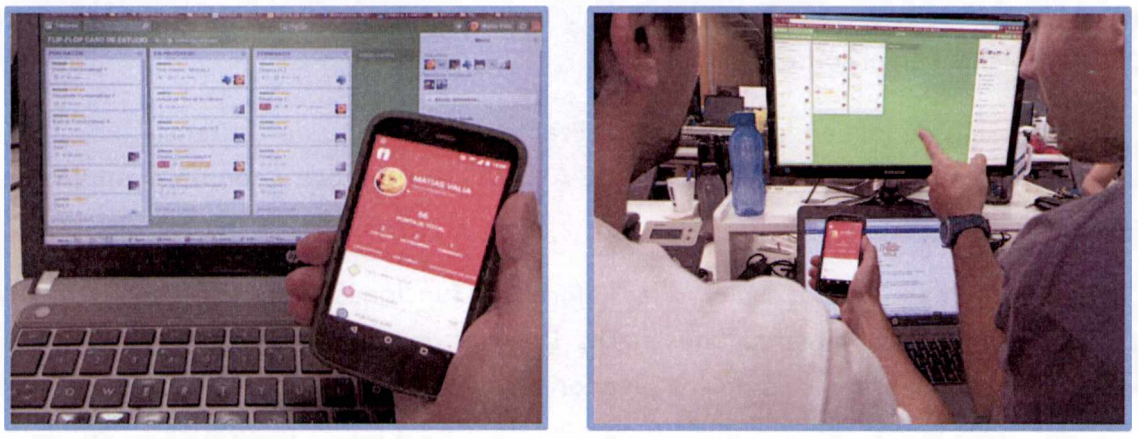

Imagen 6.2. Caso de estudio de FF.

Capítulo 7 - Conclusiones

Se comenzó realizando un estudio y documentación de la norma ISO 29110, en particular de las prácticas/actividades propuestas para el Proceso de Implementación de Software. Se analizaron y describieron los roles involucrados en las mismas. Como resultado del estudio de la norma ISO 29110 y sus roles, se lograron crear funcionalidades en FF directamente relacionadas al Proceso de Implementación de Software. Por ejemplo, la Puntualidad de un jugador está relacionada con la estimación de las tareas; y la Versatilidad está ligada a los roles de los jugadores.

Luego se investigó el concepto de Gamificación y múltiples mecánicas de juego que luego fueron incluidas en FF. Esta investigación permitió a los tesistas diseñar una experiencia de juego muy satisfactoria a través del diseño y desarrollo de diversas mecánicas de juego: logros/insignias, tablas de clasificación, sistema de puntos y desafíos cooperativos.

Se realizó un análisis de diferentes herramientas de Gestión de Proyectos y se seleccionó Trello principalmente por poseer una API web RESTful y múltiples opciones de personalización y configuración. Estas características fueron fundamentales para funciones de FF como la documentación de una tarea (utilizando la posibilidad de adjuntar archivos), y la definición de roles y niveles de dificultad (a través de etiquetas de Trello). Todas las funciones de Trello utilizadas en FF fueron documentadas en la tesina incluyendo ejemplos de uso.

Finalmente, los tesistas lograron desarrollar una componente lúdica utilizando las mecánicas y dinámicas de juego para incentivar la utilización de buenas prácticas, en el desarrollo de software. Dicha componente es una aplicación nativa desarrollada para el sistema operativo Android. Adicionalmente, y con el fin de enriquecer la aplicación, se investigaron los servicios que proporciona Google para sistemas de juegos y para notificaciones. Ambos servicios se incluyeron satisfactoriamente. Por otro lado, el desarrollo de una aplicación backend permitió la integración con los servicios de Trello y la persistencia de datos.

En conclusión, el resultado final es una aplicación completamente diseñada por los tesistas que hace uso de las últimas tecnologías y servicios de juegos de Google para lograr una experiencia de juego dinámica y motivadora para los usuarios con el objetivo principal de incorporar calidad en las actividades diarias a través del uso de las mejores prácticas.

Capítulo 8 - Trabajos futuros

Se describen a continuación una serie de mejoras y funcionalidades adicionales con las que se puede extender el comportamiento de FF.

Flip-Flop Multiplataforma

En la presente tesina se planteó el desarrollo de una aplicación nativa para el sistema operativo Android. Dicha elección se fundamenta, como se mencionó en el *Capítulo 5*, en los beneficios obtenidos al desarrollar una aplicación utilizando herramientas específicas que otorga el proveedor del sistema operativo elegido. Sin embargo, aquellos desarrolladores que no poseen un dispositivo Android no pueden utilizar FF. Por esta razón, una de las principales mejoras para FF es extenderlo. El desarrollo de una versión web puede ser un punto de partida. Otra posibilidad es el desarrollo de las versiones para iOS y Windows Mobile.

Funciones adicionales para el Líder Técnico

El rol de Líder Técnico puede extenderse con funcionalidades diseñadas específicamente para dicho rol. Partiendo de la base que FF guarda el historial de las tareas de todos los jugadores, se podrían presentar diferentes tablas de clasificación para que el Líder Técnico pueda analizar mejor la actividad de los integrantes de su equipo de desarrollo. También se puede incluir un módulo de estadísticas en donde a través de gráficos y tablas se pueda analizar la actividad de los jugadores.

Configuración inicial de Trello automatizada

En el *Capítulo 4,* se describieron las características que deben poseer el tablero, las listas, las etiquetas y las tarjetas de Trello para que sean válidas en FF. La creación de estos elementos se debe realizar de forma manual, y esto es propenso a errores. En consecuencia, una mejora que ven los tesistas es utilizar la API de Trello para crear automáticamente un tablero para FF. De esta forma, cuando un jugador ingresa a FF por primera vez podría elegir un tablero de la lista o crear uno nuevo. En el caso de crear uno nuevo, debería ingresar el nombre para el tablero y a través de los servicios de Trello se crearía un tablero válido para FF.

Ayuda para el uso de FF

Actualmente las aplicaciones móviles brindan algún tipo de ayuda para orientar al usuario sobre el uso de la misma y mostrarle funciones que pueden estar ocultas y que quizá el usuario no llegue a descubrir. Una mejora muy importante para FF es la inclusión

de ayudas. Una opción que los tesistas evaluaron es el uso de la librería ShowcaseView [7.1]. La misma permite crear una pantalla de introducción que sirve como un tutorial en el que se pueden resaltar partes de la aplicación y de esta forma orientar al usuario sobre el uso. Este tutorial aparece solo la primera vez que se usa la aplicación y se pueden ir agregando tutoriales más pequeños a medida que se adicionen funcionalidades nuevas.

Capítulo 9 – Bibliografía

[1.1] ISO 29110-2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs)

[1.2] http://es.wikipedia.org/wiki/Software_de_administraci%C3%B3n_de_proyectos

[1.3] Gamification by Design. Implementing Game Mechanics in Web and Mobile Apps. Gabe Zichermann, Christopher Cunningham, ISBN 978-1449397678, Año 2013

[1.4] Understanding gamification mechanisms for software development (http://dl.acm.org/citation.cfm?id=2494589) Daniel J. Dubois - Giordano Tamburrelli

[1.5] Trello, https://trello.com/

[1.6] Servicios de Juegos de Google Play, https://developers.google.com/games/services/

[2.1] Pantaleo Guillermo, Calidad en el desarrollo de software, Editorial Alfaomega, ISBN 978-987-1609-23-9, Año 2011

[2.2] ISO 9126, Software engineering - Product quality

[2.3] ISO 25000, Software Engineering - Software product Quality Requirements

[2.4] ISO 14598, Software engineering -- Product evaluation

[2.5] Chrissis, Mary, CMMI 2ed Guía para la interpretación de procesos y mejora de productos, Editorial Pearson Educación,ISBN 978-84-7829-096-3, Año 2009

[2.6] ISO .15504: 2004 Information Technology - Process assessment

[2.7] ISO 12207, Systems and software engineering - Software life cycle processes

[2.8] ITIL "Information Technology Infrastructure Library" - https://www.axelos.com/best-practice-solutions/itil

[2.9] ISO 20000 , Information Technology - Service management

[2.10] IRAM-ISO. 9001:2008, "Sistema de gestión de la calidad. Requisitos" Segunda Edición 2008 IRAM

[2.11] ISO 90003:2004, "Software engineering - Guidelines for the application of ISO 9001:2000 to computer" 2004, ISO

[2.12] Moprosoft "Modelo de Procesos para la Industria del Software" - http://www.moprosoft.com.mx/

[2.13] MPS BR "Mejora de Procesos de Software de Brasil" - http://www.softex.br/mpsbr/

[2.14] Piattini, Oktaba, Orozco, "COMPETISOFT. Mejora de procesos software para pequeñas y medianas empresas", Editorial Ra-Ma, Año 2008

[2.15] ISO TR 29110-5-1-1:2012, "Management and engineering guide" 2012, ISO

[3.1] Gamification in software engineering – A systematic mapping. Oscar Pedreira, Félix García, Nieves Brisaboa, Mario Piattini.

[3.2] Farm Ville, http://zynga.com/games/farmville

[3.3] Diner Dash, http://www.playfirst.com/games/viewGame/70

[3.4] Diaper Dash, http://www.gamehouse.com/download-games/diaper-dash

[3.5] Sally's Salon, http://www.gamehouse.com/download-games/sallys-salon

[3.6] Flight Control, https://es.wikipedia.org/wiki/Flight_Control

[3.7] SAPS, http://www.gamification.co/2010/10/18/cash-is-for-saps/

[3.8] Gabe Zichermann, https://en.wikipedia.org/wiki/Gabe_Zichermann

[3.9] Gilt Noir , http://www.gilt.com/

[3.10] Nicole Lazzaro, https://www.linkedin.com/in/nicolelazzaro

[3.11] Lazzaro, "Why We Play Games: Four Keys to More Emotion Without Story", 2004

[3.12] Bartle,"HEARTS, CLUBS, DIAMONDS, SPADES: PLAYERS WHO SUIT MUDS", http://mud.co.uk/richard/hcds.htm

[3.13] Richard Bartle, https://en.wikipedia.org/wiki/Richard_Bartle

[3.14] Matriz SEKA, tipos de jugadores de Richard Bartle, http://elviejoclub.blogspot.com.ar/2015/04/matriz-seka-tipos-de-jugadores-de.html

[3.15] Is it all a game? Understanding the principles of gamification. Karen Robson,Leyland Pitt , Kirk Plangger, Jan H. Kietzmann, Ian McCarthy.

[3.16] Facebook, https://www.facebook.com

[3.17] Twitter, https://twitter.com/

[3.18] Boom Beach, http://www.playboombeach.com/

[3.19] Foursquare, https://es.foursquare.com/

[4.1] Android - History, https://www.android.com/history/#/marshmallow

[4.2] Método Kanban, http://www.javiergarzas.com/2011/11/kanban.html

[4.3] API,https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

[4.4] Richardson, Amundsen, "RESTful Web APIs", Año 2013

[4.5] URI, https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

[4.6] Google Drive, https://www.google.com.ar/intl/es/drive/using-drive/

[4.7] Dropbox, https://www.dropbox.com/es/

[4.8] Box, https://www.box.com/

[4.9] OneDrive, https://onedrive.live.com/about/es-ar/

[4.10] Cuentas de Google, https://myaccount.google.com/intro

[5.1] Fowler, "NoSQL Distilled", ISBN 978-0321826626

[5.2] Manual de MongoDB, https://docs.mongodb.org/manual/

[5.3] Webber, "REST in Practice", ISBN 978-0596805821, Año 2010

[5.4] HTTP, https://www.w3.org/Protocols/

[5.5] JSON, http://www.json.org/

[5.6] CRUD, https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

[5.7] Hipermedia, http://www.hipertexto.info/documentos/hipermedia.htm

[5.8] BSON, http://bsonspec.org/

[5.9] Big Data, https://docs.mongodb.org/ecosystem/use-cases/hadoop/

[5.10] MongoLab, https://mongolab.com/

[5.11] Java, https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)

[5.12] Spring Boot, http://projects.spring.io/spring-boot/

[5.13] Spring Data MongoDB, http://projects.spring.io/spring-data-mongodb/

[5.14] Swagger, http://swagger.io/

[5.15] iOS, http://www.apple.com/es/ios/what-is/

[5.16] Apple, http://www.apple.com/

[5.17].NET Compact Framework,
https://en.wikipedia.org/wiki/.NET_Compact_Framework

[5.18] Microsoft, https://www.microsoft.com/es-ar/

[5.19] Objective-C,
https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html

[5.20] Visual C++, https://en.wikipedia.org/wiki/Visual_C%2B%2B

[5.21] Windows Mobile, http://www.windowsphone.com/es-es/how-to/wp8

[5.22] Linux, https://www.linux.com/

[5.23] Android SDK, http://developer.android.com/intl/es/tools/sdk/tools-notes.html

[5.24] APK, https://es.wikipedia.org/wiki/APK_(formato)

[5.25] JAR, https://es.wikipedia.org/wiki/Java_Archive

[5.26] Google Play Store, https://play.google.com/store

[7.1] ShowcaseView, https://github.com/amlcurran/ShowcaseView

Todas las url mencionadas fueron accedidas en Febrero de 2016.