UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
Maestría en Ingeniería de Software

End User Grammar Extended for Business Processes - EUGEBP

Gramática de Usuario Final Extendida para Procesos de Negocios

Autor
Enrique Eduardo Aramayo
Director
Dr. Matías Urbieta
Co-Director
Dr. Gustavo Rossi

Tesis presentada para obtener el grado de Magister en Ingeniería de Software.

Agosto – 2020

Resumen

El concepto proceso de negocio está estrechamente vinculado a la forma en la que una organización gestiona sus operaciones. Conocer y comprender las operaciones de una organización es un punto clave que se debe tener presente dentro del proceso de desarrollo de software. A su vez, el enfoque de desarrollo dirigido por modelos denominado MockupDD captura requerimientos usando prototipos de interfaz de usuario denominados Mockups. Los usuarios finales pueden comprender fácilmente dichos prototipos y realizar anotaciones sobre los mismos. Este enfoque se basa en ésta característica principal y a partir de la misma genera valiosos modelos conceptuales que luego pueden ser aprovechados por todos los integrantes de un equipo de desarrollo de software. Utilizar el lenguaje natural para realizar anotaciones sobre los Mockups es un aspecto clave que puede ser aprovechado. En éste último aspecto una rama de la inteligencia artificial denominada "Natural Language Processing – Procesamiento del Lenguaje Natural" (NLP) viene realizando importantes aportes vinculados al uso y al aprovechamiento del lenguaje natural de las personas.

El presente trabajo de tesis propone una nueva técnica denominada "End User Grammar Extended for Business Processes – Gramática de Usuario Final Extendida para Procesos de Negocios" (EUGEBP). La misma está compuesta por un conjunto de reglas de redacción diseñada para ser aplicada sobre Mockups, y por una serie de pasos que permiten procesar dichas anotaciones con el propósito derivar procesos de negocios desde las mismas. Esto se logra a través de la identificación de los elementos que componen los procesos de negocios y de las relaciones que existen entre ellos. En esencia el presente trabajo propone utilizar las anotaciones de usuario final realizadas sobre los Mockups en lenguaje natural y a partir de las mismas derivar procesos de negocio. Las anotaciones del usuario final tienen como objetivo ayudar a describir las interfaces de usuario, pero también pueden ayudarnos a identificar los procesos de negocio que el sistema debe soportar. Mientras en analista recopila información para el desarrollo de una aplicación, implícitamente también está describiendo los procesos de negocio de la organización.

Palabras Claves: BP, BPMN, NLP, Mockup, MockupDD, EUG, EUGEBP.

Agradecimientos

A mis padres Cecilio y Susana, por enseñarme el camino de la educación con mucha sabiduría y cariño.

A mi esposa Luciana, por haberme acompañado en todo este proceso con mucho amor y paciencia.

A mi hijo Santiago, por haberme escuchado y comprendido.

A mi amigo y colega Pablo, por acompañarme en el cursado de la maestría con alegría.

A todos mis seres queridos y amigos, por brindarme su apoyo.

Al Dr. Matías Urbieta, por su entusiasmo, guía y atención.

Tabla de Contenidos

Índice de Anexos

Índice de Figuras

Índice de Tablas

Capítulo 1 Introducción

En este capítulo se presenta la motivación de la tesis y el contexto en el que se desarrolló la misma. Se describe la situación problemática inicial que da origen a la tesis. Además, se describe el objetivo principal que se persigue y la metodología de trabajo que se llevará a cabo en consecuencia. Por último, se mencionan los temas que se tratarán y la organización de la tesis en sí misma.

1.1 Motivación de la tesis

Las organizaciones en general necesitan documentar la forma en la que se gestionan sus operaciones. Conocer y comprender las mismas es un punto relevante que se debe tener presente dentro del proceso de desarrollo de software. En el ámbito académico las operaciones de una organización son mencionadas como procesos de negocio. Un proceso de negocio se define como: *"una unidad de trabajo persistente iniciada por un suceso dentro del negocio denominado evento. El proceso es direccionado mediante reglas que activan la ejecución de tareas y sub-procesos con cada transición de estado ejecutándose dentro de una transacción. A las tareas y sub-procesos se les asignan recursos, a los cuales se los denomina unidades organizativas que son capaces y están autorizadas a ejecutar roles específicos dentro de los procesos"* [1]. Un enfoque utilizado en las organizaciones para la gestión de los procesos de negocios es "Business Process Management – Gestión de Procesos de Negocio" (BPM) [2]. Su propósito fundamental es definir procesos de negocios comprensibles para toda la organización. Del mismo se deprende "Business Process Model and Notation –Notación y Modelos de Procesos de Negocio" (BPMN) y un aspecto clave de dicha notación es que permite representar visualmente una secuencia detallada de flujos de información y de actividades organizacionales necesarias para comprender un proceso de negocio[3]. Se diferencia de la creación de mapas de procesos de negocio, en que realiza diagramas de procesos actuales para propósitos tales como la estandarización, la capacitación a empleados, el control de calidad, entre otros.
Los procesos de negocio son el sistema arterial de las organizaciones y de las redes inter-organizacionales. Cualquier falla en los mismos pueden detener la vida corporativa de la organización. Los procesos de negocio son el alma de una organización. Las crecientes demandas de globalización, integración, estandarización, innovación, agilidad y eficiencia operativa, junto con las oportunidades generadas por las tecnologías digitales, han aumentado el apetito para reflexionar y mejorar los procesos de negocio existentes y diseñar nuevos procesos de negocio completos. En consecuencia, en las últimas dos décadas ha surgido el conjunto integral de herramientas, técnicas, métodos y metodologías completas para respaldar todas las etapas del ciclo de vida de los procesos de negocio dentro de las organizaciones, llamado "Business Process Management" (BPM). El mismo consolida una gran cantidad de herramientas y enfoques provenientes de diversas disciplinas, que incluyen Ingeniería Industrial, Gestión de Operaciones, Gestión de Calidad, Gestión de Capital Humano, Gobierno Corporativo, Ciencias de la Computación e Ingeniería de Sistemas de Información [4].

Además, en el contexto de los "Business Process – Procesos de Negocios" (BP), existe una rama de la inteligencia artificial que puede ser aplicada al estudio de los mismos denominada "Natural Language Processing – Procesamiento del Lenguaje Natural" (NLP). El procesamiento del lenguaje natural es el estudio del modelado matemático y computacional de varios aspectos del lenguaje y el desarrollo de una amplia gama de sistemas afines. Los sistemas de procesamiento del lenguaje natural juegan un papel crucial en nuestra comunicación con las máquinas e incluso entre nosotros. La importancia de NLP para el progreso en las telecomunicaciones y la informática no puede subestimarse [5]. Los sistemas de NLP incluyen sistemas de reconocimiento de voz, comprensión del lenguaje y generación de lenguaje. Diversas investigaciones [6][7][8][9][10] mencionan que es posible utilizar técnicas y herramientas NLP para producir modelos de procesos de negocios basados en el procesamiento del lenguaje natural [11]. Esto se debe a que gran parte de los procesos de

negocios, dentro de las organizaciones, están descriptos en lenguaje natural. Aplicar NLP sobre dichas descripciones es una idea que se profundiza en las investigaciones mencionadas.

Por otro lado, un factor clave del éxito de los proyectos es la adopción de las metodologías ágiles. En los últimos tiempos el uso de las mismas como un marco de referencia creció de forma considerable [12]. Los miembros de un equipo que llevan a cabo la elicitación de requerimientos, para comprender las necesidades de los stakeholders y de los procesos de negocios de una organización, a menudo adoptan el marco de trabajo ágil para desarrollar sus tareas. Cuando se capturan requerimientos de aplicaciones web uno de los artefactos más utilizado por los practicantes ágiles es el "Wireframe" o "Mockup" [13]. Los Mockups permiten capturar decisiones básicas en un solo documento que sirve como referencia para el trabajo de diseño visual. Pueden contener distintos niveles de detalle y en general su uso es bastante ágil [14]. En esencia un Mockup es un prototipo o bosquejo de una interfaz de usuario. Uno de los enfoques que propone utilizar los mismos como una herramienta central es el denominado "Mockup Driven Development - Desarrollo Dirigido por Mockups" (MockupDD). El mismo propone un enfoque ágil y respeta la filosofía propuesta por las metodologías ágiles [15][16]. El uso del enfoque mencionado implica que el usuario final comente o haga anotaciones sobre el Mockup. De ésta manera se logra la participación fácil y sencilla del usuario final. Y, al mismo tiempo se documentan fuertes bases de conocimiento sobre los procesos de negocio de la organización que luego pueden ser aprovechadas y explotadas. Este enfoque entonces, en lugar de descartar los Mockups, transforma los mismos en especificaciones de UI independientes de la plataforma, y los enriquece gradualmente para luego obtener un conjunto del dominio, navegación y modelos de presentación que pueden ser además enriquecidos interactivamente. De ésta forma los Mockups sirven como una herramienta de fácil comprensión que dispara requerimientos, desde su inicio, muy rápidamente.

A su vez, el modelado y la comprensión de los procesos de negocio de una organización es responsabilidad de los Analistas de Requerimientos [17]. Sin embargo, en las primeras etapas del proceso de desarrollo de software el conocimiento sobre los mismos puede ser limitado o nulo, además es probable que los integrantes del equipo de desarrollo desconozcan los procesos de negocios a los cuales el desarrollo va a dar soporte. Para dar solución a la situación previamente descripta BPMN propone diversos métodos para la identificación de los procesos de negocio: entrevistas con los usuarios (expertos en sus dominios), lectura de los documentos de la organización (fuentes como informes, formularios, políticas comerciales, etc.), entre otras [18] [19]. Dentro del marco de trabajo de MockupDD no existe una forma de trabajar en la identificación de BP. Lo que existe es una forma de trabajo en la que se utilizan los Mockups para recopilar explícitamente datos vinculados a los requerimientos como por ejemplo "End User Grammar – Gramática de Usuario Final" (EUG) [15][20]. EUG es un catálogo de anotaciones de usuario final que permite enriquecer las especificaciones que se registran dentro de un Mockup. Cada anotación colocada sobre el Mockup representa una especificación relativa independiente, para la instancia, el contexto, la navegación, el comportamiento o cualquier otro aspecto que pueda ser representado a través de una representación visual de la "User Interface – Interfaz de Usuario" (UI). EUG es una poderosa técnica que propone una notación coloquial y amigable para describir: datos, navegación, requisitos de interacción y de negocios según especificaciones realizadas sobre los Mockups, que se apoya en MockupDD. Sin embargo, la misma no se usa para recopilar explícitamente información sobre procesos de negocios. Si, por ejemplo, tratamos de modelar una situación de proceso de negocio en la que se requiere reconocer cual es el papel que juegan diferentes socios de la organización (proveedores, clientes, empleados) en una determinada situación de venta de productos (que involucra participantes, actividades que realizan los mismos, eventos importantes, interacciones, flujos de secuencia y de mensajes, todos elementos BPMN), la técnica EUG no sería suficiente para describirla. En éste contexto existe una importante brecha entre lo que podemos describir sobre un Mockup y los BP.

La presente tesis propone realizar anotaciones de usuario final en lenguaje natural y simple, pero de una manera sistematizada (es decir siguiendo una serie de reglas de redacción), sobre los Mockups de la aplicación. Y además propone utilizar dichas anotaciones para, colaborar con la captura del conocimiento que poseen los usuarios sobre los procesos de negocios de la organización, y contribuir con la identificación de los mismos. En otras palabras se propone colaborar con la fase del "Levantamiento del Proceso" [19] mediante la adopción del artefacto Mockup proveniente de las metodologías ágiles [15]. Si bien las anotaciones de los usuarios finales sobre los Mockups están orientadas a ayudar al proceso de desarrollo de software en su fase inicial, las mismas también pueden ser de gran ayuda para identificar los procesos de negocios a los que estamos dando soporte. En [21] se menciona: "*Diseñar y desarrollar software es, en esencia, describir los procesos de negocios de una organización a un muy bajo nivel de abstracción"*. Es decir que mientras recopilamos información para el desarrollo de una aplicación implícitamente estamos describiendo los procesos de negocios de la organización. La presente tesis entonces propone aprovechar las anotaciones realizadas sobre los Mockups (por los stakeholders) e inferir desde las mismas los BP de la organización a los cuales se está dando soporte. Este es el vacío que se aborda con el presente trabajo y la motivación del mismo.

1.2 Objetivo de la tesis

El objetivo general de la tesis es proponer una nueva técnica que permita capturar la gestión operativa de una organización. En éste contexto, la intención de la tesis es contribuir con una nueva forma de identificar procesos de negocios combinando características provenientes de EUG, BPMN y NLP, y aplicar dicha combinación sobre el artefacto central denominado Mockup.

1.2.2 Objetivos específicos

El objetivo general se desagrega en los siguientes objetivos específicos para su tratamiento:

- Realizar un estudio del estado del arte de técnicas y/o metodologías vinculadas al estudio y descubrimiento de los procesos de negocios dentro de una organización.
- Profundizar el estudio del enfoque EUG y descubrir cuales características de dicho enfoque son aplicables a la descripción de procesos de negocios
- Profundizar el estudio de técnicas que aplican la rama de la inteligencia artificial NLP para descubrir los procesos de negocios de una organización
- Proponer un conjunto de anotaciones sistemáticas que permitan identificar a los elementos que componen los procesos de negocios y a los procesos de negocios en sí.
- Evaluar el conjunto de anotaciones propuesto validando el mismo mediante un experimento.

1.3 Metodología de la investigación

Nuestro estudio parte de una fase de carácter bibliográfico dentro del cual se realizó una elaboración teórica de los conceptos "proceso" y "proceso de negocio". Nuestra perspectiva inicial se encuentra ya establecida en la "motivación de la tesis" presentada. Sobre la misma se trabajó en un proceso de afinamiento de los conceptos presentados.

Se planteó utilizar *investigación exploratoria* para determinar el estado del arte de los principales temas de investigación: proceso de negocio, gramática de usuario final y procesamiento del lenguaje natural para capturar procesos de negocio.

Sobre la base mencionada en los párrafos anteriores, en primer lugar, se diseñó una prueba de concepto destinada a probar la base teórica generada.

La metodología de la investigación que se utilizó para el desarrollo de la presente tesis está compuesta por las siguientes actividades:

- Determinar la situación problemática inicial:
  Identificar una situación puntual que proporcione la oportunidad de investigar y proporcionar un aporte ante la misma.

- Revisar fuentes de información seleccionadas:
  Luego de realizar la búsqueda de bibliografía vinculada a la tesis se procedió a seleccionar las principales fuentes de información que guiaron el desarrollo de la tesis. Luego se procedió a la lectura puntual de las mismas.

- Delimitar los principales temas de investigación:
  En nuestro caso se estudiaron y explotaron los siguientes temas: proceso de negocio, gramática de usuario final y procesamiento del lenguaje natural.

- Desarrollar la investigación:
  Redactar el marco teórico de la tesis delimitando el alcance de la misma. Establecer el estado del arte para el contexto de la tesis. Determinar a qué tipo de aplicaciones va dirigida la propuesta. Describir el aporte que se espera brindar al conocimiento teórico. Proponer la técnica "End User Grammar Extended for Business Processes – Gramática de Usuario Final Extendida para Procesos de Negocios" (EUGEBP) y describir la misma en detalle.
  Planificar la prueba de concepto. Diseñar de la prueba de concepto. Verifica que los escenarios de evaluación y todos los componentes de la prueba de concepto estén integrados correctamente. Producir datos para la evaluación. Capacitar a los usuarios para la prueba de concepto.

- Evaluar la técnica "End User Grammar Extended for Business Processes – Gramática de Usuario Final Extendida para Procesos de Negocios" (EUGEBP) con usuarios finales:
  Los usuarios y el tesista llevarán a cabo la evaluación y retro alimentación de la prueba de concepto.

- Analizar los resultados obtenidos y elaborar conclusiones:
  Determinar en qué medida se han cumplido los objetivos planteados.

1.4 Estructura de la tesis

De acuerdo a la metodología de la investigación planteada, a continuación, se presenta la estructura de la tesis.

Capítulo 1 Introducción
Este capítulo presenta la situación problemática inicial que motiva la presente tesis. Describe las características generales del escenario de investigación, el objetivo que persigue la tesis y la metodología de la investigación que se implementará para alcanzarlo.

Capítulo 2 Estado del Arte
En este capítulo se describe el estado del arte de las diferentes fuentes que dan sustento a la investigación: procesos de negocios, gramática de usuario final, procesamiento del lenguaje natural. El estado del arte se presenta como el marco teórico sobre el cual se propone la técnica "EUGEBP - End User Grammar Extended for Business Processes - Gramática de Usuario Final Extendida para Procesos de Negocios".

Capítulo 3 Técnica "End User Grammar Extended for Business Processes"
"End User Grammar Extended for Business Processes - Gramática de Usuario Final Extendida para Procesos de Negocios" (EUGEBP).
En este capítulo se describe en detalle la técnica que permite identificar y derivar procesos de negocios.

Capítulo 4 Herramientas, para el análisis de oraciones, aplicadas a EUGEBP
En éste capítulo se describen básicamente dos herramientas que pueden asistir y ser aplicadas a EUGEBP: "PEG.js" y "Stanford CoreNLP".

Capítulo 5 Evaluación de EUGEBP
En éste capítulo se describe la construcción de una prueba de concepto y la implementación de la misma. Se evalúa el funcionamiento de la técnica propuesta.

Capítulo 6 Conclusiones
En éste capítulo se recapitulan los principales hallazgos. Se detalla el aporte que surge a partir de la investigación y se presentan las conclusiones de la investigación realizada.

Capítulo 2 Estado del arte

En el estado del arte nos interesa presentar todos aquellos conceptos sobre los cuales se apoya la tesis. Es decir que en éste capítulo se presenta la base teórica utilizada para desarrollar la tesis. En éste contexto, en primer lugar, se define y clarifica el concepto proceso. A continuación, se define proceso de negocio. Seguidamente presentamos brevemente BPMN. Cabe mencionar que éste último tema sirvió de marco fundamental de referencia para la tesis.
En segundo lugar, presentamos una rama de la inteligencia artificial denominada NLP (Natural Language Processing – Procesamiento del Lenguaje Natural). Y, orientamos dicha presentación hacia aquellas características que están vinculadas estrechamente con el estudio de los procesos de negocio.
Por último, presentamos formalmente el "proceso MockupDD" y la técnica "End User Grammar - Gramática de Usuario Final" (EUG). El origen de dicha técnica, el artefacto principal que sustenta la misma (el Mockup), mostrando lo que propone dicho enfoque y lo que se puede producir a partir del mismo.

2.1 Definición de Proceso

La palabra proceso es de origen latino, del vocablo "processus, de procederé", que viene de pro (para adelante) y cere (caer, caminar), lo cual significa ir hacia adelante o ir hacia un determinado fin.
Según el diccionario de la real academia española ésta palabra es definida como la acción de ir hacia adelante, o como un conjunto de fases sucesivas de un fenómeno natural o de una operación artificial.
Según la norma ISO 9001:2015 el concepto de proceso es utilizado dentro de otro denominado "Enfoque basado en procesos", el cual no es tema de estudio de la presente tesis. Sin embargo, en esencia dentro de la norma, un proceso es un conjunto de numerosas actividades relacionadas entre sí. Las actividades utilizan recursos y transforman los elementos de entrada en resultados (ver figura 1)

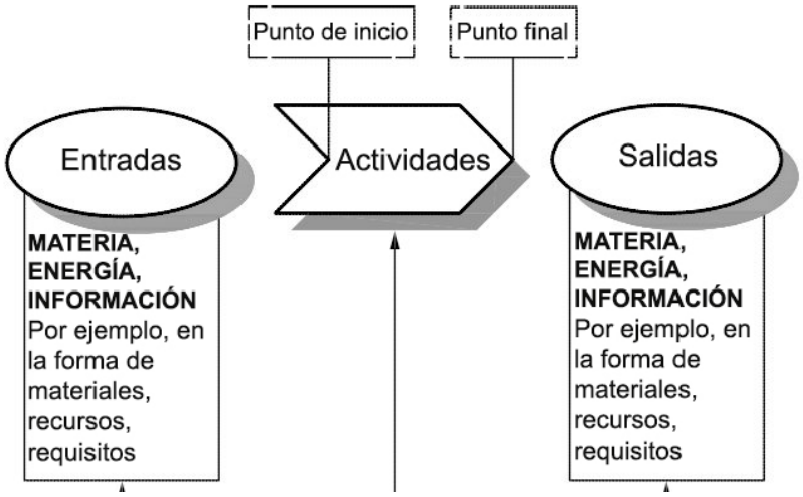


Figura 1 – Proceso según la norma ISO 9001

Para el contexto de la tesis, se interpreta proceso como un conjunto de actividades que se relacionan entre sí. El proceso puede hacer uso de diferentes recursos según su necesidad. Y, para todo proceso se puede identificar un origen y un final.

2.1.1 Definición de Proceso de Negocio

Desde una perspectiva muy general podemos decir que un proceso de negocio es la forma en la que una organización gestiona sus operaciones. En los siguientes párrafos indagamos en el concepto "proceso de negocio" en mayor profundidad.

La definición más adecuada, para el desarrollo de la tesis, encontrada dentro de la bibliografía relacionada es la que se menciona a continuación:

"*Un **proceso de negocio** es una **unidad de trabajo** persistente iniciada por un suceso dentro del negocio denominado **evento**. El proceso es direccionado mediante **reglas** que activan la ejecución de **tareas y sub-procesos** con cada transición de estado ejecutándose dentro de una transacción. A las tareas y sub-procesos se les asignan **recursos**, a los cuales se los denomina unidades organizativas que son capaces y están autorizadas a ejecutar **roles** específicos dentro de los procesos"* [1].

La definición presentada describe a un proceso de negocio como un componente, dentro de la organización, bien identificado que persiste en el tiempo. El proceso de negocio tiene un inicio que se activa cuando la organización lo requiere. Una vez activado una serie de actividades se disparan y son guiadas por normas que rigen dentro de la organización. La organización provee a las actividades de los recursos necesarios para su correcto desempeño. Cada recurso cumple un papel específico dentro del proceso de negocio. Es decir que la definición se concentra en denotar la importancia de lo que sucede dentro del proceso de negocio en sí. La idea principal entonces está puesta en analizar qué es lo que sucede dentro del proceso de negocio.

2.1.2 Tipos de Procesos de Negocio

Los procesos de negocio se encuentran presentes dentro de toda la organización. Desde un punto de vista global se pueden observar diversas operaciones interactuando y haciendo uso de los recursos. En un nivel de abstracción más bajo podemos encontrar que existen diferentes tipos de procesos de negocio. Según [22] los procesos de negocio pueden ser: estratégicos, operativos o de soporte. Los estratégicos son aquellos que despliegan las políticas y estrategias de una organización. Proporcionan directrices y límites de actuación, al resto de los procesos. Por ejemplo, una comunicación interna dentro de una organización. Los operativos son aquellos que están vinculados directamente al servicio que se presta. Por ejemplo, una inscripción de alumno. Los de soporte son aquellos que sirven de apoyo a los procesos operativos. Por ejemplo, la formación del personal. Podemos afirmar entonces que dentro de una organización existen muchos tipos de procesos de negocio en términos de cuál es su propósito o para que sirve.

Otra forma de agrupar los procesos de negocios es en términos de cómo son realizados desde el punto de vista formal. Es decir que pueden ser: formales o informales. Los formales son aquellos que son repetibles, que están bien estructurados, y hasta pueden ser automatizados. Por ejemplo, la creación de una cuenta bancaria. Los informales son aquellos que son flexibles, impredecibles (altamente variables), y difíciles de definir o repetir. Por ejemplo, desarrollar una estrategia de venta.

Podemos ver que dentro del universo de los procesos de negocios existen diferentes formas de agrupar a los mismos. Y podríamos seguir indagando y encontrando nuevas formas de agruparlos. Sin embargo, siguiendo el hilo conductor de la tesis, nos vamos a quedar con la última agrupación y vamos a concluir la presente sección con el siguiente párrafo:

"*Existe un grupo particular de procesos de negocio que son susceptibles de ser automatizados denominados formales. Los mismos, a partir de sus características naturales (repetibles, estructurados), se pueden hacer visibles mediante el uso de alguna herramienta para dicho propósito.*"

2.2 BPMN

Un enfoque actual utilizado en la gestión de los procesos de negocio es "Business Process Management" (BPM). Un aspecto clave del mismo es que permite representar visualmente una secuencia detallada de flujos de información y de actividades organizacionales necesarias para comprender un proceso de negocio.

En ésta sección vamos a mostrar cómo se puede representar un proceso de negocio en sí.

2.2.1 Definición de BPMN

"Business Process Model and Notation" (BPMN), es una notación para modelar procesos de negocio. Es uno de los estándares clave surgido desde el ámbito BPM [21]. Su propósito es modelar y representar un proceso de negocio. En los últimos años su nombre cambio un poco: Notación y Modelado de Procesos de Negocios, pero conserva la sigla BPMN. Permite realizar diagramas de procesos de negocio actuales para propósitos tales como la estandarización, la capacitación a empleados, y el control de calidad, entre otros.

2.2.2 Gestión de procesos de negocio dentro del entorno BPMN

Todas las organizaciones generalmente se encuentran tratando de *mejorar la manera en la que se hacen las cosas* dentro de las mismas. Este concepto se encuentra en el corazón de la gestión de procesos de negocio. Cuanto más tiempo una organización se enfoque en mejorar sus procesos de negocio, más repetibles y escalables serán sus operaciones y, en consecuencia, mejor será su desempeño en general [23] .

En la gestión de procesos de negocio la atención está puesta en la comprensión de los procesos de negocio en primer lugar. Luego, le idea es mejorarlos. Dentro del contexto de la comprensión de los procesos de negocio el desafío consiste en mostrar de alguna manera: el flujo del trabajo que se realiza, las tareas involucradas en dicho flujo, y los recursos necesarios que se utilizan en el desarrollo del proceso de negocio. La esencia de la comprensión es observar las operaciones en marcha de la organización y luego modelarlas.

BPMN proporciona un estándar para representar procesos de negocios tanto para propósitos descriptivos de alto nivel de abstracción, como niveles detallados y rigurosos de comprensión. En BPMN se utilizan distintos niveles de modelado:

- Mapas de procesos: simples diagramas de flujo de las actividades, sin más detalle que el nombre de las actividades y condiciones de decisión generales.
- Descripción de procesos: información más extensa acerca del proceso, como personas involucradas en llevarlo a cabo (roles), datos, información, etc.
- Modelos de proceso: diagramas de flujo detallados, con suficiente información como para poder analizar el proceso y simularlo. Esto permite ejecutarlo directamente o exportarlo a otras herramientas que pueden interpretar y ejecutar el modelo de proceso.

2.2.3 Notación para el modelado de procesos de negocio dentro del entorno BPMN

En ésta sección vamos a ver cuáles son los elementos principales que componen la notación para modelar procesos de negocio. Dentro de la notación BPMN, existen muchos detalles de implementación que se van a dejar de lado. Solo nos interesa lo esencial para tener un marco de referencia dentro del cual apoyarnos.

BPMN dispone de un conjunto de objetos que permiten describir todo el proceso de negocio. BPMN utiliza un conjunto de elementos gráficos especializados para describir un proceso de negocio y de qué manera es realizado (ver figura 2). Los elementos más relevantes de un proceso de negocio BPMN son los "*Objetos de Flujo*" (*Actividades, Eventos, Compuertas)* y el "*Flujo de secuencia*".

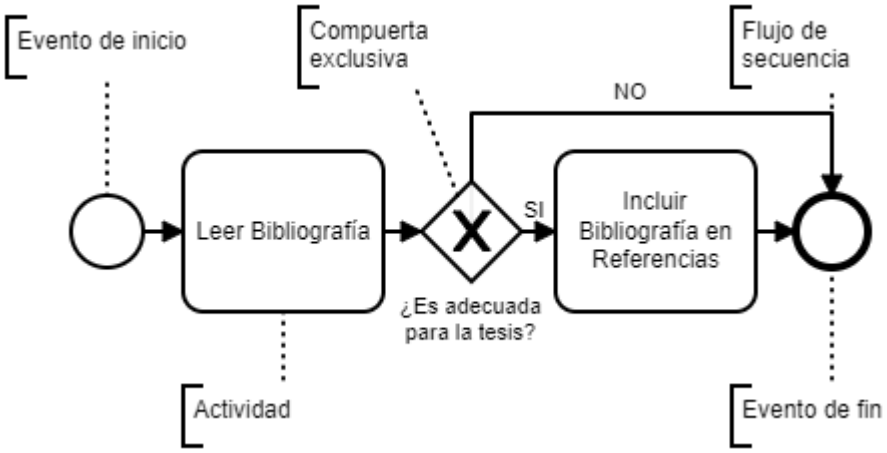


Figura 2 – Elementos BPMN

A continuación, describimos brevemente los elementos BPMN más relevantes.

2.2.3.1 Actividad

Una actividad representa algo realizado en un proceso de negocio. Tiene una forma rectangular con esquinas redondeadas. La mismas se toma cierto tiempo para ejecutarse. Involucra uno o más recursos de la organización. Requiere de algún tipo de entrada y produce algún tipo de salida. Puede ser atómica o compuesta. Cuando decimos atómica nos referimos a que la misma no puede ser subdividida en mayor detalle. Cuando hablamos de compuesta la misma puede ser subdividida en mayor detalle. La actividad atómica se conoce como tarea. La actividad compuesta se conoce como sub-proceso y el mismo es representado con un signo más en la parte inferior central de la forma (ver figura 3).

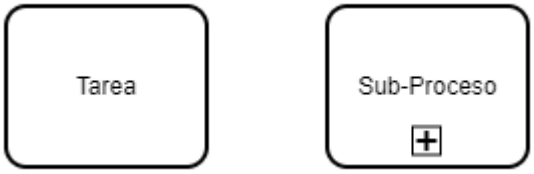


Figura 3 – Tarea y Sub-Proceso

A continuación, se describen algunos tipos de actividades que son de relevancia para el contexto de la tesis.

2.2.3.1.1 Tipos de Actividades

Actividad de usuario: Una actividad de usuario es ejecutada por una persona (usuario), pero el control lo lleva el sistema de "workflow" o "Process Engine". Luego de ejecutarse la tarea por el usuario, el sistema espera que por lo menos le comunique su estado, antes de poder continuar. Este tipo de actividad está inserto en lo que se denomina «Human Workflow Management».
Algunos ejemplos de "Human Workflow Management" son: "Revisar una factura", "Aprobar una solicitud de vacaciones", o "Administrar una solicitud de soporte" [19].

Actividad de servicio: Una actividad de servicio es una actividad automática que es ejecutada completamente por algún software. BPMN parte normalmente de la base, de que se trata de un servicio web, pero no es mandatorio. De todas formas, se trata de un componente de integración de aplicaciones, con lo cual tenemos por esta vía la entrada al puente con las arquitecturas orientadas al servicio (SOA).
Ejemplo de servicios de integración: "Solicitud de clasificación de riesgo crediticio a un sistema interbancario", "Verificación de stock de bodega para una orden de compra", "Disponibilidad de asiento para una reserva de pasajes" [19] (ver figura 4).

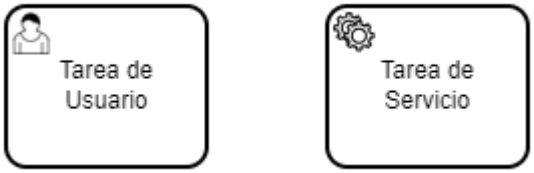


Figura 4 – Tarea de Usuario y Tarea de Servicio

Actividad manual: Una actividad manual es ejecutada por una persona, cuyo control no lo lleva un sistema de "workflow" o "Process Engine". Ejemplos: "Guardar un acta en un archivo físico", "Aclarar por teléfono una factura mal emitida" [19].

2.2.3.1.2 Sub-proceso

Dependiendo de la herramienta de modelado de proceso, la idea detrás del signo más es la de expandir el diagrama BPMN del sub-proceso. Es decir que al abrir el mismo, se puede ver en mayor detalle cómo se compone el diagrama BPMN correspondiente [21]. Un sub-proceso describe en su interior la lógica en detalle, pero en el **diagrama del proceso superior** no toma más lugar que una propia actividad [19]. En esencia, un sub-proceso es un diagrama BPMN comprimido que, si se expande puede mostrar en detalle cómo está compuesto (ver figura 5).

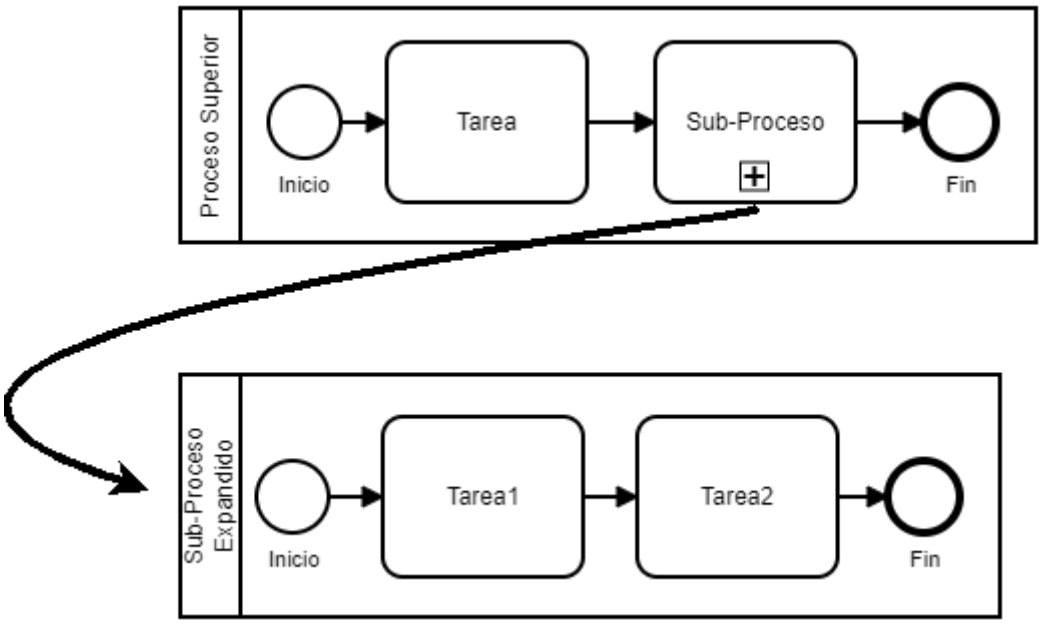


Figura 5 – Sub-Proceso comprimido y expandido

El proceso superior se inicia y nace un nuevo token. El token pasa por la tarea y llega al sub-proceso. Esto conlleva a que el proceso superior cree una instancia del sub-proceso. Dentro del sub-proceso se crea un nuevo token que sigue la lógica del flujo del sub-proceso desde el evento de inicio hasta el Evento que termina el sub-proceso. El token del evento superior espera el arribo del token del sub-proceso [19].

2.2.3.2 Eventos

Un evento es algo que "sucede" durante el curso de un proceso. Afecta al flujo del proceso. Puede iniciar, retrasar, interrumpir o finalizar el flujo del proceso. Se representa por círculos y existen tres tipos: "evento de inicio" representado por un circulo de línea fina única, "evento intermedio" representado por un circulo de línea fina doble, "evento de fin" representado por un circulo de línea gruesa única (ver figura 6).

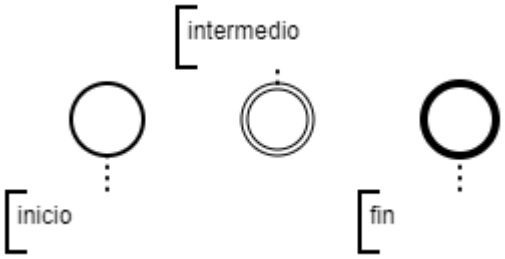


Figura 6 – Tipo de Eventos

Evento de inicio
Un evento de inicio muestra donde empieza un proceso. Es un pequeño círculo abierto, con una única línea fina como límite. Existen diferentes tipos de eventos de inicio para indicar diferentes circunstancias de inicio que se denominan disparadores. Los mismos se dividen entre básicos y avanzados.

Entre los eventos de inicio básicos están los siguientes:

- Simple: No se define ningún disparador. También es denominado genérico y representa el inicio manual del proceso por un usuario.
- Temporizador: El disparador es una fecha u hora específica.
- Mensaje: El disparador es un mensaje que llega desde otra entidad de negocio o rol (participante). Señal: El disparador es una señal difundida desde otro proceso.

Entre los eventos de inicio avanzados están los siguientes:

- Condicional: El disparador es una expresión de condición que debe ser satisfecha para que empiece el Proceso.
- Múltiple: Define uno o más disparadores que pueden ser cualquier combinación de mensajes, temporizadores, condiciones o señales.

Los eventos de inicio solo tienen flujos de secuencia de salida. Los eventos de inicio son donde el flujo de secuencia comienza.

Eventos de fin

Un evento de fin marca cuando un proceso finaliza. En este momento es posible que ocurra el lanzamiento de mensajes para otros procesos. Es un pequeño círculo abierto con una única línea gruesa marcando su límite. Existen diferentes tipos de eventos de fin que indican diferentes categorías de resultados para el proceso. Un resultado es algo que ocurre al final de un camino particular del proceso (ejemplo, un mensaje es enviado). Todos los eventos de fin son lanzadores de resultados.

Entre los eventos de fin básicos existen los siguientes:

- Básico: No se define ningún resultado.
- Mensaje: Comunicación con otra entidad de negocio (participante o proceso).
- Señal: Define un evento “broadcast”, que cualquier otro proceso puede ver y puede reaccionar como consecuencia.
- Terminador: Detiene todas las actividades del proceso, incluso si están en curso en otros caminos paralelos.

Entre los eventos de fin avanzados existen los siguientes:

- Error: Un estado final que interrumpirá el proceso o requerirá corrección.
- Cancelación: Usado junto con el sub-proceso de transacción, este evento causa la cancelación de este tipo de sub-procesos.
- Compensación: Este evento lanza el disparador para deshacer.
- Múltiple: Define dos o más resultados

2.2.3.3 Compuertas

Las compuertas son elementos de modelado que controlan cómo el proceso diverge o converge, es decir representan puntos de control para los caminos dentro de los procesos. Dividen y unifican el flujo del proceso. Tiene la forma de un rombo (ver figura 7).

Figura 7 – Compuerta BPMN

Una compuerta divide el flujo cuando este tiene múltiples flujos de secuencia salientes y unifica el flujo cuando este tiene múltiples flujos de secuencia entrantes.

Existen diferentes tipos de compuertas entre los cuales convenientemente mencionamos a los siguientes:

- Básica exclusiva: la compuerta envía el flujo por un solo camino saliente dependiendo de la evaluación de la condición del flujo de secuencia.
- Básica paralela: la compuerta envía un flujo a través de todos los caminos salientes, es decir en paralelo.

2.2.3.4 Canales (Swimlanes)

BPMN utiliza canales (swimlanes) para ayudar a dividir y organizar actividades en un diagrama.
Los tipos de canales que existen son los siguientes:

- Pool: actúa como contenedor para un proceso, cada uno representa un participante en un diagrama de procesos de negocio colaborativo.
- Lane (Carril): utilizado para representar un rol de negocio interno dentro de un proceso. Proveen un mecanismo para particionar los objetos dentro de un pool.

Un "participante" se define como un rol de negocio, por ejemplo, un comprador o un vendedor. Cada pool puede representar un solo participante. Los pools son representados como una caja rectangular que actúa como contenedor para los objetos de flujo, del proceso de un participante. Cada pool representa un proceso diferente y cada participante tiene su propio pool.
Existen un pool conocido como "caja negra", éstos nos muestran actividades o flujos de secuencia. Se utilizan cuando no se posee conocimiento del comportamiento del participante.

Los flujos de mensaje manejan todas las interacciones entre pools. Cuando un pool es una caja negra, el flujo de mensaje se conecta a su límite. Cuando un pool tiene elementos de proceso, el flujo de mensaje se conecta a estos elementos (ver figura 8).

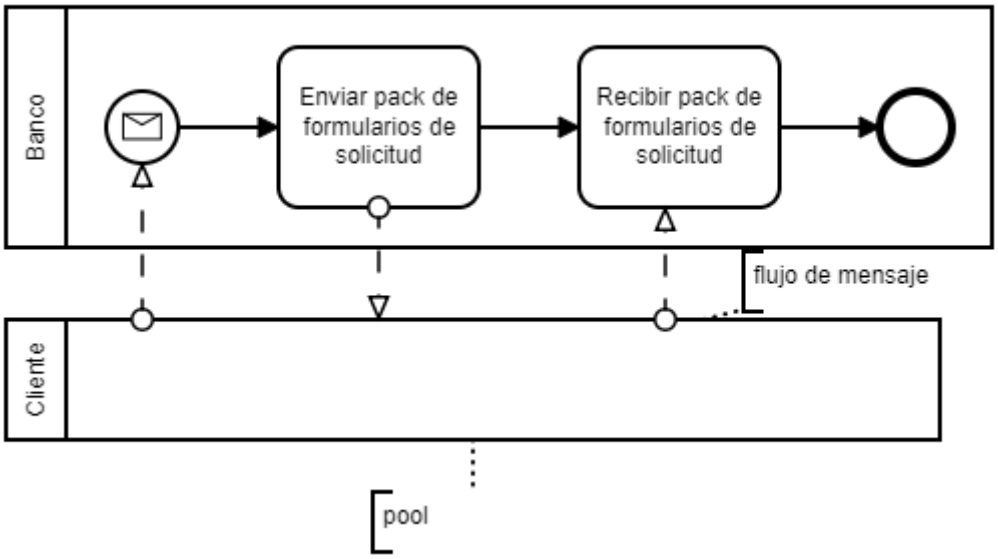


Figura 8 – Colaboración entre dos Pools

Los lanes o carriles crean sub-particiones para los objetos dentro de un pool. Estás particiones son utilizadas para agrupar elementos del proceso de negocio (mostrando cómo estos están relacionados), o qué roles tienen la responsabilidad de llevar a cabo las actividades (ver figura 9).

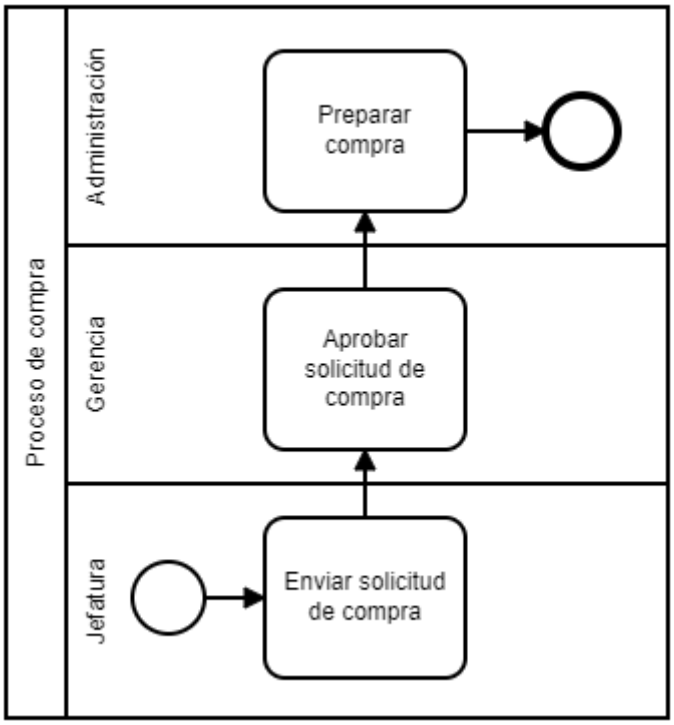


Figura 9 – Carriles dentro de un Pool

2.2.3.5 Artefactos

Los artefactos proporcionan un mecanismo para capturar información adicional sobre un proceso. Esta información no afecta directamente a las características del diagrama de flujo de proceso.

Existen tres tipos de artefactos:

- Objetos de datos: se utilizan para representar los documentos y datos que son manipulados por los procesos. Tienen la forma estándar de un documento, es decir un rectángulo con la esquina doblada. Suelen definir entradas y salidas de las actividades. Están ligados a la ejecución de las actividades.
- Grupos: proporcionan un mecanismo para resaltar y clasificar una sección del modelo o un conjunto de objetos. Un grupo es un rectángulo punteado y redondeado para rodear un grupo de objetos de flujo con el propósito de destacarlos. No añade restricciones adicionales. No afectan el flujo del proceso y no son parte de la descomposición de un proceso.
- Anotaciones de texto: Añaden información descriptiva a un modelo. Ayudan a la comprensión del mismo. proveen al modelador la posibilidad de añadir más notas o información descriptiva sobre un proceso o sus elementos. Las anotaciones de texto pueden conectarse a cualquier objeto en el diagrama o flotar libremente en cualquier parte de un diagrama (ver figura 10).

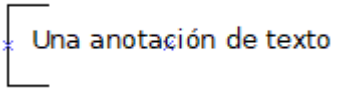


Figura 10 – Anotación de texto

2.2.3.6 Conectores

Los conectores vinculan dos objetos de un diagrama. Existen tres tipos de conectores.

Flujo de Secuencia
Un "flujo de secuencia" es el elemento que conecta los "objetos de flujo" (actividades, eventos y compuertas), Cada actividad puede tener uno o más flujos de secuencia de entrada y uno o más flujos de secuencia de salida. El "flujo del proceso" está determinado por todos los caminos de "flujo de secuencia". Cuando un flujo llega a una actividad, la actividad está lista para comenzar [21]. Además, el "flujo de secuencia" define el orden de los objetos de flujo en un proceso y crea los caminos del proceso que son navegados durante su ejecución.
El origen y destino de la línea del flujo de secuencia (el destino es manifestado por la punta de flecha en la línea) solo puede conectar eventos, actividades y compuertas.

Flujo de Mensaje
Define el flujo de comunicación entre dos participantes o entidades (por ejemplo, una organización y sus proveedores). El objeto de la comunicación es un mensaje. El flujo de mensajes define las comunicaciones entre dos participantes diferentes (representados como pools) del diagrama. El flujo de mensajes debe darse entre dos pools separados y no puede conectar dos objetos dentro de un mismo pool.

Asociaciones
Se utilizan para vincular artefactos (datos e información) con otros objetos del diagrama, incluyendo objetos de flujo (actividades, eventos y compuertas). Una asociación une un objeto del diagrama, es decir que crea una relación, con otro objeto del diagrama como por ejemplo artefactos o actividades (ver figura 11).

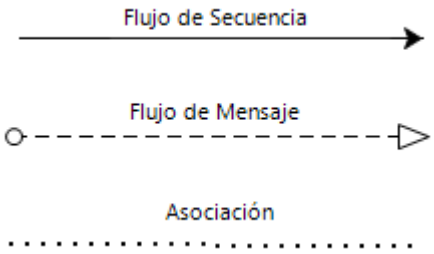


Figura 11 - Conectores

Flujo de Datos
El flujo de datos representa el movimiento de objetos de datos de entrada y de salida de las actividades. El flujo de datos está disociado del flujo de secuencia. El flujo de secuencia maneja el ordenamiento de las actividades. Las asociaciones manejan el flujo de datos hacia y desde las actividades.

Flujo Normal
El concepto de "flujo normal" es a menudo conocido como la "ruta feliz". Se refiere al proceso básico, donde los objetos de flujo (eventos, actividades y compuertas) están conectados a través del flujo de secuencia. Comenzando con un evento inicial y siguiendo las principales alternativas hasta que el proceso concluye en un evento final. El flujo normal describe la estructura básica del proceso o ***la ruta con más alta probabilidad de ocurrencia****. El centrarse en el flujo normal permite al modelador crear un diagrama de flujo relativamente más simple.* En la sección "3.5.1.4 – Proceso de Identificación del Flujo Normal" éste concepto va a ser referenciado.

Hasta este punto, se expuso una notación que permite hacer visible un proceso de negocio.

A continuación, presentamos una nueva sección vinculada al estudio del lenguaje natural y, su aplicación y vinculación con los diagramas BPMN.

2.3 NLP y BPMN

En ésta sección presentamos una de las bases desde la cual se construyó la presente tesis. En primer lugar, se presentan conceptos generales vinculados al campo de la lingüística. En segundo lugar, se presenta como el procesamiento del lenguaje natural (NLP) se aplica el campo mencionado. Y en tercer lugar se presenta como diversas investigaciones aplican NLP a BPMN.

2.3.1 Lingüística

La importancia del lenguaje natural en la vida de los seres humanos es un hecho que no debe pasarse por alto. El mismo nos permite el intercambio de pensamientos, la expresión de sentimientos y la preservación del conocimiento [24].
La lingüística se puede definir como el estudio científico del lenguaje humano. La lingüística se subdivide típicamente en dos campos la lingüística teórica y aplicada. La lingüística teórica se considera a menudo como el núcleo de la lingüística y se centra en la descripción de la naturaleza básica y la estructura del lenguaje. Por el contrario, la lingüística aplicada es un campo interdisciplinario que ofrece soluciones a problemas en el mundo real donde está involucrado el lenguaje natural.

La lingüística teórica se ocupa de la forma y estructura del lenguaje. Debido a que la tesis se refiere al lenguaje escrito solamente, en la misma vamos a mencionar aquellas ramas de la lingüística teórica que son relevantes para analizar y describir la forma escrita del lenguaje. Las ramas de la lingüística teórica se dividen en tres: Morfología, Sintaxis, Semántica.

Morfología
La Morfología se centra en el estudio de la estructura interna de las palabras y como la misma cambia en diferentes contextos. Para describir adecuadamente a la morfología, la misma introduce tres conceptos lingüísticos importantes: el lexema, el lema y la forma de la palabra (los cuales no son objetos de estudio de la tesis y por lo tanto solo se hace mención de los mismos).

Sintaxis
La Sintaxis estudia cómo las palabras se pueden combinar con frases, cláusulas y oraciones. Una oración puede consistir en una o más cláusulas, una cláusula consiste en una o más frases y una frase contiene una o más palabras. A nivel de palabra, podemos identificar diferentes categorías léxicas, o partes del discurso. Las partes principales del discurso incluyen sustantivos (N), verbos (V), adjetivos (Adj), adverbios (Adv), preposiciones (Prep) y determinantes (Det). Por ejemplo, una oración se puede describir de la siguiente manera:

[Det Los] [N estudiantes] [V leen] [Det un] [N libro]

Además de las partes del discurso, podemos asignar funciones gramaticales a las palabras. En general, las oraciones contienen un sujeto y un predicado. En el ejemplo precedente el sujeto está dado por “los estudiantes” y el predicado está dado por “leen un libro”. Además, las oraciones se pueden enriquecer incorporando objetos y adverbios. Los adverbios proporcionan más detalles sobre el tiempo y el lugar. A partir del conocimiento sobre las categorías léxicas y las funciones gramaticales, es posible investigar cómo se construyen las oraciones, las cláusulas y las frases.

Semántica
La Semántica hace referencia al estudio del significado de las palabras. El estudio lingüístico del significado se centra en el significado de las palabras (semántica léxica). Las palabras pueden relacionarse semánticamente con otras palabras de varias maneras. Dado que tales relaciones sensoriales son un concepto importante de la semántica léxica y una cuestión predominante en el contexto del análisis del lenguaje natural, mencionamos que las relaciones sensoriales más importantes son las relaciones de sinonimia, homonimia, hipernomía y meronimia.

2.3.2 Procesamiento del lenguaje natural

El "Natural Language Processing - Procesamiento del Lenguaje Natural" (NLP) *es un área de investigación interdisciplinaria que se ocupa del análisis automático del lenguaje natural*. Los fundamentos de NLP se pueden encontrar en diversas disciplinas, incluidas la informática, la lingüística, la inteligencia artificial y la psicología, por ejemplo. NLP se aplica en varios dominios, como el procesamiento del texto en el lenguaje natural, la traducción automática de textos y el reconocimiento de la voz. A continuación, presentamos los conceptos principales que son usados para el análisis del texto en lenguaje natural.

Cuerpos de texto
Los cuerpos de texto o "Text Corpora" son una gran colección de textos (millones de palabras). Por lo general, se codifican electrónicamente de modo que puedan procesarse automáticamente y utilizarse para análisis estadísticos. Los cuerpos de texto representan una fuente valiosa para análisis gramatical a gran escala y pueden, por ejemplo, proporcionar información sobre la frecuencia de las palabras o la probabilidad de una parte particular de la secuencia del habla (por ejemplo, con qué frecuencia ocurre un sustantivo después de un verbo). Por lo tanto, los cuerpos de texto desempeñan un papel importante para el desarrollo de herramientas automáticas de procesamiento de lenguaje, como los etiquetadores y analizadores.

Un cuerpo de texto utilizado para el español es el "AnCora-ES" [25]. El mismo es un corpus del español con diferentes niveles de anotación:

- lema y categoría morfológica
- constituyentes y funciones sintácticas
- estructura argumental y papeles temáticos
- clase semántica verbal
- tipo denotativo de los nombres deverbales
- sentidos de WordNet nominales
- entidades nombradas
- relaciones de correferencia

Contiene 500.000 palabras y está constituido fundamentalmente por textos periodísticos.

Para permitir el procesamiento de textos de manera automatizada, los cuerpos de texto se basan en un conjunto de etiquetas, es decir, un conjunto de códigos que se utilizan para referirse a las diferentes partes del habla. El "Penn TreeBank" (ver tabla 1) es una colección de corpus de diferentes idiomas, que se emplea con frecuencia para entrenar analizadores y etiquetadores.

Etiqueta	Descripción	Ejemplo
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	there is
FW	foreign word	les
IN	preposition, subordinating conjunction	in, of, like
IN/that	that as subordinator	that
JJ	adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest
LS	list marker	1)
MD	modal	could, will
NN	noun, singular or mass	table
NNS	noun plural	tables
NP	proper noun, singular	John
NPS	proper noun, plural	Vikings
PDT	predeterminer	both the boys
POS	possessive ending	friend’s
PP	personal pronoun	I, he, it
PPZ	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
SENT	Sentence-break punctuation	. ! ?
SYM	Symbol	/ [= *
TO	infinitive ‘to’	togo
UH	interjection	uhhuhhuhh
VB	verb be, base form	be
VBD	verb be, past tense	was, were
VBG	verb be, gerund/present participle	being
VBN	verb be, past participle	been
VBP	verb be, sing. present, non-3d	am, are
VBZ	verb be, 3rd person sing. present	is
VH	verb have, base form	have
VHD	verb have, past tense	had
VHG	verb have, gerund/present participle	having
VHN	verb have, past participle	had
VHP	verb have, sing. present, non-3d	have
VHZ	verb have, 3rd person sing. present	has
VV	verb, base form	take
VVD	verb, past tense	took
VVG	verb, gerund/present participle	taking
VVN	verb, past participle	taken
VVP	verb, sing. present, non-3d	take
VVZ	verb, 3rd person sing. present	takes
WDT	wh-determiner	which
WP	wh-pronoun	who, what

WP$	possessive wh-pronoun	whose
WRB	wh-abverb	where, when
#	#	#
$	$	$
“	Quotation marks	‘ “
``	Opening quotation marks	‘ “
(	Opening brackets	({
)	Closing brackets	) }
,	Comma	,
:	Punctuation	– ; : — …

Tabla 1 - Penn Treebank

Generalmente el conjunto de etiquetas Penn TreeBank es utilizado para codificar partes del discurso. Por ejemplo, la etiqueta VB denota la forma base de un verbo, mientras que VBD y VBN indican la forma de participio simple y pasado de un verbo. En consecuencia, las etiquetas que comienzan con NN denotan sustantivos, las etiquetas que comienzan con JJ indican adjetivos y las etiquetas que comienzan con RB representan adverbios.

Para el lenguaje español, dentro de Stanford CoreNLP, existe un conjunto de etiquetas similar a “Penn TreeBank” denominado “DEFT Spanish TreeBank” (ver tabla 2). El mismo se muestra a continuación.

Etiqueta	**Descripción**	**Ejemplo(s)**
Adjetivos		
`ao0000`	Adjective (ordinal)	*primera*, *segundo*, *últimos*
`aq0000`	Adjective (descriptive)	*populares*, *elegido*, *emocionada*, *andaluz*
Conjunciones		
`cc`	Conjunction (coordinating)	*y*, *o*, *pero*
`cs`	Conjunction (subordinating)	*que*, *como*, *mientras*
Determiners		
`da0000`	Article (definite)	*el*, *la*, *los*, *las*
`dd0000`	Demonstrative	*este*, *esta*, *esos*
`de0000`	"Exclamative" (TODO)	*qué* (*¡Qué pobre!*)
`di0000`	Article (indefinite)	*un*, *muchos*, *todos*, *otros*
`dn0000`	Numeral	*tres*, *doscientas*
`do0000`	Numeral (ordinal)	*el **65** aniversario*
`dp0000`	Possessive	*sus*, *mi*
`dt0000`	Interrogative	*cuántos*, *qué*, *cuál*
Puntuación		
`f0`	Other	*&*, *@*
`faa`	Inverted exclamation mark	*¡*
`fat`	Exclamation mark	*!*
`fc`	Comma	*,*
`fca`	Left bracket	*[*
`fct`	Right bracket	*]*
`fd`	Colon	*:*
`fe`	Double quote	*"*
`fg`	Hyphen	*-*
`fh`	Forward slash	*/*

Etiqueta	Descripción	Ejemplo(s)
fia	Inverted question mark	*¿*
fit	Question mark	*?*
fp	Period / full-stop	*.*
fpa	Left parenthesis	*(*
fpt	Right parenthesis	*)*
fra	Left guillemet / angle quote	*«*
frc	Right guillemet / angle quote	*»*
fs	Ellipsis	*...*, *etcétera*
ft	Percent sign	*%*
fx	Semicolon	*;*
fz	Single quote	*'*
Interjecciones		
i	Interjection	*ay*, *ojalá*, *hola*
Sustantivos		
nc00000	Unknown common noun (neologism, loanword)	*minidisc*, *hooligans*, *re-flotamiento*
nc0n000	Common noun (invariant number)	*hipótesis*, *campus*, *golf*
nc0p000	Common noun (plural)	*años*, *elecciones*
nc0s000	Common noun (singular)	*lista*, *hotel*, *partido*
np00000	Proper noun	*Málaga*, *Parlamento*, *UFINSA*
Pronombres		
p0000000	Impersonal *se*	*se*
pd000000	Demonstrative pronoun	*éste*, *eso*, *aquellas*
pe000000	"Exclamative" pronoun	*qué*
pi000000	Indefinite pronoun	*muchos*, *uno*, *tanto*, *nadie*
pn000000	Numeral pronoun	*dos miles*, *ambos*
pp000000	Personal pronoun	*ellos*, *lo*, *la*, *nos*
pr000000	Relative pronoun	*que*, *quien*, *donde*, *cuales*
pt000000	Interrogative pronoun	*cómo*, *cuánto*, *qué*
px000000	Possessive pronoun	*tuyo*, *nuestra*
Adverbios		
rg	Adverb (general)	*siempre*, *más*, *personalmente*
rn	Adverb (negating)	*no*
Preposiciones		
sp000	Preposition	*en*, *de*, *entre*
Verbos		
va00000	Verb (unknown)	*should*
vag0000	Verb (auxiliary, gerund)	*habiendo*
vaic000	Verb (auxiliary, indicative, conditional)	*habría*, *habríamos*
vaif000	Verb (auxiliary, indicative, future)	*habrá*, *habremos*
vaii000	Verb (auxiliary, indicative, imperfect)	*había*, *habíamos*
vaip000	Verb (auxiliary, indicative, present)	*ha*, *hemos*
vais000	Verb (auxiliary, indicative, preterite)	*hubo*, *hubimos*
vam0000	Verb (auxiliary, imperative)	*haya*
van0000	Verb (auxiliary, infinitive)	*haber*
vap0000	Verb (auxiliary, participle)	*habido*

Etiqueta	Descripción	Ejemplo(s)
vasi000	Verb (auxiliary, subjunctive, imperfect)	*hubiera*, *hubiéramos*, *hubiese*
vasp000	Verb (auxiliary, subjunctive, present)	*haya*, *hayamos*
vmg0000	Verb (main, gerund)	*dando*, *trabajando*
vmic000	Verb (main, indicative, conditional)	*daría*, *trabajaríamos*
vmif000	Verb (main, indicative, future)	*dará*, *trabajaremos*
vmii000	Verb (main, indicative, imperfect)	*daba*, *trabajábamos*
vmip000	Verb (main, indicative, present)	*da*, *trabajamos*
vmis000	Verb (main, indicative, preterite)	*dio*, *trabajamos*
vmm0000	Verb (main, imperative)	*da*, *dé*, *trabaja*, *trabajes*, *trabajemos*
vmn0000	Verb (main, infinitive)	*dar*, *trabjar*
vmp0000	Verb (main, participle)	*dado*, *trabajado*
vmsi000	Verb (main, subjunctive, imperfect)	*diera*, *diese*, *trabajáramos*, *trabajésemos*
vmsp000	Verb (main, subjunctive, present)	*dé*, *trabajemos*
vsg0000	Verb (semiauxiliary, gerund)	*siendo*
vsic000	Verb (semiauxiliary, indicative, conditional)	*sería*, *serían*
vsif000	Verb (semiauxiliary, indicative, future)	*será*, *seremos*
vsii000	Verb (semiauxiliary, indicative, imperfect)	*era*, *éramos*
vsip000	Verb (semiauxiliary, indicative, present)	*es*, *son*
vsis000	Verb (semiauxiliary, indicative, preterite)	*fue*, *fuiste*
vsm0000	Verb (semiauxiliary, imperative)	*sea*, *sé*
vsn0000	Verb (semiauxiliary, infinitive)	*ser*
vsp0000	Verb (semiauxiliary, participle)	*sido*
vssf000	Verb (semiauxiliary, subjunctive, future)	*fuere*
vssi000	Verb (semiauxiliary, subjunctive, imperfect)	*fuera*, *fuese*, *fuéramos*
vssp000	Verb (semiauxiliary, subjunctive, present)	*sea*, *seamos*

Tabla 2 – DEFT Spanish TreeBank de Stanford CoreNLP

Etiquetado de parte del discurso

El etiquetado de parte del discurso (o etiquetado) es el proceso de asignar una parte del discurso a cada palabra en un texto dado. La entrada para un algoritmo de etiquetado es un conjunto de cadenas y un conjunto de etiquetas. La salida del algoritmo es una etiqueta única para cada palabra y para cada signo de puntuación. Para ilustrar el concepto de etiquetado de parte del discurso, considere la siguiente salida de un etiquetador de parte del discurso para la siguiente oración: “Process the customer data”.

Process/VB the/DT customer/NN data/NNS

Para una persona, es bastante evidente que se trata de una oración imperativa que le indica a otra que procese los datos del cliente. Sin embargo, el desafío es asignar la etiqueta correcta de parte del discurso a la palabra ambigua “process”. Dicha palabra podría ser un verbo (como “to process the data”) o un sustantivo (como “the process is efficient”). El desafío general del etiquetado de parte del discurso es resolver tales ambigüedades. En general, los algoritmos de etiquetado de parte del discurso se pueden subdividir en dos categorías principales: enfoques basados en reglas y enfoques estocásticos.

Análisis del lenguaje natural

El análisis de lenguaje natural tiene como objetivo determinar la estructura gramatical de una oración (representados cómo árboles de estructura). El mismo también adolece de la naturaleza ambigua del

lenguaje y la resolución de tales ambigüedades no es una tarea sencilla (a veces ni siquiera para los humanos). Para resolver las ambigüedades estructurales de los árboles, muchos analizadores se basan en modelos probabilísticos de sintaxis. Sobre la base de tales modelos, es posible asignar probabilidades para analizar árboles y, en consecuencia, decidir qué árbol de análisis es más apropiado para una oración. El modelo de sintaxis más empleado en este contexto es la gramática probabilística libre de contexto (PCFG). Si una oración dada es estructuralmente ambigua, las probabilidades le permiten al analizador seleccionar la opción más probable. La variedad de analizadores basados en PCFG es enorme. Entre los ejemplos más destacados se incluyen el Collins 'Parser, el Berkeley Parser, el Charniak & Johnson's Parser y el **Stanford Parser**.

WordNet
Es una base de datos léxica del Idioma inglés (ampliada además a otros idiomas como el español) que agrupa palabras en inglés en conjuntos de sinónimos llamados synsets, proporcionando definiciones cortas y generales y almacenando las relaciones semánticas entre los conjuntos de sinónimos [26]. Su propósito es doble: producir una combinación de diccionario y tesauro cuyo uso sea más intuitivo, y soportar los análisis automáticos de texto y las aplicaciones de inteligencia artificial. Así, WordNet es el lexicón computacional de inglés comúnmente más usado para desambiguar el significado de las palabras (word sense disambiguation (WSD)), una tarea que tiene como objetivo asignar el concepto más apropiado (i.e. synsets) a los términos en contexto. En el contexto de WordNet, las relaciones entre synsets se denominan relaciones semánticas y las relaciones entre palabras se denominan relaciones léxicas. La distinción entre ambos es necesaria ya que existen algunas relaciones entre synsets mientras que otras existen únicamente entre palabras. En general, WordNet está disponible gratuitamente en línea y se puede descargar del sitio web de la Universidad de Princeton. Una de las posibilidades más fáciles de acceder a WordNet es el uso de la interfaz en línea en la página de inicio de WordNet. Para incorporar WordNet en prototipos de desarrollo propio, Existe un gran conjunto de interfaces de aplicación (API) para una variedad de lenguajes de programación diferentes.

2.3.3 Procesamiento del lenguaje natural y modelos de proceso

En la actualidad existe una gran disponibilidad de herramientas NLP como etiquetadores, analizadores y bases de datos léxica. El procesamiento del lenguaje natural se aplica en una multitud de dominios. Dado que el campo de los modelos de procesos de negocio contiene mucha información en lenguaje natural, el mismo también califica como un campo de aplicación de NLP.
Los modelos de proceso comparten muchos puntos en común con los modelos conceptuales en general. En general, la aplicación de técnicas NLP en el modelado conceptual a menudo apunta a mejorar el diseño de dichos modelos. En particular, la literatura existente sobre la aplicación de técnica NLP revela dos direcciones principales que se han considerado para lograr este objetivo: enfoques que aplican NLP en material de texto externo para inferir información útil sobre el diseño del modelo, y enfoques que aplican NLP en el texto de los elementos del modelo para facilitar diferentes tipos de análisis.

Muchos de los enfoques que aplican técnicas de NLP sobre material de texto externo apuntan a la inferencia automática de modelos conceptuales. Los ejemplos incluyen la creación automática (o semi-automática) de modelos de proceso [8], [27], [28], [29], diagramas de dependencia conceptual [30], modelos de entidad – relación [31] y diagramas UML [32], [33], [34], [35]. Lo que todos estos trabajos tienen en común es la aplicación de herramientas estándar NLP como, etiquetadores, analizadores o reconocimientos de partes del discurso para inferir información relevante de fuentes de texto escritas o verbales. Debido a que tales fuentes generalmente contienen oraciones gramaticalmente ricas, los autores obtienen resultados satisfactorios con esta estrategia.

Las técnicas que aplican las herramientas NLP sobre un modelo conceptual en sí mismo deben cumplir diferentes requisitos. Típicamente, los modelos conceptuales no contienen oraciones completas y gramaticalmente correctas. Como resultado, los autores evitan el uso de herramientas tradicionales NLP como analizadores y etiquetadores. Algunos autores recomiendan explícitamente evitar la aplicación de herramientas NLP debido a estos problemas [36], [37]. Sin embargo, hay varios intentos de hacer uso del lenguaje natural dentro de la producción de modelos conceptuales. Si bien muchos trabajos que aplican NLP sobre modelos conceptuales se refieren a la calidad del modelo, también hay enfoques que apuntan a descubrir conocimiento que está implícitamente capturado por dichos modelos. Los ejemplos incluyen la identificación de correspondencias de actividades entre modelos [38], [39], el descubrimiento de servicios [40] y la elicitación (obtención) de patrones de proceso [41].

Las herramientas NLP para obtener la estructura sintáctica de las oraciones, como el analizador sintáctico y los etiquetadores, solo se emplean en enfoques que funcionan con material de texto. Las técnicas que aplican las herramientas NLP sobre un modelo conceptual no hacen uso de analizadores o etiquetadores. Si suponen cierto formato del lenguaje en el modelo [[42], [43], [44]] o ignoran la sintaxis. El último enfoque en general elimina las palabras de detención, como artículos y conjunciones, y usa las palabras restantes para sus análisis. Si bien esto puede ser suficiente para algunos casos de uso, inhibe el uso preciso del lenguaje natural en los modelos conceptuales.

Por ejemplo, considere las actividades del modelo de proceso “Crear factura para el cliente” y “Creación de factura para el cliente”. Aparentemente, ambas actividades transmiten el mismo significado. Sin embargo, la forma sintáctica es diferente. Si bien el texto de la primera contiene un verbo imperativo al principio, el texto de la segunda etiqueta no contiene ningún verbo. Sin embargo, ambas actividades dan instrucciones para crear una factura. Simplemente observamos diferentes representaciones gramaticales del mismo significado. Este ejemplo ilustra que la estructura sintáctica de un texto o una oración, así como el papel de cada palabra, es una valiosa fuente de información. Un análisis sofisticado de los modelos de proceso de negocio solo puede lograrse si somos capaces de abordar el problema de reconocer la estructura sintáctica de las etiquetas de los elementos del modelo de proceso de negocio. La siguiente tabla (ver tabla 3) muestra una vista rápida de algunos enfoques que aplican NLP, sacado desde [24].

Enfoque	**Autor**	**Herramienta NLP**
Construcción de Modelos		
Modelo BPMN desde Textos	Friedrich	Parser, WordNet
Modelo BPMN desde Grupos de Historias	Goncalves	Parser
Modelo BPMN desde Fuentes Textuales	Ghose	Parser
Modelo BPMN desde Casos de Uso	Sinha	Parser
Diagramas de Dependencia desde Textos	Gangopadhyay	Parser
Modelo-ER desde Textos	Gomez	Parser
Modelo-ER desde Textos	Omar	Tagger, Parser
Modelo de Clases UML desde Textos	Bajwa & Choudhary	Tagger, Parser
Modelo UML desde Textos	More & Phalnikar	Tagger, Parser
Modelo UML desde Textos	Deeptimahanti & Babar	Parser, WordNet
Modelo Conceptual desde Requerimientos	Fliedl	Tagger, Parser
Modelo Conceptual desde Requerimientos	Montes	Tagger, Parser
Soporte de Diseño		
Visualización de Descripciones de Casos de Uso	Richards	Parser
Consistencia de Modelos de Objetos	Bolloju	Parser
Reconocimiento de Voz para UML	Lahtinen & Peltonen	Speech Analysis
Construcción de Especificaciones Formales		

Declaraciones SQL	Tseng and Chen	Parser
Algebra Relacional	Tseng	Parser
Aseguramiento de la Calidad		
Detección de Inconsistencia de Términos en PMs	Koschmider & Blanchard	WordNet
Soporte de Usuario para PMs	Koschmider	WordNet
Verificación de la Consistencia Lingüística para CMs	van der Vos	WordNet
Aplicación de Convenciones de Nomenclatura en PMs	Becker	Domain Thesauri
Reducción de Variaciones Lungüisticas en PMs	Breuker	WordNet
Detección de Errores Semánticos en EPCs	Gruhn and Laue	Propietary
Anotación Semántica para EPCs	Bögl	Propietary
Generación de Texto		
Generación desde Diagramas de Clase UML	Meziane	WorNet
Generación desde Modelos de Objetos	Lavoie	Propietary
Generación desde Modelos de Datos	Verheijen & van Bekkum	Propietary
Generación desde Modelos de Datos	Nijssen & Halpin	Propietary
Generación desde Modelos Conceptuales	Dalianis	Propietary
Información de Elicitación		
Descubrimiento de Servicios desde CMs	Knackstedt	WordNet
Detección de Patrones de Proceso	Gacitua-Decar & Pahl	WordNet
Medición de Similitud en PMs	Ehrig	WordNet
Mapeo de Actividad del Proceso	Becker	Propietary

Tabla 3 - Enfoques que aplican NLP

2.3.4 Oraciones en lenguaje natural, BPMN y NLP

Gran parte de los procesos de negocios, dentro de las organizaciones, están descriptos en lenguaje natural. Es decir que están redactados o descriptos con oraciones en lenguaje natural. Por otro lado, recordemos que NLP es un conjunto de técnicas y herramientas que se aplican al estudio y a la investigación del lenguaje natural. Del campo de investigación mencionado precedentemente surgieron diversos trabajos que se ocuparon del mismo [6], [7], [9], [11], [45], [46] (ver tabla 4, la misma se presenta en [7]).

Categorías
Extracción de Modelos de Proceso desde Textos
Generación de Modelos de Proceso desde textos
Friedrich
Chueng
Minería de Procesos desde Textos en Lenguaje Natural
Santoro
Jiexun
Goncalves
Generación de Textos desde Modelos de Proceso
Texto Generado desde Modelos de Proceso
Leopold
Meitz
Inconsistencias entre Modelos de Proceso y Textos
van der Aa
Estructuración de texto
Heinonen
Hearst
Hynes and Bexley

Tabla 4 – Identificación de Elementos de Proceso en Textos en Lenguaje Natural

Dichos trabajos coinciden en que es posible utilizar técnicas y herramientas NLP para producir modelos de procesos de negocios basados en el procesamiento del lenguaje natural. *Todos convergen en utilizar NLP para analizar metódicamente tres aspectos del lenguaje natural* o de las oraciones en sí: el *análisis sintáctico*, que es la determinación de un árbol de sintaxis y las relaciones gramaticales entre las partes de una oración; *análisis semántico*, que es la extracción del significado de palabras o frases; y la *resolución de anáforas* (referencia a un término o a una parte del discurso), que implica la identificación de los conceptos que son referenciados usando pronombres dentro de una oración [7]. Es difícil y complejo procesar el lenguaje natural. Por este motivo existen diferentes técnicas y herramientas vinculadas a NLP que benefician y se usan para el procesamiento del lenguaje natural:

- Tecnicas: Tokenization, Parsing, Chunking, Part-Of-Speech (POS)
- Herramientas: GATE, CRF++, Ling-pipe, Stanford POS tagger, Stanford parser, FrameNet, WordNet, CoreNLP, NLTK tagger, Bag-of-words, VerbNet.

Dado que *es fácil para las personas definir procesos de negocio con oraciones escritas en lenguaje natural,* NLP se utiliza en el contexto de BPMN para extraer procesos de negocios desde el lenguaje natural. Dentro del marco de NLP y BPMN se realizó una investigación que propone un enfoque semiautomático para identificar elementos de proceso en textos en lenguaje natural (descripciones de procesos) [7]. Es decir que, con base en el estudio del procesamiento del lenguaje natural, éste trabajo presentó un conjunto de reglas de mapeo para identificar elementos de proceso dentro de textos en lenguaje natural. A continuación, se hace una pequeña referencia al mismo.

2.3.4.1 Enfoque semiautomático para identificar elementos de proceso en textos en lenguaje natural

El enfoque está compuesto por cuatro pasos principales: datos de entrada, análisis sintáctico de texto, análisis de lógica de texto y salida. El propósito de dicha investigación fue proponer un enfoque que permitiera *identificar elementos de procesos de negocio a partir del análisis de textos en lenguaje natural*. Se apoyó en NLP y el principal componente que tomamos desde dicha investigación para la

presente tesis fue un "*conjunto de reglas de mapeo y correlaciones de palabras para identificar elementos de proceso en textos de lenguaje natural"*.

2.3.4.2 Conjunto de reglas de mapeo para identificar elementos BPMN en lenguaje natural

Con el fin de apoyar y minimizar el esfuerzo del analista de procesos en el paso de modelado, el enfoque propone un *conjunto de reglas de mapeo y correlaciones de palabras* para identificar elementos de proceso en textos de lenguaje natural. Las reglas se originan en un conjunto diverso de clases gramaticales (verbo, pronombre, artículo, sustantivo, entre otras). Basado en el estudio de las clases gramaticales, no existe un patrón que describa la forma en que las clases gramaticales deberían presentarse en un texto. Esto muestra que las mismas están relacionadas entre sí y representan algunos elementos de proceso. Por ejemplo, *las oraciones que contienen el sujeto, el verbo y el objeto representan un elemento del proceso, como, por ejemplo, una tarea manual* y no hay dependencia del orden en el que aparecen en el texto.

Las reglas de mapeo se definieron manualmente, y cada regla se clasifica de acuerdo con una categoría del conjunto básico de elementos de modelado BPMN. Las reglas de mapeo se aplican sobre oraciones en lenguaje natural para identificar elementos de proceso de negocio. Por último y por ello no menos importante se hace mención a las siguientes restricciones sobre el texto a analizarse: *los textos* para la generación de modelos de proceso *deben ser gramaticalmente correctos*, es decir, *es necesario eliminar y corregir manualmente palabras u oraciones que son gramaticalmente incorrectas*, de modo que el texto se vuelva gramaticalmente correcto. Además, *el texto no debe contener preguntas y debe describirse de forma secuencial*.

Las características presentadas en el texto secuencial tienen similitudes con los modelos de procesos de negocio. Hay palabras clave comunes en textos secuenciales, como primero, segundo, cerca, luego, finalmente, siguiente, ahora, después, entre otros. Estas palabras muestran posibles relaciones (correlaciones) con elementos de modelado de BPMN como actividades, gateways, pools, swimlanes, etc.

A continuación, se muestran un conjunto de tablas (ver tablas 5 - 9) que representan el conjunto de reglas de mapeo propuestas.

Activities – primary rules		
Rules	**Description**	**Sentence example**
Rule 1	<subject>+ <verb>+ <object>	The Support Officer
Rule 2	<subject>+ <aux>+ <verb>+ <object><in the future)	The secretary <subject>will <aux>send<verb>to dispatch <object.>
Rule 3	<verb>+<article>+ <object>	- choose <verb>a <article>document <object>. - it do <verb>a<article>order <object>.
Rule 4	<subject>+<verb>+ <object>+ <conjunction>+ <verb>+ <object>	A client <subject>calls <verb>the help desk <object>and <conjunction >makes <verb>a request <object>.
Rule 5	<object>+<subject>+ <verb>	The severity <object>of the claimant <subject>is evaluated <verb>.
Rule 6	<subject (occult)>+ <verb>+ <conjunction>+<verb>+ <object>	The first activity is to check <verb>and <conjunction>repair <verb>the hardware <object>.

Tabla 5 – Reglas para la identificación de actividades primarias

Events – primary rules		
Rules	**Description**	**Sentence example**
Rule 1	<subject>+ <verb>+ <object>	After the agent <subject>has confirmed <verb>the claim <object>to the clerk.
Rule 2	<subject>+<verb>+ <agent >+<object>	The SCT physical <subject>file was stored <verb in the past>by <agent>the Back office <object>. (passive voice)
Rule 3	<object>+ <verb present perfect>	. . . Urgent document <object>has been received <verb>by the Manager. . . .
Rule 4	<object>+<verb past> + <subject>	. . . a message <object>was generated <verb>to the customer <subject>.

Tabla 6 – Reglas para la identificación de eventos primarios

Exclusive gateways (XOR) – primary rules		
Rules	**Description**	**Sentence example**
Rule 1	<verb>+ <signal word>+ <subject>+ <object>	It first checked <verb>whether <signal word>the claimant <subject>is insured <object>by the organization.
Rule 2	<signal word>+ <condition>+<task/event>+<alternative signalword>+ <task/event>	If <signal word>the claimant requires two or more forms <condition>, the Department of customer select the forms <task>. Otherwise <alternative signal word>, Department of customer it requires documentation <task>.
Rule 3	<task/event>+ <signal word>+<condition>	After that they enter into a firm commitment to buy the stock and then offer it to the public <task>, when <signal word>they haven't still found any reason not to do it <condition>.
Rule 4	<task>+ <signal word>+ <condition>+<alternative signal word>+<task>	The clerk checks <task>whether<signal word>the beneficiary's policy was valid at the time of the accident <condition>. If not <alternative signal word>, it send to Department of the intelligence <task>.

Tabla 7 – Reglas para la identificación de compuertas exclusivas primarias (XOR)

Parallel Gateway (AND) – primary rules		
Rules	**Description**	**Sentence example**
Rule 1	<task/event>+<signal word>+<task/ event>	Forward the document <task>, In parallel with this <signal word>, the CC shall also notify the Executive Board <task>.
Rule 2	<signal word>+<task>+ <conjunction>+<task>+<task>	In parallel with this <signal word>Department of sell send the document <task>and <conjunction>notify the department of engineering <task>. Then, the document is processed <task>.
Rule 3	<signal word>+<task/event>	In the meantime <signal word>, the engineering department prepares everything for the assembling of the ordered bicycle <task>.

Tabla 8 – Reglas para la identificación de compuertas paralelas primarias (AND)

Swimlanes		
Rules	**Description**	**Sentence example**
Rule 1	The subject of the sentence.	<subject>perform <task/event >
Rule 2	<task >+ <indirect object >	She then submits an order <task> to the customer <indirect object>.
Rule 3	<event >+ <indirect object>	The Manager forwarded the form <event> to Official <indirect object>.

Tabla 9 – Reglas para la identificación de swimlanes

Para mayor información sobre el enfoque mencionado remitirse a la referencia [7].

Hasta aquí se presentó cómo NLP se aplica a BPMN y una investigación que ejemplifica lo mencionado.

2.4 Desarrollo Dirigido por Mockups

En ésta sección presentamos el artefacto Mockup. Y a continuación, exponemos el enfoque denominado MockupDD. El mismo toma como base el Mockup y a partir del mismo propone un proceso de desarrollo ágil. Por último, presentamos una técnica denominada “End User Grammar – Gramática de usuario Final” (EUG). La misma también se apoya sobre el artefacto Mockup. EUG se desprende de MockupDD y propone realizar anotaciones amigables para el usuario final sobre los Mockups.

2.4.1 Mockup

Un Mockup es un prototipo de interfaz de usuario final. En la literatura el mismo también es conocido como Wireframe. La idea detrás de un Mockup es dibujar rápidamente un bosquejo de la interfaz de usuario final. Existen muchas herramientas digitales que ayudan a la construcción de los Mockups (Balsamiq, Pencil o MockuPlus).

El propósito principal de un Mockup es ayudar en la discusión de las especificaciones de la interfaz de usuario con la colaboración de los usuarios finales. Además, el mismo permite descubrir y definir requerimientos que no son de interfaz de usuario en un lenguaje más familiar a ellos, en contraposición a las especificaciones textuales planas que se estilan en algunas metodologías [[47],[48]]. En adición, los Mockups trabajan no solo como artefactos de requerimientos, sino como

ayuda en la elicitación de requerimientos general [49]. Existen estudios estadísticos que muestran como los Mockups mejoran la recolección de requerimientos en comparación con el método tradicional textual, sin implicar un esfuerzo adicional en el proceso [50]. Por último, los Mockups se vienen proponiendo como una herramienta exitosa para capturar y registrar requerimientos de forma fluida [51] – aquellos que generalmente se expresan oralmente o informalmente y que son una parte implícita (y generalmente se pierden) del proceso de elicitación. Luego, el equipo de desarrollo captura los conceptos principales desde los Mockups hacia un "Modelo de Interfaz de Usuario Estructural - Structural User Interface" (SUI) preservando el mapeo entre ambos.

2.4.2 El proceso MockupDD

El enfoque se inicia con la construcción de interfaces de usuario o Mockups; de allí proviene el nombre "Mockup Driven Development - Desarrollo Dirigido por Mockups (MockupDD)" [15] (ver figura 12).

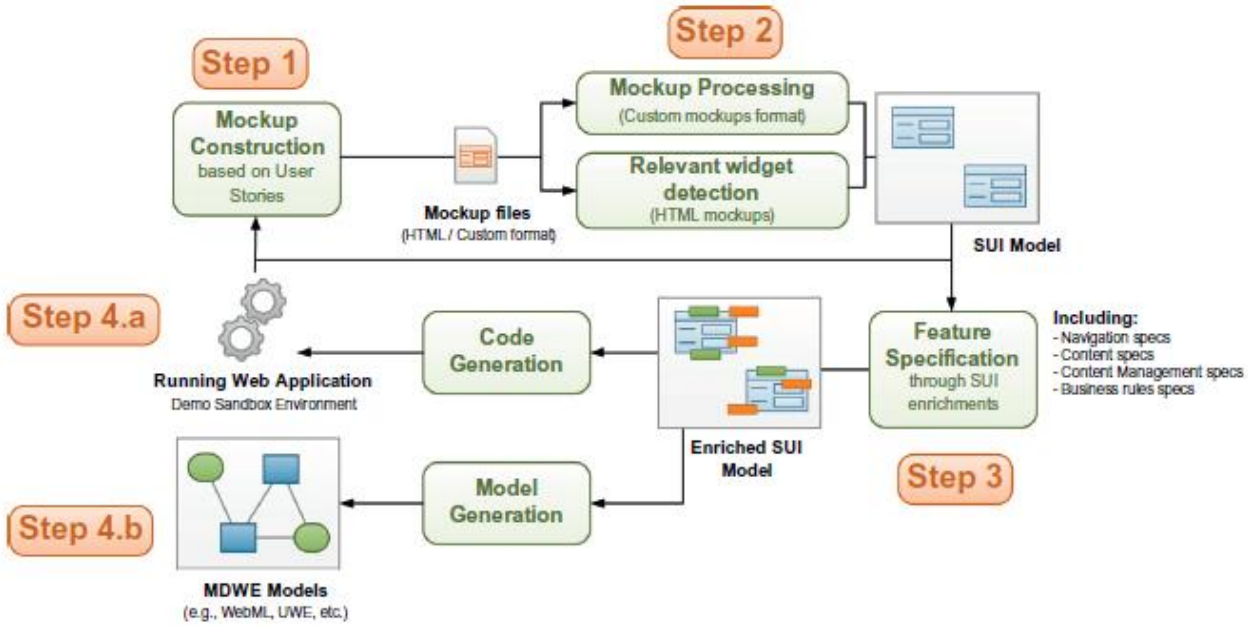


Figura 12 - El flujo de trabajo de una iteración MockupDD, incluyendo los pasos técnicos

El proceso MockupDD comienza con la rápida etapa de generación de requerimientos, cuyo resultado es un "Conjunto de Historias de Usuarios" que deben ser desarrolladas (Paso 1, figura 12). Los usuarios finales luego crean, a partir de las "Historias de Usuario", los Mockups que las representan gráficamente.

Una vez que todas las *Historias de Usuario* son representadas con *Mockups* y, luego mapeadas hacia *Modelos SUI* (Paso 2, figura 12) los requerimientos son especificados en la forma de etiquetas. Para hacer esto, el equipo de desarrollo trabaja junto con los stakeholders para interpretar la semántica en las *Historias de Usuario* y los *Mockups*, y etiqueta los *Mockups* con anotaciones que representan dicha semántica. Aunque los Modelos SUI preservan el mapeo desde el Mockup original fuente, las etiquetas se pueden ver aplicadas sobre los Mockups iniciales, pero de hecho son enlazadas a elementos SUI. Esto permite mantener una herramienta independiente que enriquece la estrategia (aunque los modelos SUI no dependen de ninguna plataforma, herramienta o tecnología UI) y, al mismo tiempo, permite la aplicación intuitiva de las especificaciones sobre los Mockups iniciales en su representación gráfica original, facilitando la trazabilidad de los requerimientos y la comprensión por parte de los usuarios finales.

2.4.2.1 Construcción y Procesamiento de Mockups

El proceso MockupDD comienza por la construcción de UI Mockups. El número de Mockups necesarios en cada iteración depende de cómo los pasos de interacción están distribuidos entre los nodos de navegación de la Aplicación Web. Además, varias *Historias de Usuarios* pueden compartir uno o más Mockups, ya que pueden implicar funcionalidades transversales.
Para formalizar la estructura de la interfaz de usuario, se usa el meta Modelo SUI. El meta Modelo SUI en MockupDD es muy similar a los populares Lenguajes de Descripción UI y a los estándares tales como UsiXML [52], XAML [1] o XUL [2] . Ellos definen una clase abstracta Widget; los widgets pueden ser SimpleWidgets (atómicos) o CompositeWidgets (contenedor de widgets), y ellos son agrupados en unidades navegacionales (Páginas). Sin embargo, en lugar de definir una estructura UI, el meta modelo SUI es diseñado para soportar la especificación de un extenso y personalizable conjunto de características usando elementos UI a través de etiquetas.

Una vez que se resaltaron y mapearon todos los widgets relevantes de acuerdo a los conceptos de las *Historias de Usuario*, se trabaja sobre la especificación de las características de los widgets.

2.4.2.2 Especificación de características

Para enriquecer los Mockups con diferentes tipos de especificaciones, se introduce el concepto de *tag/etiqueta*. Una *etiqueta* es un enriquecimiento atómico compuesto por un nombre y cero o más parámetros textuales (p.e. *Tag(Param1, Param2, ..., ParamN)*). Todo tipo de etiqueta puede ser aplicado solo sobre un subconjunto de widgets SUI. Además, cada etiqueta puede definir una sintaxis personalizada por cada parámetro, extendiendo su semántica tanto como sea necesaria. Las etiquetas son agrupadas en conjuntos de etiquetas para aislar cuestiones específicas como de navegación o de manipulación de datos, que pueden ser abordadas por separado en la mayoría de los casos. La figura 13 describe todas las etiquetas incluidas en la especificación MockupDD, junto con su semántica, aplicabilidad y correspondiente conjunto de etiquetas. La semántica de las etiquetas puede ser usada para descubrir y mapear requerimientos desde la participación de los usuarios finales (cuando aportan cuestiones vinculadas a la estructura UI), y además para especificar como la aplicación final debe comportarse.

[1] XAML Overview (WPF) – https://docs.microsoft.com/en-us/dotnet/desktop-wpf/fundamentals/xaml
[2] XUL | Mozilla Developer Network - https://developer.mozilla.org/en-US/docs/Archive/Mozilla/XUL

Tag set	Tag structure and semantics	Applicable over
Navigation/Interaction	**Node(<nodeId>)** "This page will be indentified with the id <nodeId>"	Page
Navigation/Interaction	**Link(<nodeId>)** "Clicking this Link/Button will trigger a navigation to previously identified <nodeId> Page"	Button/Link
Content	**Data(<typeName>)** "This widget must show or allow editing a(n) <typeName>"	Widget
Navigation/Interaction	**Select()** "This widget will be used to mark objects in order to perform some action"	SimpleWidget
Navigation/Interaction	**Transfer(<type1, ..., typeN)** "When navigating, <type1>, ... and <typeN> will be transferred to the destination page"	Button/Link
Content	**Save([<type1, ..., typeN>])** "Clicking that Link/Button will save <type1>, ... <typeN>"	Button/Link
Content	**Delete([<type1>, ..., <typeN>])** "Clicking that Link/Button will delete <type1>, ... <typeN>"	Button/Link
Content	**Associate(<type1>, <type2>[, <associationName>])** "Clicking that Link/Button will specify that <type1>'s <associationName> will be <type2>"	Button/Link
Content	**Dissociate(<type1>, <type2>[, <associationName>])** "Clicking that Link/Button will specify that <type1>'s <associationName> will not be <type2> anymore"	Button/Link
Content	**Query(<description>, <type>)** "Widgets content in this panel will be used to get a list of <type> through the query expression <description>"	Panel
Content	**Action(<action> [, <type1>, ..., <typeN>])** "Clicking in that Link/Button will trigger a(n) <action> action, involving <type1>, ..., <typeN>"	Button/Link

Figura 13 – Conjunto principal de etiquetas MockupDD – descripción, semántica y aplicabilidad

2.4.2.3 Refinamiento de etiquetas

En ésta etapa se pretende que las etiquetas sean lo más conceptuales posibles. Ello habilita a especificar conceptos en una forma incremental, por refinación de sus parámetros o combinación de sus semánticas cuando hay muchos de ellos aplicados para el mismo widget, o relacionados al mismo Mockup. Se define una etiqueta ambigua como una etiqueta que semánticamente no está completa por sí misma, y que puede tener más de una interpretación semántica válida. Si bien es más rápido de aplicar, estas etiquetas no tienen valor expresivo concreto hasta que son materializadas. En algunos casos, ésta materialización puede ser realizada automáticamente aplicando heurísticas.

2.4.2.4 Codificación y generación del modelo

Hasta este punto, se presentaron etiquetas como especificaciones aisladas y atómicas con una semántica específica. En ésta paso se muestra como ellas pueden ser transformadas hacia elementos MDWE y combinadas para especificar características de diseño más complejas. Cada iteración del proceso MockupDD involucra agregar cambios o remover etiquetas. Después de haber definido completamente las etiquetas, se usa un generador de modelos para obtener los correspondientes modelos MDWE.

2.4.2.5 Proceso de Desarrollo MockupDD

El proceso de desarrollo MockupDD, es una adaptación de Scrum [53]. Para lograr una mejor comprensión del mismo, se describe brevemente primero el proceso de desarrollo Scrum.

El proceso Scrum comienza con la construcción del *Product Backlog*, el cual es una lista de todas las características que el producto software debe tener, priorizado por el valor de la entrega al cliente. Luego el producto es construido iterativamente en algo llamado *Sprints*. Cada Sprint comienza con una reunión de planeación para desarrollar un plan detallado para la iteración. En el plan las características más importantes del *Product Backlog* se desglosan en tareas que forman el *Sprint Backlog*. Una vez que el *Sprint Backlog* es cuidadosamente definido para desarrollarse en no más de un mes, el equipo de desarrollo comienza a trabajar en él. Una corta reunión diaria Scrum (*Daily Scrum Meeting*) es realizada todos los días de trabajo para compartir el progreso del trabajo del Sprint y los nuevos problemas encontrados durante el Sprint. Finalmente, al final del Sprint, una aplicación potencialmente entregable es mostrada al *Product Owner*, luego el *Product Backlog* es re priorizado si es necesario y el objetivo para el próximo *Sprint* es definido.

Con MockupDD, el primer paso de un Sprint Scrum sigue la misma estrategia, excepto que el *Sprint Backlog* puede ser expresado como una lista de Historias de Usuario (*User Stories*) cuyo nivel de detalle debería ser suficiente como para ser descrito en Mockups. En lugar de usar código directamente, los Mockups son construidos para concretar las Historias de Usuario. Estos Mockups son luego traducidos hacia una representación SUI. Después de que los Mockups son completamente etiquetados, una demo de la aplicación funcionando, puede ser corrida para los usuarios finales o para el *Product Owner*. Para mostrar como la aplicación se está comportando con el modelado hecho hasta ese momento. Así, en lugar de esperar el final del *Sprint*, los stakeholders pueden ver un prototipo funcionando para evaluar si la aplicación se está comportando como se esperaba.

2.4.3 End User Grammar – Gramática de usuario Final (EUG)

Durante los últimos años, las *Metodologías Ágiles* han sido ampliamente adoptadas y su uso se ha convertido en un factor clave del éxito de los proyectos. Dichas metodologías reducen la distancia entre las expectativas del cliente y los entregables. La práctica ágil consiste pequeños ciclos de desarrollo en los cuales un entregable es liberado para ser confrontado con los requerimientos al final de cada iteración. Una de las herramientas más importantes para documentar adoptada por los practicantes ágiles es el Mockup. Mediante el uso de ésta técnica, la forma en la que los escenarios son instanciados se basa en ligeras descripciones textuales tales como "*Historias de Usuario (User Stories)*" o un marco de diseño de alta fidelidad que permite comunicar fácilmente el comportamiento de la aplicación a los stakeholders. Los Mockups son usados para describir escenarios en donde la información de ejemplo está lo más cerca posible de un uso real dentro del sistema que se pretende desarrollar. Sin embargo, la herramienta Mockup no es una especificación formal y la misma no provee la posibilidad de enumerar abstracciones tales como variables y entidades UML, por ejemplo. Un Mockup puede ser usado, por ejemplo, para describir nuevas iniciativas comerciales y el mismo puede poseer reglas de negocio diferentes que no pueden ser apreciadas con solo mirarlo. Este tipo de inconsistencias ciertamente no son detectadas debido a la ausencia de documentación apropiada. Por otro lado, además es difícil rastrear las definiciones de los requerimientos (y sus cambios) en aquellos entornos en donde los mismos se basan en Mockups y en *User Stories*.

Del contexto mencionado en el párrafo anterior proviene EUG [20](End User Grammar – Gramática de usuario Final) y dicha técnica propone lo siguiente: una notación coloquial y amigable para describir datos, navegación, requisitos de interacción y de negocios según especificaciones realizadas sobre Mockups, basados en MockupDD [15].

Los Mockups vienen prestando mucha atención al campo de la ingeniería de requerimientos ya que los mismos ayudan a construir especificaciones de interfaz de usuario en compañía de los usuarios finales, y además permite descubrir y definir requerimientos "que no son de interfaz de usuarios" en un lenguaje que es amigable para ellos, en contraposición a especificaciones planas textuales [[48], [47]] propuestas por otros enfoques. Además, los Mockups han demostrado ser un método efectivo para la captura de requerimientos fluidos [51]– aquellos que usualmente son expresados oral o informalmente (y que se pierden generalmente) y son una parte implícita del proceso de elicitación.

EUG propone un enfoque que no está ligado a una herramienta específica y se centra en mejorar la recopilación de requerimientos mediante anotaciones fáciles de usar. EUG depende en gran medida de la metodología MockupDD. Recordar que en el contexto de MockupDD, los Mockups son el núcleo de la metodología, es decir artefactos de especificación de requerimientos obligatorios, los cuales, en lugar de ser descartados como en los enfoques de desarrollos tradicionales, son reusados para definir especificaciones de software más específicas.

2.4.3.1 ¿Cómo se mejoran los Mockups con EUG?

Luego de que los analistas de software comprenden las necesidades iniciales de los clientes, los mismos están habilitados a comentar los Mockups con el fin de describir informalmente cómo la aplicación puede ser navegada y usada. Los Mockups pueden ser modelados usando cualquier herramienta con la que el analista posea experiencia. Los Mockups son una herramienta de validación de requerimientos que se consensua con los stakeholders. Los Mockups describen situaciones que provienen de escenarios de la vida real y dicha descripción se logra utilizando el cómo luciría la interfaz de usuario. Cuando el Mockup es ilustrado, *los analistas toman ventaja del lenguaje que los stakeholders utilizan*. Es decir que las anotaciones utilizadas sobre los Mockups son entendibles para los stakeholders y para los analistas. Basándose en los requisitos precedentes EUG proporciona un catálogo de anotaciones o una gramática de usuario final. Dicho catalogo permite reducir el nivel de ambigüedad en las anotaciones y mejorar los diagramas realizados por los analistas en una descripción de datos coloquial.

2.4.3.2 Catálogo de anotaciones EUG

End User Grammar (EUG) [16] se centra en describir fuentes de información, formatos e información relacionada. Cada anotación es una definición coloquial estructurada la cual es legible para los usuarios finales ya que la misma no presenta ningún concepto técnico que pueda limitar su entendimiento. A continuación, se presenta el mismo.

"Mockup Name" view (number)
Define un ID (número) para que un Mockup pueda ser referenciado como un destino para navegar/activar por otra etiqueta.

a[n] [list of] class
Denota que un objeto o una lista de objetos de una clase *Class* es mostrada o puede ser manipulada en la UI (Interfaz de Usuario).

Class's attribute [which is a datatype | with options: value1,…, and valueN]
Especifica que el atributo de un objeto de una clase de objetos *Class* (llamado attribute) es mostrado o puede ser editado a través de un widget gráfico básico. Opcionalmente, un datatype puede ser definido para ese atributo (tipo Date, String, Integer, Decimal, Boolean, una enumeración Integer o Blob). Si no es especificado un datatype, por defecto se asume como String. En los eventos de una enumeración es posible listar los valores posibles usando la cláusula "with options o1, o2, …, oN".

a Class1 has a[n][optional][list of] Class2, called "aRealName"
Denota que un objeto de *Class2* es mostrado o puede ser mostrado a través de un elemento básico en la UI. Sin embargo, este elemento se puede navegar desde una asociación llamada *associationName* desde otro elemento de la *Class1*.

Subclass is a type of Superclass
Denota que un objeto de la clase *Subclass* es mostrado o puede ser manipulado en la Interfaz de Usuario y que la clase de dicho objeto (*Subclass*) hereda de otro llamado *Superclass*.

Navigates to <destination>
Opens a popup <destination>
Denota que, cuando se ejecuta una acción por defecto a través de un elemento gráfico básico (por ejemplo, un click) el Mockup destino puede ser mostrado, navegado, o enfocado – el Mockup destino puede ser etiquetado con *mockupName view (number)* y *<destination>* puede referenciar a dicho número.

Class's attribute is required
Denota que un valor no puede ser vacío para el atributo attribute de la clase *Class*. El valor es requerido.

Class's attribute min value is minimumValue
Class's attribute max value is maximumValue
Class's attribute values must be between minimumValue and maximumValue
Denota que los valores para el atributo *attribute* de la clase *Class* deben ser menor o igual al *maximumValue* y/o mayor o igual a *minimumValue*.

Class's attribute matches regularExpresion
El contenido del atributo *attribute* de la clase *Class* debe ser formateado para que coincida con un patrón (regularExpresion). Por ejemplo, fechas, números telefónicos y códigos pueden tener restricciones específicas de formateo.

[Saves | Deletes] a Class
Denota que, al hacer clic o activar una acción predeterminada sobre el widget, una instancia de *Class* (la cual está siendo editada) puede ser creada o eliminada respectivamente.

Triggers "action description"
Indica que se invocará una acción arbitraria (descrita textualmente) al ejecutar la acción predeterminada sobre el widget. Esta construcción se utiliza cuando el comportamiento esperado no está definido aún, pero debe señalarse.

2.4.3.3 Descripción de datos coloquial de un Mockup con EUG

Los Mockups a menudo usan un escenario de la vida real definido con datos ilustrativos que describen cual debe ser la experiencia del usuario de la aplicación - User eXperience (UX). Durante este paso, los analistas pueden enriquecer los elementos gráficos del Mockup con plantillas basadas en las descripciones coloquiales mencionadas.

La ventaja principal de los EUG es que pueden ser fácilmente comprendidos por los usuarios finales y además proporcionan la formalidad requerida para ser procesados y, consecuentemente, permiten una mejor validación de los requerimientos. *Cada expresión descripta debe coincidir con una plantilla específica, con marcadores de posición bien definidos que facilitarán la automatización posterior del procesamiento*.

En la siguiente figura (ver figura 14) podemos ver como un Mockup fue documentado utilizando etiquetas EUG. En la figura se presenta un conjunto simple de anotaciones que especifican los requerimientos, sin embargo, dicho conjunto de anotaciones puede ser aún más extenso y cubrir un conjunto más grande de elementos.

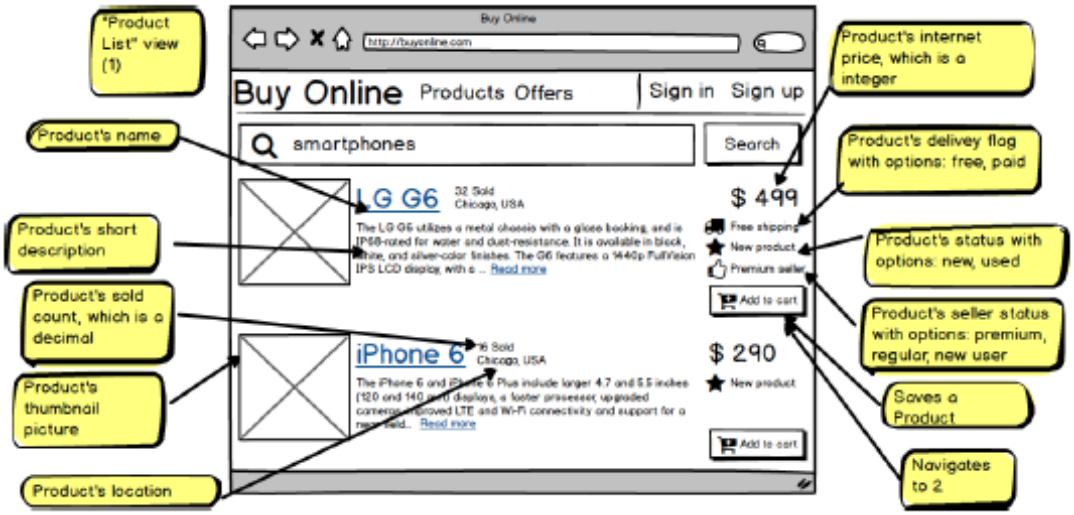


Figura 14 - Mockup anotado con etiquetas EUG

Hasta aquí hemos presentado el "Estado del Arte" de todos aquellos enfoques y conceptos que sirvieron como base para la elaboración de la presente tesis. En la misma se conjugan y combinan BPMN, NLP y EUG. A continuación, presentamos "End User Grammar Extended for Business Processes – Gramática de Usuario Final Extendida para Procesos de Negocios".

Capítulo 3 Técnica End User Grammar Extended for Business Processes

En éste capítulo hacemos referencia a aquellos conceptos que sirvieron de base para el desarrollo de la tesis mencionados en el capítulo 2. Luego, se expone un ejemplo que guía la presentación de la Técnica EUGEBP.

3.1 Introducción

La presente tesis propone realizar anotaciones de usuario final en lenguaje natural y simple, pero de una manera sistematizada, sobre los Mockups de la aplicación. Además, propone utilizar dichas anotaciones para, colaborar con la captura del conocimiento que poseen los usuarios sobre los procesos de negocios de la organización, y contribuir con la identificación de los mismos. En otras palabras se propone colaborar con la fase del ”Levantamiento del Proceso”[19] mediante la adopción del artefacto Mockup proveniente de las metodologías ágiles[15]. Para ello se desarrolló una técnica que consiste en utilizar un conjunto de reglas que sirve de referencia para el proceso de producción de anotaciones, elaboradas por los stakeholders y por el analista, sobre los Mockups. En esencia dichas reglas indican que las anotaciones deben producirse utilizando oraciones simples en lenguaje natural, pero con una estructura gramaticalmente correcta para el idioma en cuestión (para el contexto de la tesis el español). El diseño de dicho conjunto de reglas está basado en aprovechar el correcto uso del lenguaje natural. Si las anotaciones sobre los Mockups son realizadas de acuerdo a las reglas de redacción propuestas, luego es posible extraer de las oraciones que componen las mismas los elementos BPMN contenidos implícitamente en él texto. Dicha extracción puede realizarse manualmente (es decir a través del análisis minucioso de las oraciones realizado por el equipo de analistas) o a través del uso de herramientas NLP (que ayudan en el procesamiento de oraciones en lenguaje natural, pero de una manera automática). Una vez realizada la extracción de los Elementos BPMN (junto con todas sus características) es posible derivar manualmente él o los Modelos BPMN contenidos en los Mockups.

La técnica propuesta por la tesis se denominó “End-User Grammar Extended for Business Processes” (EUGEBP) o “Gramática de Usuario Final Extendida para Procesos de Negocio”. La misma ésta compuesta por un conjunto de reglas que guían el proceso de producción de anotaciones sobre los Mockups y por una serie de pasos que luego permiten derivar manualmente el o los Modelos BPMN desde las anotaciones realizadas sobre los Mockups. Se aclara además que *los Modelos BPMN derivados están acotados a la expresión mínima de los mismos*. Es decir que *se identifica solamente el Flujo Normal de los BP* (sección 2.2.3.6). En consecuencia, en la sección 3.3 se mencionan las limitaciones de la técnica propuesta en cuanto a los elementos BPMN que no son considerados por la misma.

A continuación, en las siguientes secciones se presenta en detalle cómo está conformada la técnica.

3.2 Elementos BPMN a identificar

En ésta sección presentamos brevemente los elementos que componen a un proceso de negocio (para el contexto de la tesis) desde el punto de vista de la notación BPMN. La notación BPMN es una notación gráfica que permite modelar procesos de negocios. La misma es un estándar que incluye cuatro categorías de elementos: Objetos de Flujo (Actividades, Eventos y Compuertas de Decisión), Delimitadores (Pools o Contenedor de Carriles, y Lanes o Carriles), Artefactos (Objetos de datos, Anotaciones de texto) y Objetos de conexión (Flujos de Secuencia, Flujos de Mensajes y Asociaciones).

El enfoque propuesto en ésta tesis propone identificar y derivar desde los Mockups el siguiente conjunto de elementos BPMN:

Actividades

De Usuario, de Servicio

Eventos

Simples, de Inicio y *de Fin*. Indican simples puntos de inicio y de fin.

Compuertas

Básica Exclusiva (selecciona exactamente un flujo de secuencia saliente) y *Básica Paralela* (todos los caminos salientes son activados).

Swinlanes

Pools y *Lanes*. A través de los mismos es posible identificar a los participantes, es decir los roles que deben ejecutar las actividades encontradas.

Conectores

Flujos de Secuencia (caminos que comunican *Lanes*) y *Flujos de Mensaje* (caminos que comunican *Pools)*.

Todos los elementos mencionados provienen de la literatura relacionada a BPMN [3], [19], [21], [23] y son solo una parte de los mismos. Existen más elementos vinculados a BPMN, sin embargo, en la presente tesis solo nos vamos a ocupar de los mencionados precedentemente. Para mayor información remitirse al siguiente enlace [3].

3.3 Limitaciones: Elementos BPMN no considerados

En ésta sección presentamos brevemente los elementos que no van a ser tratados en la tesis. Si bien los mismos podrían estar contenidos dentro de un Modelo BPMN derivado, se consideró adecuado solo enfocarse en los elementos BPMN vinculados a la rápida identificación de Flujo Normal de un BP. Los elementos BPMN que no van ser considerados podrían ser considerados en una futura investigación. A continuación, listamos los mismos:

Actividades

Manual, De Envío, de Recepción, de Regla de Negocio, de Ejecución de Script.

Eventos

Simple Intermedio, Mensaje, Temporal, Escalable, Condicional, Enlace, Error, Cancelación, Compensación, Señal, Múltiple, Paralelo Múltiple, Terminación.

Compuertas

Basada en Eventos, Inclusiva, Exclusiva Basada en Eventos, Compleja, Paralela Basada en Eventos.

Conectores

Flujos Por Defecto, Condicional

Objeto de Datos

Asociaciones

Para mayor información remitirse al siguiente enlace [4].

3.4 Un ejemplo de la EUGEBP aplicado

Para explayarnos en el detalle de EUGEBP, a continuación, enunciamos un breve ejemplo (que va a ser utilizado como marco de referencia de aquí en adelante) que ayuda a la comprensión de la técnica.

[3] http://www.bpmb.de/index.php/BPMNPoster
[4] http://www.bpmb.de/index.php/BPMNPoster

Supongamos que estamos trabajando en una aplicación destinada a la venta de productos en la web. Nuestros stakeholders nos mencionan reglas de negocio, conceptos de negocio y una idea inicial del proceso de negocio que desean implementar: "*Solo un cliente registrado en el sitio puede entrar a comprar. El cliente puede comprar uno o más de un producto. El cliente puede pagar con cualquier medio de pago electrónico. El empleado debe visualizar todas las compras realizadas en el sitio*". Para comprender las necesidades y los distintos puntos de vista de las partes interesadas en la aplicación, los analistas inician un proceso de elicitación de requerimientos de software. Utilizan la técnica EUGEBP y trabajan (junto con los stakeholders) en la producción de anotaciones de interacción y de negocios sobre los Mockups.

Para simplificar el ejemplo, a continuación, se muestra cómo debe ser aplicado EUGEBP en un entorno de "Elicitación de Requerimientos de Software Inicial" sobre un conjunto de Mockups (Figuras 15, 16, 17, 18), de acuerdo a las especificaciones mencionadas por los stakeholders:

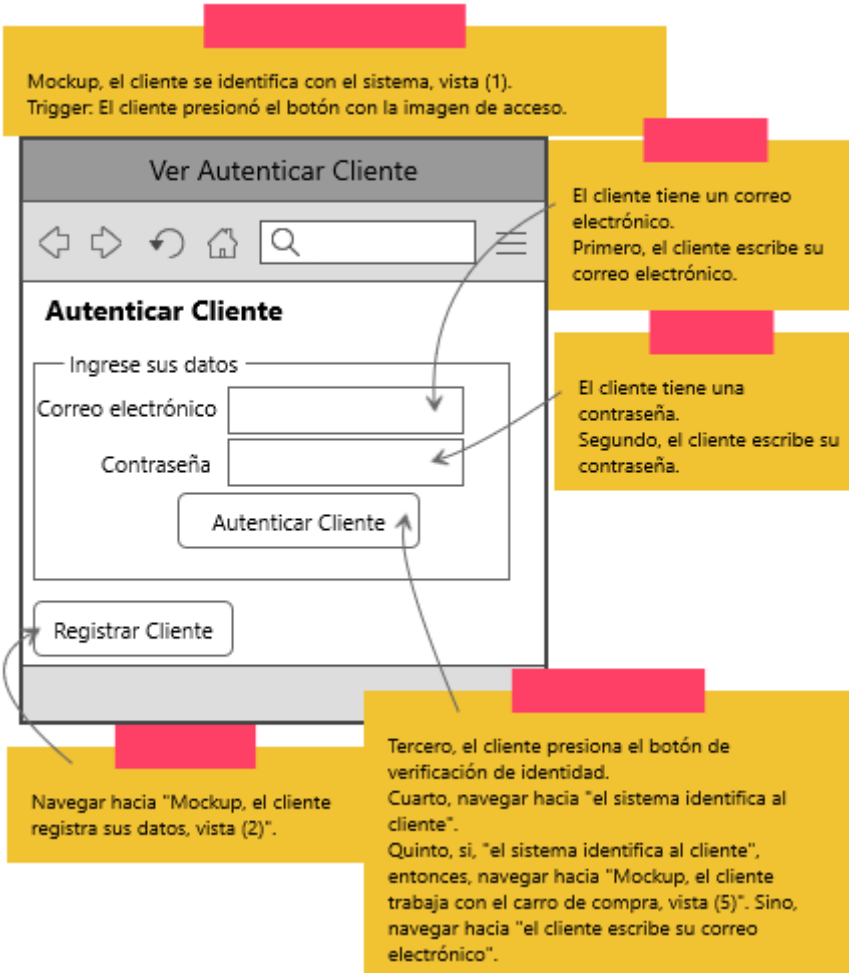


Figura 15 – Mockup, el cliente se identifica con el sistema, vista (1)

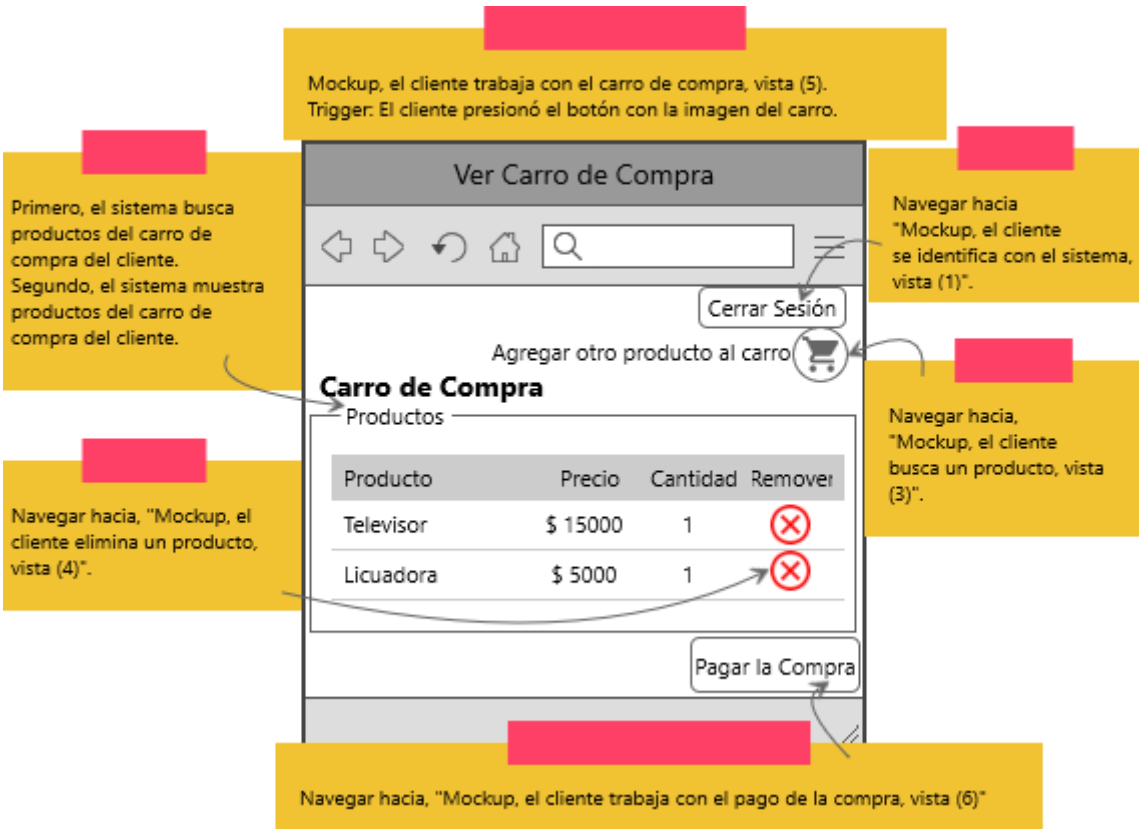


Figura 16 – Mockup, el cliente trabaja con el carro de compra, vista (5)

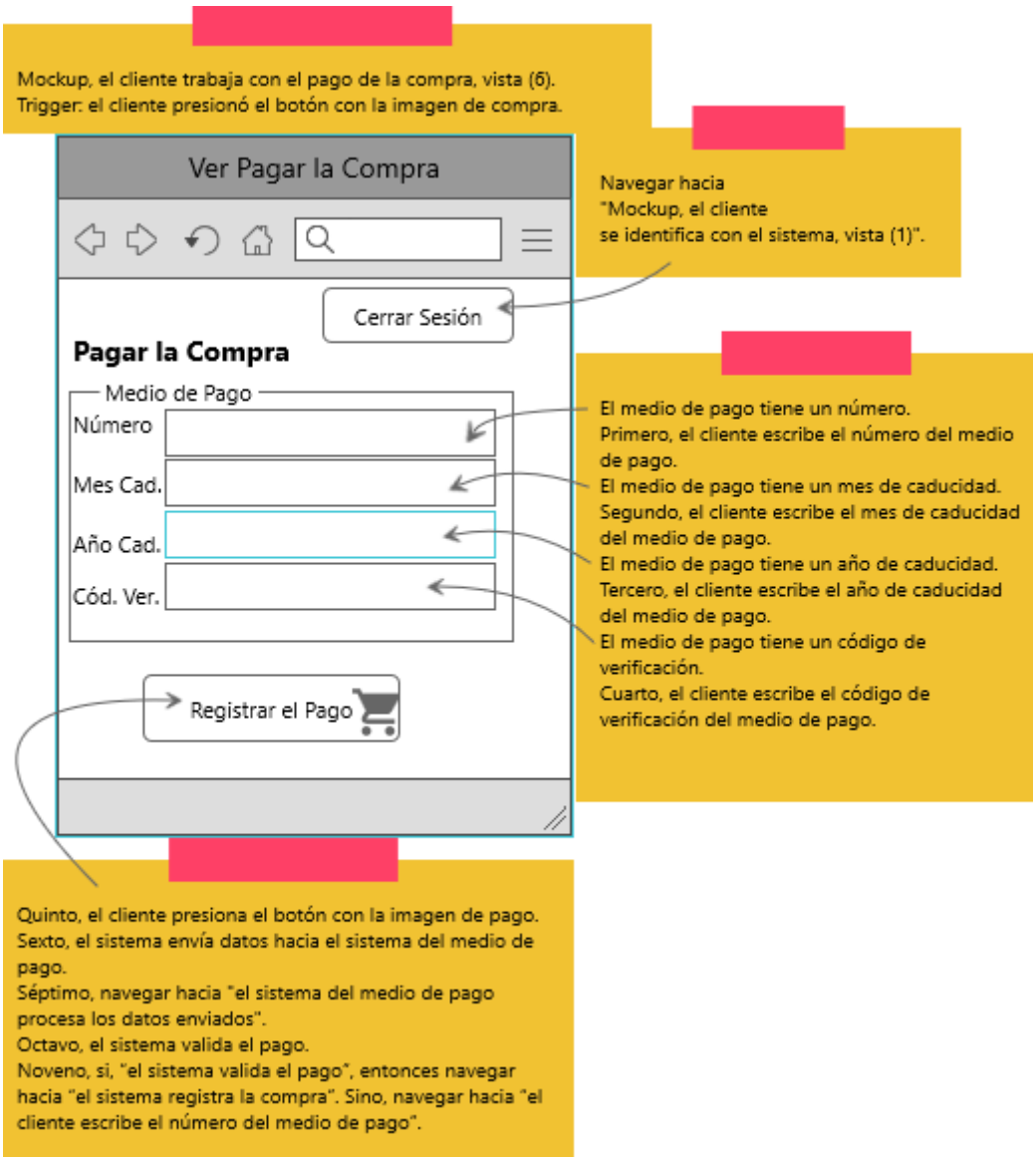


Figura 17 – Mockup, el cliente trabaja con el pago de la compra, vista (6)

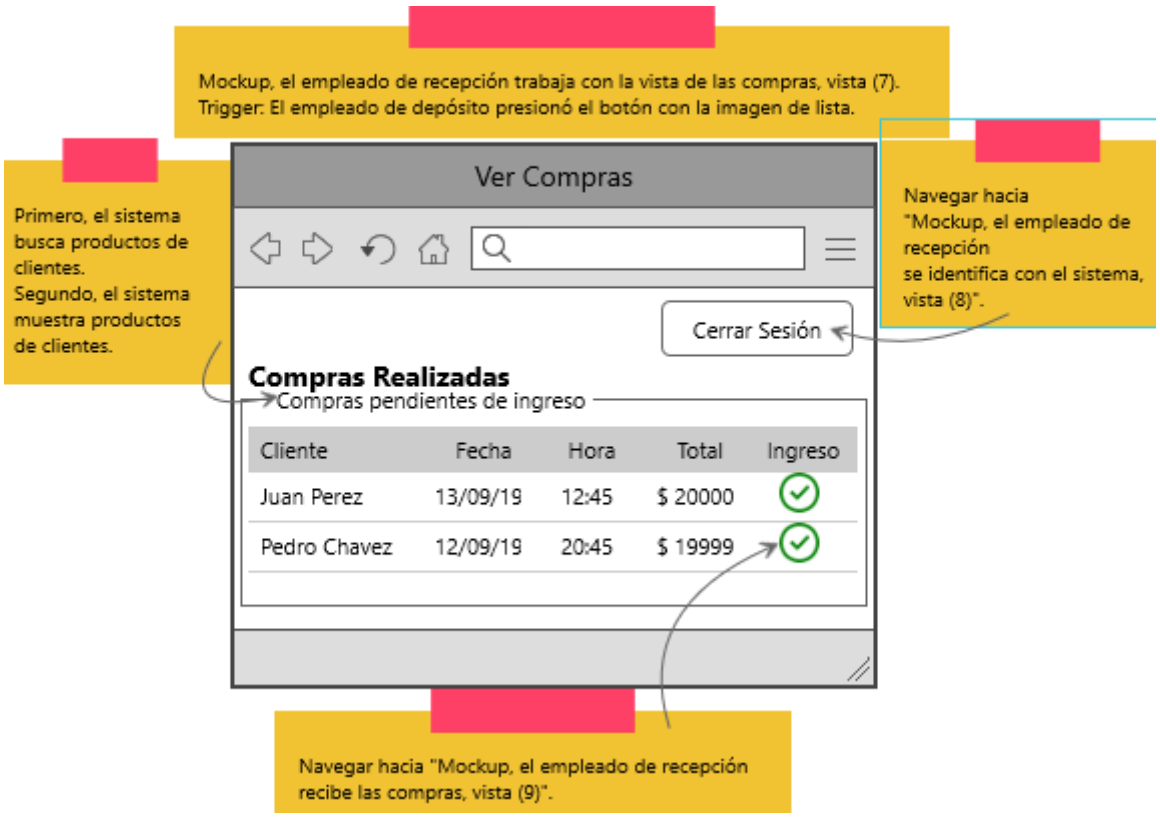


Figura 18 – Mockup, el empleado de recepción trabaja con la vista de las compras, vista (8)

A continuación, presentamos cómo identificar elementos BPMN desde las anotaciones EUGEBP aplicadas sobre los Mockups del ejemplo precedente.

3.4.1 Identificando Swimlanes BPMN en Mockups

BPMN utiliza Swimlanes (Canales) para ayudar a dividir y organizar actividades en un diagrama. Los tipos de canales que existen son los siguientes:

Pool: actúa como contenedor para un proceso, cada uno representa un participante en un diagrama de procesos de negocio colaborativo.

Lane (Carril): utilizado para representar un rol de negocio interno dentro de un proceso. Proveen un mecanismo para particionar los objetos dentro de un Pool.

Un “participante” se define como un rol de negocio, por ejemplo, un cliente o un vendedor. Cada pool puede representar un solo participante. Los pools son representados como una caja rectangular que actúa como contenedor para los objetos de flujo, del proceso de un participante. Cada pool representa un proceso diferente y cada participante tiene su propio pool.
Existe un pool conocido como “caja negra”, éstos no muestran actividades o flujos de secuencia. Se utilizan cuando no se posee conocimiento del comportamiento del participante. [21]

Para identificar canales dentro de anotaciones u oraciones en lenguaje natural podemos hacer referencia al "sujeto"[5] o al "objeto indirecto"[6] de la oración [7]. El "sujeto" o el "objeto indirecto" pueden ser, algo o alguien que ejecuta la acción (por ejemplo, un usuario final o un sistema). Es decir que es posible utilizar al "sujeto" y al "objeto indirecto", contenidos dentro de las oraciones, para identificar "Swimlanes". Este concepto se adoptó y valiéndose del mismo se realizaron anotaciones sobre los Mockups. Esto permitió derivar el elemento BPMN Swimlane.
Las reglas de mapeo para identificar "Swimlanes" (ver tablas 10 y 11) que se proponen en la presente tesis son:

Regla	Descripción	Oración Ejemplo	SwimLane
Regla 1	El sujeto de una oración	El cliente *[sujeto]* presionó el botón con la imagen del carro. (ver figura 16)	cliente (ver figura 19)

Tabla 10 – Regla 1 de mapeo para identificar "Swimlanes"

Figura 19 –SwimLane: "c*liente*"

Regla	Descripción	Oración Ejemplo	SwimLane
Regla 2	El objeto indirecto de una oración	El sistema envía datos hacia el sistema del medio de pago *[objeto indirecto]*. (ver figura 17)	sistema del medio de pago (ver figura 20)

Tabla 11 – Regla 2 de mapeo para identificar "Swimlanes"

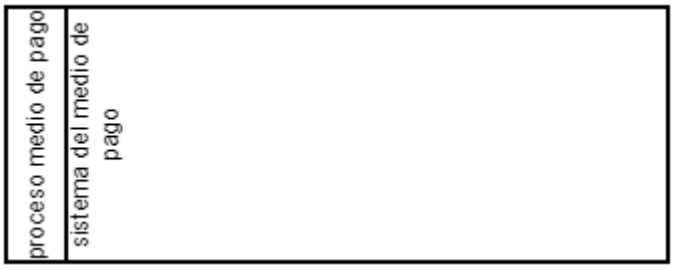


Figura 20 –SwimLane: "*sistema del* m*edio de pago*"

[5] En toda oración se pueden diferenciar dos partes: sujeto y predicado.
El sujeto indica quien realiza la acción o de quien se dice algo.
Ejemplo de sujeto que realiza la acción: "El tren llega a la estación"
Ejemplo de sujeto de quien se dice algo: "La pared es blanca".
El predicado describe la acción que realiza el sujeto o lo que se dice del sujeto. En el predicado siempre hay un verbo, que concuerda en persona y número con el sujeto.
Ejemplo de predicado que describe la acción: "El tren llega a la estación".
Ejemplo de predicado que dice algo del sujeto: "La pared es blanca".
[6] El objeto indirecto, también conocido como complemento indirecto, es la parte del predicado que indica a quién está dirigida o quién recibe la acción que realiza el sujeto. Está formado por una preposición que indica hacia quién se dirige (a, para, entre otras) y un sustantivo o un pronombre que lo sustituye. Para identificar el objeto indirecto (la parte del predicado junto con la acción del verbo) debe responder a las siguientes preguntas: ¿A qué?, ¿A quién?, ¿Para qué?,¿Para quién? Ejemplos:
Victoria enseña a sus amigas a cocinar. (¿A quién enseña a cocinar?, a sus amigas).
El doctor atendió a sus pacientes. (¿A quiénes atendió?, a sus pacientes).

Desde EUG se tomó el patrón "La *Clase tiene un atributo"*. El mismo permite especificar el *atributo* de una *Clase*. Para el contexto de EUGEBP se utiliza la anotación gramaticalmente correcta. Simplemente se anota sobre el Mockup la oración simple correspondiente. Por ejemplo: "*El medio de pago tiene un número*". De esta manera se logra percibir que la clase "Medio de Pago" es también el sujeto de una oración. Por lo tanto, en el contexto de EUGEBP, "Medio de Pago" además es un "SwimLane". Una nueva regla sería entonces (ver tabla 12) la siguiente:

Regla	Descripción	Oración Ejemplo	SwimLane
Regla 3	El/La [Clase] tiene un [Atributo]	El medio de pago *[Clase]* tiene un número *[Atributo]*. (ver figura 17)	medio de pago (ver figura 21)

Tabla 12– Regla 3 de mapeo para identificar "Swimlanes"

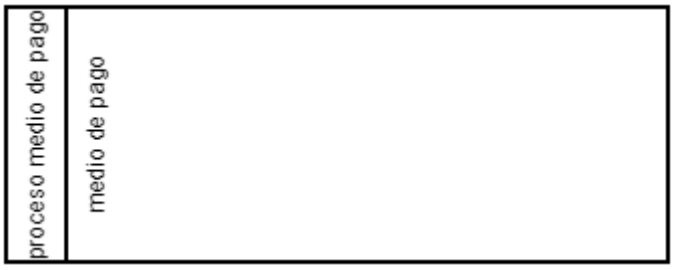


Figura 21 - SwimLane: "m*edio de pago*"

3.4.2 Identificando Actividades BPMN en Mockups

Desde la literatura vinculada a BPMN se define lo siguiente:
Una "actividad" es una unidad de trabajo a realizar dentro de un proceso de negocio. Las actividades se pueden denotar de la siguiente manera: <<acción sobre un objeto>>. Donde, la actividad se denomina con un verbo (acción) y un sustantivo (objeto).
Las actividades pueden ser del tipo atómicas (es decir, ser el nivel más bajo de detalle presentado en un diagrama) o compuestas (es decir, no ser atómica, en el sentido de que se puede expandir para ver otro nivel inferior de proceso).
El tipo de actividad atómica se conoce como tarea.
El tipo de actividad compuesta se conoce como subproceso.
Las actividades pueden ser ejecutadas por personas y/o aplicaciones. [21]

Teniendo presente la sección 2.2.3.1.1 (Tipos de Actividades) y la definición de "actividad" a continuación se presentan dos tipos de actividades que se pueden identificar dentro del ámbito de los Mockups: "actividades de usuario" y "actividades de servicio".

Actividad de usuario
Es importante notar que toda acción que pueda llevarse a cabo sobre un Mockup puede ser interpretada como una actividad de usuario o como una tarea pequeña. Tener presente que de hecho el Mockup es una representación de lo que finalmente será una pantalla real de la aplicación final con la que el usuario final interactuará. Por ejemplo, un Mockup puede contener un objeto del tipo campo de texto para el cual se espera que un usuario final introduzca un dato necesario o requerido. Por ejemplo, "*el cliente escribe su correo electrónico*". Dicha acción, dentro del Mockup, es una actividad de usuario (ver figura 22).

Figura 22 – Elemento BPMN del tipo "Actividad de usuario" derivado desde un Mockup

Además, este tipo de actividad, al ser indivisible, también puede ser catalogada como una actividad atómica.

Actividad de usuario del tipo Sub-proceso

Es de destacar además que cada Mockup representa una tarea de usuario para realizar en sí misma. Por ejemplo, el "*Mockup, el cliente trabaja con el pago de la compra, vista (6)*", de la figura 17, representa en si la actividad de usuario "*el cliente trabaja con el pago de la compra*". Y, a su vez representa una Actividad del tipo Sub-proceso porque puede desglosarse en un nivel mayor de detalle (ver figura 23).

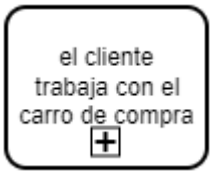


Figura 23 – Elemento BPMN del tipo "Sub-proceso" derivado desde un Mockup

Actividad de servicio

Es posible notar que dentro de un Mockup pueden existir actividades que no son llevadas a cabo por un usuario final, sino que, las mismas son ejecutas por aplicaciones o sistemas. En el contexto de un Mockup no nos interesa saber cómo se van llevar a cabo dichas tareas de sistema (las mismas se pueden ver como cajas negras a las cuales les enviamos datos de entrada y nos devuelven datos de salida), pero si nos interesa saber si a dicha tarea la va a implementar de forma automática una aplicación. Una actividad de servicio puede ser, por ejemplo, la tarea de procesar una serie de cálculos complejos (que involucra recursos y otras actividades automáticas de sistema) y retornar un valor hacia el Mockup. Es decir que dicha actividad la puede llevar a cabo, de forma automática, el sistema. Por ejemplo, la oración "*el sistema del medio de pago procesa los datos enviados*" puede ser considerada como una actividad de servicio que tiene que llevarse a acabo por el sistema que da soporte al "medio de pago" (ver figura 24).

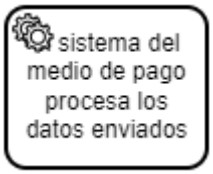


Figura 24 – Elemento BPMN del tipo "Actividad de servicio" derivada desde un Mockup

Para identificar actividades BPMN dentro de oraciones en lenguaje natural tomamos como referencia el siguiente concepto: las oraciones simples que contienen tiempos verbales en presente[7], representan actividades [7].

[7] Los verbos en presente son aquellos verbos que refieren a acciones que se están realizando en el momento actual. Por ejemplo: Juan juega al fútbol

Por otro lado, para identificar explícitamente una actividad de usuario del tipo sub-proceso, la presente tesis propone utilizar el prefijo "Mockup," al comienzo de la oración. Y, además propone, utilizar el prefijo "E/el sistema" dentro de la oración para identificar actividades de servicio.

En consecuencia, se proponen las siguientes reglas para identificar "Actividades" BPMN dentro del ámbito de los Mockups. Ver tablas 13, 14 y 15.

Regla	Descripción	Oración Ejemplo	Actividad
Regla 4	[Sujeto] + [Verbo en tiempo presente] + [Objeto]	El cliente *[Sujeto]* escribe *[Verbo en tiempo presente]* su número de medio de pago *[Objeto]*. (figura 17)	el cliente escribe su número de medio de pago (ver figura 25)

Tabla 13 – Regla 4 de mapeo para identificar "Actividades"

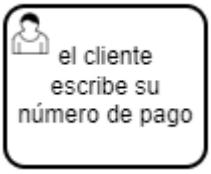


Figura 25 – Actividad de usuario: "e*l cliente escribe su número de medio de pago*"

Observación: Para el contexto EUGEBP la palabra reservada "Mockup" implica que las anotaciones que las contienen permiten identificar Actividades BPMN del tipo Sub-Proceso.

Regla	Descripción	Oración Ejemplo	Actividad
Regla 5	Mockup, [Sujeto] + [Verbo en tiempo presente] + [Objeto] , vista (número)	Mockup, el cliente *[Sujeto]* trabaja *[Verbo en tiempo presente]* con el carro de compra *[Objeto]*, vista (5) (figura 16).	el cliente trabaja con el carro de compra (ver figura 26)

Tabla 14 – Regla 5 de mapeo para identificar "Actividades"

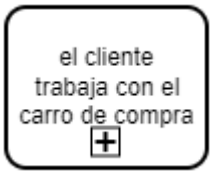


Figura 26 – Actividad tipo sub-proceso: "e*l cliente trabaja con el carro de compra*"

Regla	Descripción	Oración Ejemplo	Actividad
Regla 6	E/el sistema [Verbo en tiempo presente] + [Objeto]	el sistema identifica *[Verbo en tiempo presente]* al cliente *[Objeto]* (figura 1)	el sistema identifica al cliente (ver figura 27)

Tabla 15 – Regla 6 de mapeo para identificar "Actividades"

Figura 27 – Actividad de servicio: "e*l sistema identifica al cliente*"

3.4.3 Identificando Eventos BPMN en Mockups

Se define a un "evento" como algo que sucede durante el curso de un proceso de negocio. Afecta al flujo del proceso de negocio y puede iniciar, retrasar, interrumpir o finalizar al mismo. [21]

Los eventos simples que vamos a cubrir en ésta sección son los de *Inicio* y de *Fin*. Lo que tratamos de identificar, dentro del Mockup, es un elemento que genere un suceso o un comportamiento relevante. El elemento UI del Mockup que se señala en primer lugar, en este contexto, es el "Botón o Link". La acción de presionar un botón o un link, desencadena o dispara explícitamente una acción o suceso (la llamada a otro Mockup, por ejemplo). Recordemos además que el simple hecho de terminar con una actividad de usuario y dirigirse a realizar otra, dentro del Mockup, genera un suceso que puede aprovecharse como un evento implícito para el proceso de negocio que pretendemos identificar. Es decir que las inferencias vinculadas a los eventos que se puedan identificar dentro de un Mockup pueden ser anotadas y aprovechadas sobre el mismo.

Desde la literatura vinculada al estudió del procesamiento del lenguaje natural, las anotaciones u oraciones que contienen el tiempo verbal en pretérito perfecto[8] del tipo, "[sujeto] + [verbo en pretérito] + [objeto]", representan eventos [7]. Este tipo de oración se adoptó para el propósito de la tesis, pero con una ligera modificación. El tiempo verbal que se utilizó fue el "pretérito perfecto simple[9]". La regla de mapeo para identificar eventos propuesta es la siguiente (ver tabla 16):

Regla	Descripción	Oración Ejemplo	Evento
Regla 7	[Sujeto] + [Verbo en pretérito perfecto simple] + [Objeto]	El empleado de depósito *[Sujeto]* presionó *[Verbo en pretérito perfecto simple]* el botón con la imagen de lista *[Objeto]*. (figura 18)	el empleado de depósito presionó el botón con la imagen de lista (ver figura 28)

Tabla 16 – Regla 7 de mapeo para identificar "Eventos"

[8] El pretérito perfecto expresa acciones realizadas en el pasado y que perduran en el presente. Al igual que todos los tiempos compuestos se forma con el verbo auxiliar HABER y el participio del verbo conjugado.
(yo) he, (tú) has, (usted, él, ella) ha, (nosotros/as) hemos, (vosotros/as) habéis, (ustedes, ellos, ellas) han. El pretérito perfecto se emplea cuando queremos expresar que el pasado ha ocurrido recientemente. Así, si esta mañana he realizado un paseo diré "he paseado", pues la acción que menciono ya ha ocurrido, pero todavía forma parte del presente, ya que es algo que ha sucedido hoy. Esto quiere decir que el pretérito perfecto alude a experiencias del pasado que están en conexión con el tiempo presente.

[9] Pretérito perfecto simple, también conocido como pretérito indefinido de indicativo, se usa para expresar acciones que tuvieron lugar en el pasado de forma puntual y ya finalizaron o se interrumpieron como, por ejemplo: canté, subieron, leyó.

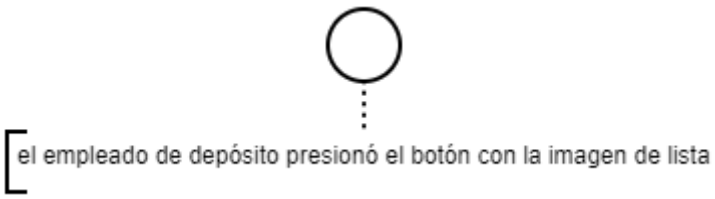


Figura 28 – Evento: "*el empleado de depósito presionó el botón con la imagen de lista*"

Por otro lado, el patrón EUG que nos permite identificar y describir elementos BPMN del tipo "evento" es *Triggers "action description"*, ya que el mismo permite explayarse en su descripción. Es decir que mediante el uso del mismo podemos anotar cuales son las situaciones que van a desencadenar un suceso. Cada movimiento del usuario dentro del Mockup puede disparar un evento (por ejemplo, tabular en un campo o presionar un botón). En este contexto, el patrón mencionado puede ser utilizado dentro del Mockup para anotar y describir cuales son los eventos disparadores encontrados.
Desde este patrón entonces, proponemos para la tesis la siguiente regla (Ver tabla 17):

Regla	Descripción	Oración Ejemplo	Evento
Regla 8	Trigger: [Sujeto] + [Verbo en pretérito perfecto simple] + [Objeto]	Trigger: El cliente *[Sujeto]* presionó *[Verbo en pretérito perfecto simple]* el botón con la imagen de compra *[Objeto]*. (figura 17)	el cliente presionó el botón con la imagen de compra (ver figura 29)

Tabla 17 – Regla 8 de mapeo para identificar "Eventos"

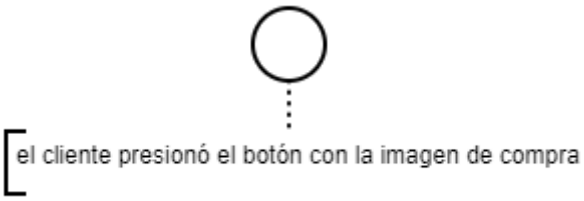


Figura 29 – Evento inicial: "*el cliente presionó el botón con la imagen de compra*"

Es importante notar que las oraciones vinculadas a la regla "8" dentro del Mockup, deben ir precedidas por el prefijo "Trigger:". Recordar que se toma desde EUG la forma *Triggers "action description"*. La misma permite describir una acción textualmente. En nuestro caso describimos una acción que ya pasó y que para nuestro contexto es un evento.

Observación: Para el contexto EUGEBP la palabra reservada "Trigger" implica que la anotación permite representar un Evento BPMN del tipo inicial.

Por último, para determinar un "Evento" del tipo "Final" proponenos lo siguiente: Si el último "Objeto de Flujo" identificado, del "Flujo Principal", no posee un "Flujo de Salida" entonces, simplemente le agregamos un "Evento Final" al "Objeto de Flujo" final (ver tabla 18).

Regla	Descripción	Oración Ejemplo	Evento
Regla 9	"Objeto de Flujo", del "Flujo Principal", sin "Flujo de Salida"	el sistema registra la compra *[Objeto de Flujo]* (figura 17)	el evento final se infiere desde el ultimo objeto de flujo sin flujo de salida (ver figura 30)

Tabla 18 – Regla 9 para identificar un "Evento final"

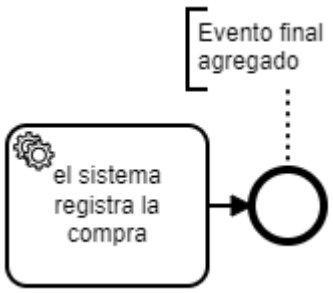


Figura 30 – Evento Final: se infiere desde la última "Actividad de servicio" sin flujo de salida

Observación: más adelante (en la sección 3.6.1.4) se muestra, con ejemplo una aplicación concreta de la regla 9.

3.4.4 Identificando Compuertas BPMN en Mockups

Las "compuertas" son elementos de modelado que controlan cómo el proceso de negocio diverge o converge. Es decir que representan puntos de control para los caminos dentro de los procesos de negocio. Dividen y unifican el flujo del proceso de negocio [21].

Cuando analizamos un Mockup se pueden observar situaciones en las que el usuario que está interactuando con el Mockup debe tomar alguna decisión. Por ejemplo, si dentro de un Mockup existiera un botón llamado "Cancelar", la acción de presionar dicho botón implicaría la decisión de abandonar la interacción con el Mockup. Otra situación de decisión se puede dar cuando, por ejemplo, se depende de algún valor. Dicho valor puede determinar que se debe continuar por un camino o por otro dentro del mismo Mockup. Las situaciones mencionadas pueden ser representadas dentro de un proceso de negocio y anotadas sobre un Mockup. Desde EUG no se pudo identificar una anotación que permita describir compuertas de decisión BPMN. Sin embargo, desde la literatura vinculada al procesamiento del lenguaje natural para la identificación de elementos BPMN se menciona que existen oraciones que permiten identificar compuertas de decisión. Aquellos párrafos que contienen oraciones que incluyen palabras de señalización del tipo exclusivas o inclusivas, básicamente hacen referencia a un flujo de control [7]. Estas palabras se dividen en dos grupos que nos permiten identificar dos tipos de compuertas.

• Compuertas exclusivas (XOR): palabras de señalización que se refieren a la exclusión como, por ejemplo, "si", "en caso de que", "cuándo", "solo sí", etc. Se aclara además que para éste tipo de compuerta se utiliza el recurso lingüístico denominado "conector de consecuencia[10]".

• Compuertas paralelas (AND): palabras que hacen referencia al paralelismo como, por ejemplo, "mientras", "en paralelo", "al mismo tiempo", "simultáneamente", etc.

Es decir que si se analizan detenidamente estos tipos de párrafos es posible extraer del mismo la compuerta de decisión que contiene.

Este tipo de párrafos se adoptaron para el desarrollo de la presente tesis.

Para identificar "Compuertas exclusivas (XOR)", la regla que proponemos es la siguiente (ver tabla 19):

[10] Los conectores son las palabras o expresiones que permiten señalar una relación entre dos oraciones o enunciados. El uso de conectores favorece la lectura y la comprensión de textos ya que aportan coherencia y cohesión. Los conectores de consecuencia se encargan de unir diferentes oraciones mostrando el efecto de aquello que se ha explicado previamente y marcando una relación de causa y efecto entre dos o más oraciones.
Ejemplos: como consecuencia, como resultado, entonces, luego, por consiguiente.

Regla	Descripción	Oración Ejemplo	Compuerta de exclusión XOR
Regla 10	[Palabra de exclusión] + "[Actividad/Evento]", + [Conector de consecuencia], + navegar hacia "[Actividad/Evento]". Sino, navegar hacia "[Actividad/Evento]".	Si *[Palabra de exclusión]*, "el sistema identifica al cliente *[Actividad/Evento]*", entonces *[Conector de consecuencia]*, navegar hacia "Mockup, el cliente trabaja con el carro de compra, vista (5) *[Actividad/Evento]*". Sino, navegar hacia "el cliente escribe su correo electrónico *[Actividad/Evento]*". (figura 15)	¿el sistema identifica al cliente? (ver figura 31)

Tabla 19 – Regla 10 de mapeo para identificar "Compuerta XOR"

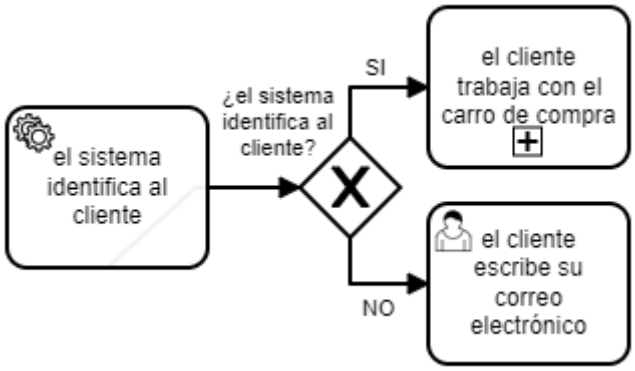


Figura 31 – Compuerta de exclusión XOR: "*¿el sistema identifica al cliente?*"

De la Regla precedente lo primero que observamos es que la *[Actividad/Evento]* que sigue a la *[Palabra de exclusión]* se convierte en pregunta. Para el contexto de EUGEBP, entonces proponemos que las "Compuertas de exclusión XOR" se representen como preguntas. También, proponemos que la/el primer *[Actividad/Evento]* de la regla se conecte a la *"Compuerta de exclusión XOR"* identificada. En este sentido de la propuesta surge la idea de que el origen es la primer *[Actividad/Evento]* y el destino es la *"Compuerta de exclusión XOR"*.

Además de identificar el elemento BPMN "Compuerta de exclusión XOR" es posible extraer los elementos BPMN a los que se conecta la compuerta. Siguiendo con ejemplo mostrando anteriormente, la salida positiva (el SI) se conecta al elemento BPMN destino del tipo "Actividad": "*el cliente trabaja con el carro de compra*". Y, la salida negativa (el NO) se conecta al elemento BPMN destino del tipo "Actividad": "*el cliente escribe su correo electrónico*". Luego vamos a profundizar las propuestas mencionadas.

Por otro lado, para identificar "Compuertas paralelas (AND)", la regla que utilizamos es la siguiente (ver tabla 20):

Regla	Descripción	Oración Ejemplo	Compuerta de paralelismo AND
Regla 11	[Palabra de paralelismo] + "[Actividad/Evento]", + [Palabra de paralelismo], + "[Actividad/Evento]".	Mientras que*[Palabra de paralelismo]* "el empleado de recepción trabaja con la vista de las compras *[Actividad/Evento]*", en paralelo *[Palabra de paralelismo]*, "el empleado de depósito trabaja con la vista del stock de productos *[Actividad/Evento]*".	Mientras que "el empleado de recepción trabaja con la vista de las compras", en paralelo, "el empleado de depósito trabaja con la vista del stock de productos" (ver figura 32)

Tabla 20 – Regla 11 de mapeo para identificar "Compuerta AND"

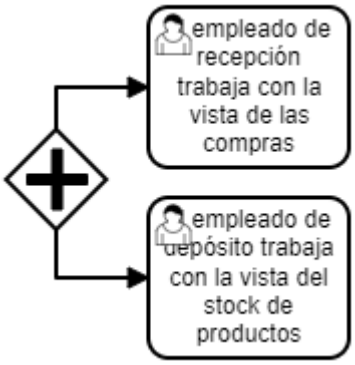


Figura 32 – Compuerta de paralelismo AND

De la regla precedente, además de identificar el elemento BPMN "Compuerta de paralelismo AND" es posible identificar los elementos BPMN a los que se conecta la compuerta. Para el ejemplo mostrando anteriormente, una salida se conecta al elemento BPMN destino del tipo "Actividad": "*el empleado de recepción trabaja con la vista de las compras*". Y, otra salida se conecta al elemento BPMN destino del tipo "Actividad": "*el empleado de depósito trabaja con la vista del stock de productos*".

Es decir que las Compuertas BPMN implícitamente permiten identificar "Flujos de Secuencia" o "Flujos de Mensaje" (dependiendo del contexto) representado por los elementos BPMN destino a los que se conectan las Compuertas BPMN. Se va a analizar con mayor profundidad en la siguiente sección éste tema.

Recordar que las formas de identificar a una "Actividad" o a un "Evento" ya fueron definidas precedentemente y nos valemos de las mismas para identificar las "Compuerta de Decisión".

3.4.5 Identificando Conectores BPMN en Mockups

Los conectores vinculan dos objetos de flujo de un diagrama BPMN. En la presente tesis nos enfocamos en los siguientes tipos de conectores:
• Flujo de Secuencia: define el orden de los objetos de flujo (Actividades, Eventos y Compuertas) en un proceso. Ordena los mismos y crea los caminos del proceso que son navegados durante su ejecución. El origen y el destino de la línea del flujo de secuencia (el destino es manifestado por la punta de flecha en la línea) solo pueden conectar objetos de flujo BPMN.
• Flujo de Mensaje: define el flujo de comunicación entre dos participantes o entidades diferentes representados como pools (por ejemplo, una organización y sus proveedores). El objeto de la

comunicación es un mensaje. El flujo de mensajes debe darse entre dos pools separados y no puede conectar dos objetos dentro de un mismo pool [21].

De la definición de "Flujo de Secuencia" podemos afirmar que todos los objetos de flujo (Actividad, Evento o Compuerta) pueden conectarse entre sí. Es decir que una vez que se identificó un objeto de flujo en particular el siguiente paso natural es determinar a que otro objeto de flujo se conecta.

Si analizamos las interacciones proporcionadas por un Mockup, cada vez que un usuario termina de realizar una acción y se dirige a realizar otra, podemos identificar un flujo de secuencia que tiene un origen (la actividad de usuario que se terminó de realizar) y un destino (la actividad de usuario que se va a realizar). En este mismo contexto, cada vez que un usuario presiona un botón o un link dentro de un Mockup y se dirige hacia otro, nuevamente podemos identificar un flujo de secuencia que tiene un origen (el Mockup que se está abandonando) y un destino (el nuevo Mockup con el que se desea trabajar). Recordar que de acuerdo a lo definido hasta el momento un Mockup representa una actividad de usuario del tipo sub-proceso. Dichos "Flujos de secuencia", están implícitos y pueden ser aprovechados para nuestro propósito (ver figura 33).

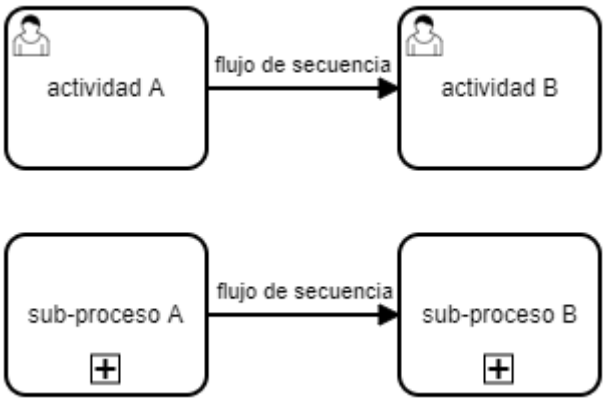


Figura 33 – Flujos de Secuencia BPMN

El patrón EUG "Navegar hacia (Navigates to destination)" nos da un indicio de ir hacia otro lugar. Es decir que nos indica un camino a tomar, una orientación o un sentido. A través del patrón "Navegar hacia" podemos identificar el siguiente concepto: "Desde un [origen] hacia un [destino]"
Donde un [origen] y un [destino], para el ámbito de los Mockups, pueden ser:
• Una actividad de usuario, o
• Una actividad de servicio.

Entonces, desde este patrón podemos proponer la siguiente regla (ver tabla 21):

Regla	Descripción	Oración Ejemplo	Flujo de Secuencia o de Mensaje
Regla 12	Navegar hacia "[Actividad]".	Navegar hacia "Mockup "El cliente registra sus datos" vista (2) *[Actividad]*". (figura 15)	Origen: es el Mockup donde se identifica la regla. Es decir, *el cliente se identifica con el sistema*. Destino: es la [Actividad] de la regla. Es decir, *el cliente registra sus datos.* (ver figura 34)

Tabla 21 – Regla 12 de mapeo para identificar "Flujo de Secuencia o de Mensaje"

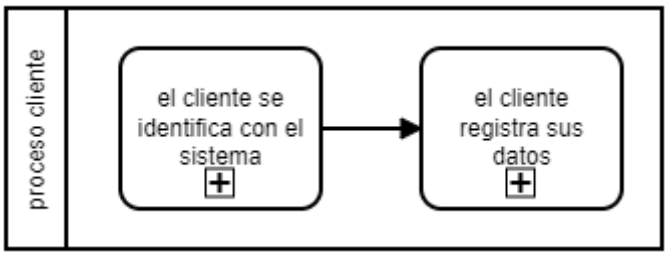


Figura 34 – Flujo de Secuencia identificado

Notar que la regla *Navegar hacia "[Actividad]"* permite identificar un *Flujo de Secuencia* o un *Flujo de Mensaje*. La identificación de uno u otro depende de la *Actividad* contenida en el destino. Si la *Actividad* es responsabilidad de una entidad interna que está representada dentro de un *Lane* entonces el flujo identificado es un *Flujo de Secuencia*. Si la *Actividad* es responsabilidad de una entidad externa que está representada dentro de un *Pool* diferente al del contexto del origen entonces el flujo identificado es un *Flujo de Mensaje*.

En el ejemplo de la regla, la siguiente oración:
Navegar hacia "Mockup "El cliente registra sus datos" vista (2) *[Actividad]*".
Permite identificar un *Flujo de Secuencia* ya que el contexto de las actividades vinculadas al flujo pertenece a la entidad interna denominada "cliente".

Luego, si analizamos el siguiente ejemplo:
Navegar hacia "el sistema del medio de pago procesa los datos enviados *[Actividad]*".
Permite identificar un *Flujo de Mensaje* ya que el contexto de las actividades vinculadas al flujo pertenece a una entidad externa llamada "sistema del medio de pago" (ver figura 35).

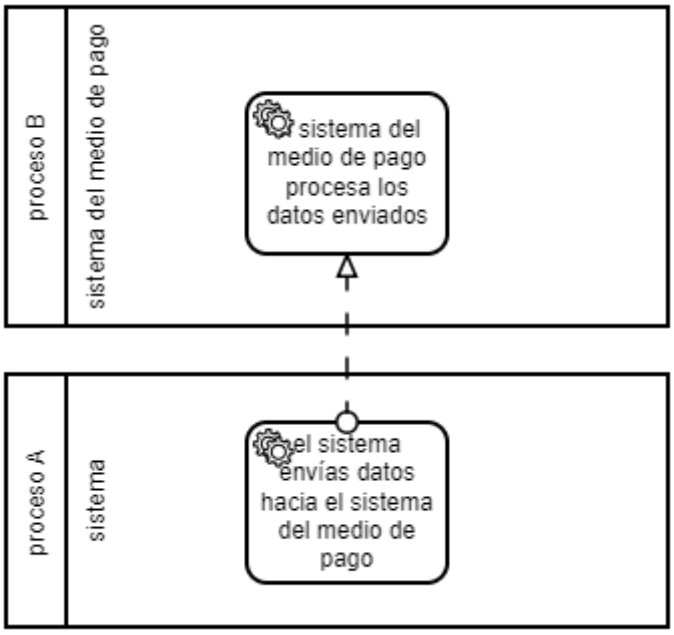


Figura 35 – Flujo de Mensaje identificado desde la oración: *Navegar hacia "el sistema del medio de pago procesa los datos enviados"*

Observación: Para el contexto EUGEBP la palabra reservada "Navegar hacia" implica que la anotación permite representar un Flujo BPMN del tipo "de Mensaje" o "de Secuencia".

Por otro lado, para identificar la secuencia del proceso de negocio, además necesitamos de algún tipo de anotación que nos permita indicar (siempre cuando sea necesario) en qué secuencia o cronología

se deben suceder los elementos de procesos de negocio identificados. Es decir que necesitamos de algún tipo de anotación que nos permita explicitar el orden en el que deben fluir los mismos. Para identificar la secuencia del proceso de negocio, las investigaciones vinculadas al procesamiento del lenguaje natural sugieren recurrir a los adjetivos ordinales[11]. Básicamente para ayudar a encontrar el orden de secuencia del proceso de negocio podemos realizar anotaciones del tipo "primero,", "segundo,", etc. Es decir que es posible utilizar "adjetivos ordinales" dentro de las anotaciones para identificar el orden. Este concepto se adoptó en la presente tesis para colaborar con el objetivo de la misma. Entonces, los adjetivos ordinales, nos permiten identificar el orden en el que se debe ejecutar de proceso de negocio.
Entonces, para identificar Flujos de Secuencia o de Mensaje, proponemos lo siguiente:

- Si el adjetivo ordinal de la anotación es de orden 1 (por ejemplo. "*Primero, el cliente escribe su correo electrónico*"), entonces, el origen es "el Evento Inicial BPMN" del Mockup donde se identifica la anotación. Para el ejemplo anterior el origen es el evento "*el cliente presionó el botón con la imagen de acceso*". Y, el destino es la Actividad inferida desde la anotación, es decir "*el cliente escribe su correo electrónico*".
- Si el adjetivo ordinal de la anotación es de orden mayor a 1 (por ejemplo, "*Segundo, el cliente escribe su contraseña*"), entonces el origen es el "Objeto de Flujo" que posee el adjetivo ordinal de orden anterior. Para el ejemplo precedente, el origen es la Actividad "*el cliente escribe su correo electrónico*". Y, el destino es la Actividad inferida desde la anotación, es decir "e*l cliente escribe su contraseña*".

Desde la idea precedente surge la siguiente regla (ver tabla 22):

Regla	Descripción	Oración Ejemplo	Flujo de Secuencia o de Mensaje
Regla 13	[Adjetivo ordinal], + [Actividad / Evento / Compuerta]	Primero *[Adjetivo ordinal]*, el cliente escribe su número de medio de pago *[Actividad]*. (figura 17)	Origen: Evento "*el cliente presionó el botón con la imagen de compra*". Destino: Actividad "e*l cliente escribe su número de medio de pago*". (ver figura 36)

Tabla 22 – Regla 13 de mapeo para identificar "Flujo de Secuencia o de Mensaje"

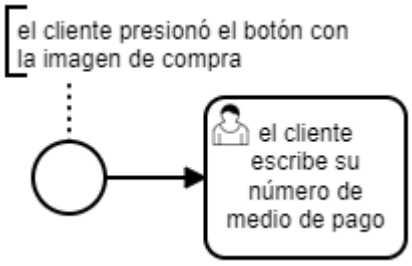


Figura 36 – Flujo de Secuencia identificado desde la oración: "*Primero, el cliente escribe su número de medio de pago*"

Notar nuevamente que la regla *[Adjetivo ordinal], + [Actividad / Evento / Compuerta]* permite identificar un *Flujo de Secuencia* o un *Flujo de Mensaje*. La identificación de uno u otro depende de la *Actividad*/*Evento*/*Compuerta* contenida en el destino. Si la *Actividad*/*Evento*/*Compuerta* es responsabilidad de una entidad interna que está representada dentro de un *Lane* entonces el flujo

[11] Los adjetivos ordinales son aquellos que se refieren a las posiciones numeradas. Se utilizan para designar una posición numérica. Por ejemplo: Primero llegué yo. Segundo llegaste vos. Adjetivos ordinales del 1 al 10: primero, segundo, tercero, cuarto, quinto, sexto, séptimo, octavo, noveno, décimo.

identificado es un *Flujo de Secuencia*. Si la *Actividad*/*Evento*/*Compuerta* es responsabilidad de una entidad externa que está representada dentro de un *Pool* diferente al del contexto del origen entonces el flujo identificado es un *Flujo de Mensaje*.

Por último, ya habíamos mencionado que las Compuertas BPMN permiten identificar implícitamente "Flujos de Secuencia" o "Flujos de Mensaje" de acuerdo al contexto en el que se encuentran las mismas. A continuación, hacemos explícitas las observaciones mencionadas en la sección anterior.

La Regla 10 proveniente de EUGEBP permite identificar, además, por lo menos tres "Flujos" (de Secuencia o de Mensaje):

- En el primer "Flujo" el origen es la [Actividad/Evento] que sigue a la [Palabra de exclusión] y el destino es la "Compuerta de exclusión XOR" identificada.
- En el segundo "Flujo" el origen es la "Compuerta de exclusión XOR" identificada y el destino es la [Actividad/Evento] de la salida positiva (el SI) a la cual se conecta la compuerta.
- En el tercer "Flujo" el origen es la "Compuerta de exclusión XOR" identificada y el destino es la [Actividad/Evento] de la salida negativa (el NO) a la cual se conecta la compuerta.

En ésta sección hacemos explícita dichas observaciones y la representamos a través de la siguiente regla (ver tabla 23):

Regla	**Descripción**	**Oración Ejemplo**	**Flujo de Secuencia o de Mensaje**
Regla 14	[Palabra de exclusión] + "[Actividad/Evento]", + [Conector de consecuencia], + navegar hacia "[Actividad/Evento]". Sino, navegar hacia "[Actividad/Evento]".	Si *[Palabra de exclusión]*, "el sistema identifica al cliente *[Actividad/Evento]*", entonces *[Conector de consecuencia]*, navegar hacia "Mockup, el cliente trabaja con el carro de compra, vista (5) *[Actividad/Evento]*". Sino, navegar hacia "el cliente escribe su correo electrónico *[Actividad/Evento]*". (figura 15)	Origen: Actividad "*el sistema identifica al cliente*". Destino: Compuerta XOR "*¿el sistema identifica al cliente?*". Origen: Compuerta XOR "*¿el sistema identifica al cliente?*". Destino: Actividad de la salida positiva, "*el cliente trabaja con el carro de compra*. Origen: Compuerta XOR "*¿el sistema identifica al cliente?*". Destino: Actividad de la salida negativa, "*el cliente escribe su correo electrónico*" (ver figura 37)

Tabla 23 – Regla 14 de mapeo para identificar "Flujo de Secuencia o de Mensaje"

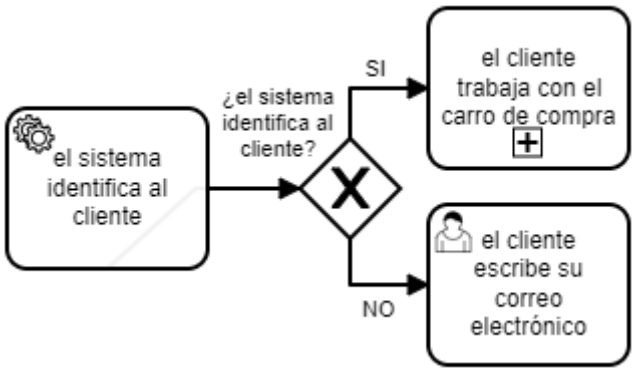


Figura 37 – Flujos de Secuencia identificados

A continuación, se presentan todas las de reglas de mapeo identificadas y a las mismas las denominamos “Reglas de redacción EUGEBP”.

3.5 Reglas de redacción EUGEBP

En ésta sección se formaliza el conjunto de reglas estructuradas diseñadas para guiar el proceso de producción de anotaciones sobre los Mockups. Cada anotación es una definición coloquial estructurada legible para los stakeholders. El mismo no presenta ningún concepto técnico que pueda limitar su entendimiento y además permite la identificación y derivación de elementos BPMN utilizando el lenguaje natural de las personas. A continuación, presentamos el conjunto de reglas de redacción EUGEBP (ver tabla 24):

Nº	**Descripción de la Regla**	**Ejemplo**	**Elemento BPMN que permite identificar**
1	El sujeto de una oración.	El cliente *[sujeto]* presionó el botón con la imagen del carro.	**SwimLane**: cliente
2	El objeto indirecto de una oración.	El sistema envió datos hacia el sistema del medio de pago *[objeto indirecto]*.	**SwimLane**: sistema del medio de pago
3	El/La [Clase] tiene un [Atributo].	El medio de pago *[Clase]* tiene un número *[Atributo]*.	**SwimLane**: medio de pago
4	[Sujeto] + [Verbo en tiempo presente] + [Objeto].	El cliente *[Sujeto]* escribe *[Verbo en tiempo presente]* su número de medio de pago *[Objeto]*.	**Actividad de usuario**: el cliente escribe su número de medio de pago.
5	Mockup, [Sujeto] + [Verbo en tiempo presente] + [Objeto] , vista (número).	Mockup, el cliente *[Sujeto]* trabaja *[Verbo en tiempo presente]* con el carro de compra *[Objeto]*, vista (5) .	**Sub-proceso**: el cliente trabaja con el carro de compra.
6	E/el sistema [Verbo en tiempo presente] + [Objeto].	El sistema identifica *[Verbo en tiempo presente]* al cliente *[Objeto]*.	**Actividad de servicio**: el sistema identifica al cliente.
7	[Sujeto] + [Verbo en pretérito perfecto simple] + [Objeto].	El empleado de depósito *[Sujeto]* presionó *[Verbo en pretérito perfecto simple]* el botón con la imagen de lista *[Objeto]*.	**Evento**: el empleado de depósito presionó el botón con la imagen de lista.

8	Trigger: [Sujeto] + [Verbo en pretérito perfecto simple] + [Objeto].	Trigger: El cliente *[Sujeto]* presionó *[Verbo en pretérito perfecto]* el botón con la imagen de compra *[Objeto]*.	**Evento inicial**: el cliente presionó el botón con la imagen de compra.
9	"Objeto de Flujo", del "Flujo Principal", sin "Flujo de Salida"	el sistema registra la compra *[Objeto de Flujo]*	**Evento final** se infiere desde el ultimo objeto de flujo sin flujo de salida.
10	[Palabra de exclusión] + "[Actividad/Evento]", + [Conector de consecuencia], + navegar hacia "[Actividad/Evento]". Sino, navegar hacia "[Actividad/Evento]".	Si *[Palabra de exclusión]*, "el sistema identifica al cliente *[Actividad/Evento]*", entonces *[Conector de consecuencia]*, navegar hacia "Mockup, el cliente trabaja con el carro de compra, vista (5) *[Actividad/Evento]*". Sino, navegar hacia "el cliente escribe su correo electrónico *[Actividad/Evento]*".	**Compuerta XOR**: ¿el sistema identifica al cliente?
11	[Palabra de paralelismo] + "[Actividad/Evento]", + [Palabra de paralelismo], + "[Actividad/Evento]".	Mientras que *[Palabra de paralelismo]* "el empleado de recepción trabaja con la vista de las compras *[Actividad/Evento]*", en paralelo *[Palabra de paralelismo]*, "el empleado de depósito trabaja con la vista del stock de productos *[Actividad/Evento]*".	**Compuerta AND**: Mientras que "el empleado de recepción trabaja con las vista de las compras", en paralelo, "el empleado de depósito trabaja con la vista del stock de productos".
12	Navegar hacia "[Actividad]".	Navegar hacia "Mockup "El cliente registra sus datos" vista (2) *[Actividad]*".	**Flujo de Secuencia**. **Origen**: **Sub-proceso,** *el cliente se identifica con el sistema.* **Destino**: **Sub-proceso,** *el cliente registra sus datos.*
	Navegar hacia "[Actividad]".	Navegar hacia "El sistema del medio de pago procesa los datos enviados *[Actividad]*".	**Flujo de Mensaje**. **Origen**: **Actividad de usuario,** *el cliente presiona el botón para pagar la compra.* **Destino**: **Actividad de servicio**, *el sistema del medio de pago procesa los datos enviados.*
13	[Adjetivo ordinal], + [Actividad / Evento / Compuerta].	Primero *[Adjetivo ordinal]*, el cliente escribe su número de medio de pago *[Actividad]*.	**Flujo de Secuencia**. **Origen**: **Evento inicial,** *el cliente presionó el botón con la imagen de compra.* **Destino**: **Actividad de usuario,** *el cliente escribe su número de medio de pago.*
14	[Palabra de exclusión] + "[Actividad/Evento]", + [Conector de	Si *[Palabra de exclusión]*, "el sistema identifica al cliente *[Actividad/Evento]*", entonces *[Conector de consecuencia]*, navegar hacia "Mockup, el	**Flujos de Secuencia**. **Origen**: **Actividad de servicio,** *al sistema identifica al cliente.* **Destino**: **Compuerta XOR,** *¿el sistema identifica al cliente?*

	consecuencia], + navegar hacia "[Actividad/Evento]". Sino, navegar hacia "[Actividad/Evento]".	cliente trabaja con el carro de compra, vista (5) *[Actividad/Evento]*". Sino, navegar hacia "el cliente escribe su correo electrónico *[Actividad/Evento]*".	**Origen**: **Compuerta XOR,** *¿el sistema identifica al cliente?* **Destino**: **Sub-proceso,** *el cliente trabaja con el carro de compra.* **Origen**: **Compuerta XOR,** *¿el sistema identifica al cliente?* **Destino**: **Actividad de usuario,** *el cliente escribe su correo electrónico.*

Tabla 24 – Reglas de redacción EUGEBP

3.5.1 Guías para la producción de anotaciones EUGEBP

Las anotaciones que se realicen sobre los Mockups se deber escribir de acuerdo a las "Reglas de Redacción EUGEBP". Sin embargo, para colaborar con el propósito de identificar elementos BPMN dentro de oraciones en lenguaje natural (de forma manual o automática), es importante tener presente además las siguientes guías al momento de realizar anotaciones sobre los Mockups:

- Las anotaciones deben ser gramaticalmente correctas. Es decir que dichas anotaciones deben estar compuestas por oraciones que respetan las reglas sintácticas y semánticas del idioma en cuestión (en nuestro caso el español). Esto ayuda al análisis automático del texto mediante analizadores de oraciones inteligentes (herramientas NLP). Si la oración es gramaticalmente correcta es fácil determinar los elementos que componen una oración.
- Las anotaciones deben estar compuestas por "oraciones simples". Cada oración debe estar compuesta por las categorías más generales de la "oración simple": el "Sujeto", el "Verbo" y el "Objeto". Dichas categorías existen como unidades sintácticas comprensibles para cualquier ser humano[54].
- Las oraciones producidas por el analista y el stakeholder deben ser cortas y sencillas. Esto permite no sobrecargar el Mockup con anotaciones y además facilita la lectura y comprensión de las mismas.
- Los adjetivos ordinales (primero, segundo, etc.) solo deben ser utilizados en aquellos casos en los que no sea fácil identificar el flujo natural del proceso de negocio. Esto también ayuda a no sobrecargar de anotaciones EUGEBP el Mockup y a la lectura de los mismos.
- El tiempo del verbo de cada oración debe estar en "presente" o "pasado (pretérito prefecto simple)". Las oraciones deben estar redactadas siguiendo el criterio mencionado.
- Desde el punto de vista estilístico, cada oración debe comenzar con mayúscula y terminar con un punto.
- Con respecto a la intención del hablante, cada oración debe ser enunciativa afirmativa. Es decir que debe informar de un hecho objetivamente afirmando el mismo.
- De acuerdo con la naturaleza del predicado, la oración debe ser predicativa activa. Es decir que el núcleo del predicado es un verbo predicativo[12] y el sujeto realiza la acción.

Todas las características mencionadas suponen la estructura más simple de las oraciones, ya que no pertenecen a ninguna unidad gramatical principal. Esto significa que este tipo de oraciones son simples y comprensibles para cualquier ser humano.

Todas las anotaciones realizadas sobre los Mockups del ejemplo fueron escritas de acuerdo a las características enunciadas precedentemente.

[12] Los verbos predicativos son aquellos que expresan ESTADO, ACCIÓN o PASIÓN del sujeto al que se refieren. Ejemplo: Pedro corta leña.

Ejemplos de correcta redacción de anotaciones EUGEBP:
El cliente tiene un correo electrónico.
El cliente escribe su contraseña.
El medio de pago tiene un número.
El sistema identifica al cliente.

Estos estilos de redacción de oraciones EUGEBP ayudan a la identificación de los elementos BPMN a través del uso de herramientas NLP como por ejemplo Stanford CoreNLP.

3.6 Técnica EUGEBP

En esta sección, presentamos el enfoque propuesto por la tesis denominado "End User Grammar Extended for Business Processes - Gramática de Usuario Final Extendida para Procesos de Negocios" o, de acuerdo a sus siglas EUGEBP.

En esencia, la técnica EUGEBP se apoya en un conjunto de reglas vernáculas estructuradas diseñadas para guiar el proceso de anotación de oraciones sobre los Mockups. Dicho conjunto de reglas está basado en el lenguaje natural y en el uso de oraciones simples gramaticalmente correctas. Es decir que utilizar dicho conjunto de reglas en la producción de anotaciones sobre los Mockups hace que las mismas sean fácilmente entendibles para un usuario final. Al mismo tiempo dichas anotaciones poseen una fuerte estructura que permite enriquecer la definición de los Mockups.

Por otro lado, luego de haber producido anotaciones EUGEBP sobre los Mockups, es posible identificar, extraer y relacionar los Elementos BPMN contenidos en los mismas. Una vez que se poseen los Elementos BPMN y sus relaciones, inferir los procesos de negocios que contienen dichos elementos es factible. En éste contexto la técnica EUGEBP, además, fue diseñada para facilitar la derivación de los "Procesos de Negocio" desde las anotaciones EUGEBP.

De acuerdo a los dos párrafos anteriores podemos definir entonces a la técnica EUGEBP cómo un conjunto de reglas estructuradas diseñadas para:

i) guiar el proceso de producción de anotaciones sobre los Mockups
ii) facilitar el proceso de derivación de los Diagramas BPMN

La técnica EUGEBP explota las anotaciones realizadas sobre los Mockups en la fase de obtención de requerimientos. En dicha fase los analistas no solo validan ideas usando Mockups, sino que además mejoran dichos documentos con oraciones simples que son naturales para los stakeholders. La propuesta de la tesis apunta a aprovechar al máximo los Mockups, a través de la técnica EUGEBP, proporcionando un plano de formalización. De esta manera, los Mockups dejan de ser un activo desechable, y se convierten en un documento formal con información que se puede extraer manualmente (el análisis de las anotaciones las realiza el analista) o automáticamente (el análisis de las anotaciones se realiza utilizando herramientas NLP). Si bien a menudo los Mockups se desechan una vez que comienza la fase de desarrollo, en éste enfoque los usamos para documentar los requerimientos e inferir procesos de negocios.

3.6.1 Pasos EUGEBP

Para aplicar la técnica EUGEBP dentro del proceso de elicitación de elicitación de requerimientos de software concretamente se propone en la tesis que el equipo de desarrollo trabaje en los siguientes pasos (ver figura 38):

a) Proceso de Producción de Anotaciones EUGEBP
b) Proceso de División de Anotaciones EUGEBP
c) Proceso de Extracción de Elementos BPMN
d) Proceso de Identificación del Flujo Normal

En la figura 38 se muestran los cuatros pasos que conforman la técnica EUGEBP.

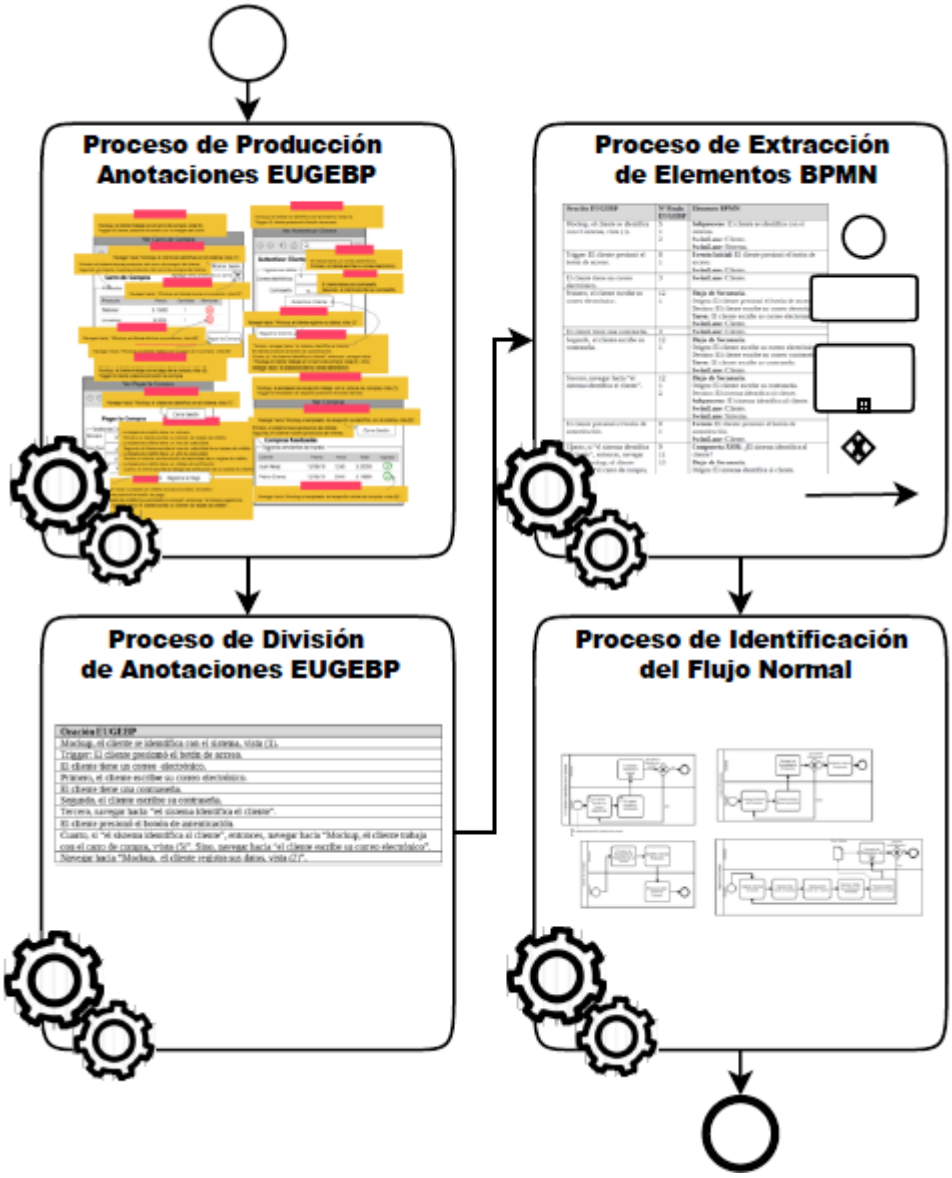


Figura 38 - Pasos EUGEBP

La realización de cada uno de los pasos mencionados es manual (algunos pasos pueden automatizarse a través del uso de herramientas NLP) y es llevada a cabo por el equipo de analistas. A continuación, se describe cada uno de ellos.

Para ejemplificar los pasos EUGEBP vamos a tomar el Mockup de la Figura 17: *"Mockup, el cliente trabaja con el pago de la compra, vista (6)."* como referencia.

3.6.1.1 Proceso de Producción de Anotaciones EUGEBP

La técnica EUGEBP permite introducir anotaciones (oraciones), sobre los Mockups, utilizando el lenguaje natural de los stakeholders. Esto implica que dichas anotaciones son fáciles de entender y no representan ningún concepto técnico que pueda limitar la comprensión de los usuarios finales. En este contexto los analistas, junto con los stakeholders, pueden escribir sobre los Mockups oraciones guiándose o utilizando las "Reglas de redacción EUGEBP".

El "*Proceso de Producción Anotaciones EUGEBP*" consiste en realizar anotaciones, sobre los Mockups, utilizando el conjunto de "Reglas de redacción EUGEBP". Es decir que dichas anotaciones deben ser oraciones simples, gramaticalmente correctas, que respetan las reglas sintácticas y semánticas del idioma. Esto facilita el análisis posterior (manual del analista o automático a través del uso de herramientas NLP) del texto.

Una vez que el proceso de "*Proceso de Producción Anotaciones EUGEBP*" sobre los Mockups ha finalizado equipo de analistas está en condiciones analizar en profundidad las oraciones producidas.

El resultado de éste paso es el Mockup comentado con anotaciones EUGEBP (ver figura 17).

3.6.1.2 Proceso de División de Anotaciones EUGEBP

Luego de que el equipo de analistas ha realizado, junto con los stakeholders, anotaciones sobre los Mockups el siguiente paso es procesar dichas anotaciones.

El "*Proceso de División de Anotaciones EUGEBP*" Consiste en juntar todo el texto conformado por las anotaciones realizadas sobre los Mockups y separar el mismo en párrafos u oraciones independientes. Básicamente en éste paso juntamos todo el texto producido en el paso anterior y lo dividimos en anotaciones, de acuerdo al mapeo (con las reglas de redacción) con el que se corresponda..

El resultado de éste paso, de acuerdo al ejemplo de la Figura 17, es una tabla con todas las anotaciones EUGEBP encontradas. Ver tabla 25.

Oración EUGEBP
Mockup, el cliente trabaja con el pago de la compra, vista (6).
Trigger: el cliente presionó el botón con la imagen de compra.
Navegar hacia “Mockup, el cliente se identifica con el sistema, vista (1)”
El medio de pago tiene un número.
Primero, el cliente escribe el número del medio de pago.
El medio de pago tiene un mes de caducidad.
Segundo, el cliente escribe el mes de caducidad del medio de pago.
El medio de pago tiene un año de caducidad.
Tercero, el cliente escribe el año de caducidad del medio de pago.
El medio de pago tiene un código de verificación.
Cuarto, el cliente escribe el código de verificación del medio de pago.
Quinto, el cliente presiona el botón con la imagen de pago.
Sexto, el sistema envía datos hacia el sistema del medio de pago.
Séptimo, navegar hacia “el sistema del medio de pago procesa los datos enviados”.
Octavo, el sistema valida el pago.
Noveno, si, “el sistema valida el pago”, entonces navegar hacia “el sistema registra la compra”. Sino, navegar hacia “el cliente escribe el número del medio de pago”.

Tabla 25 – Resultado de la división de las anotaciones EUGEBP en oraciones

3.6.1.3 Proceso de Extracción de Elementos BPMN

Una vez que las anotaciones EUGEBP fueron desglosadas, es decir divididas en oraciones independientes, el equipo analiza cada oración.

El “*Proceso de Extracción de Elementos BPMN*” consiste en identificar y extraer los elementos BPMN desde cada anotación EUGEBP encontrada. Para ello el equipo extrae la oración simple, en lenguaje natural. Luego, la contrasta contra la/s “Regla/s de Redacción EUGEBP” que le corresponda. Y, a continuación, identifica y extrae el o los elementos BPMN que dicte la regla.

Entonces, el equipo analiza oración por oración y extrae manualmente el o los elementos BPMN que contiene cada una de las mismas.

Para realizar la tarea mencionada, de forma manual. utilizamos una tabla de ayuda (Ver tabla 26) que contiene las siguientes columnas:

- la anotación
- la regla de anotación EUGEBP a la que pertenece
- el elemento BPMN que nos permite identificar o extraer

Oración EUGEBP	**Nº Regla EUGEBP**	**Elemento BPMN**
Mockup, el cliente trabaja con el pago de la compra, vista (6).	5 1 2	**Sub-proceso**: el cliente trabaja con el pago de la compra. **SwimLane**: cliente.
Trigger: el cliente presionó el botón con la imagen de compra.	8 1	**Evento Inicial**: el cliente presionó el botón con la imagen de compra. **SwimLane**: cliente.

Navegar hacia “Mockup, el cliente se identifica con el sistema, vista (1)”	12	**Flujo de Secuencia o de Mensaje.** Origen: **Sub-Proceso**, el cliente trabaja con el pago de la compra. Destino: **Sub-Proceso,** el cliente se identifica con el sistema.
El medio de pago tiene un número.	3	**SwimLane**: medio de pago.
Primero, el cliente escribe el número del medio de pago.	13 1	**Flujo de Secuencia o de Mensaje.** Origen: **Evento Inicial,** el cliente presionó el botón con la imagen de compra. Destino: **Actividad de usuario**, el cliente escribe el número del medio de pago. **SwimLane**: cliente.
El medio de pago tiene un mes de caducidad.	3	**SwimLane**: medio de pago.
Segundo, el cliente escribe el mes de caducidad del medio de pago.	13 1 2	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de usuario**, el cliente escribe el número del medio de pago. Destino: **Actividad de usuario**, el cliente escribe el mes de caducidad del medio de pago. **SwimLane**: cliente. **SwimLane**: medio de pago.
El medio de pago tiene un año de caducidad.	3	**SwimLane**: medio de pago.
Tercero, el cliente escribe el año de caducidad del medio de pago.	13 1 2	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de usuario**, el cliente escribe el mes de caducidad del medio de pago. Destino: **Actividad de usuario**, el cliente escribe el año de caducidad del medio de pago. **SwimLane**: cliente. **SwimLane**: medio de pago.
El medio de pago tiene un código de verificación.	3	**SwimLane**: medio de pago.
Cuarto, el cliente escribe el código de verificación del medio de pago.	13 1 2	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de usuario**, el cliente escribe el año de caducidad del medio de pago. Destino: **Actividad de usuario**, el cliente escribe el código de verificación del medio de pago. **SwimLane**: cliente. **SwimLane**: medio de pago.
Quinto, el cliente presiona el botón con la imagen de pago.	13 1	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de usuario**, el cliente escribe el código de verificación del medio de pago. Destino: **Actividad de usuario**, el cliente presiona el botón con la imagen de pago. **SwimLane**: cliente.
Sexto, el sistema envía datos hacia el sistema del medio de pago.	13 1 2	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de usuario**, el cliente presiona el botón con la imagen de pago. Destino: **Actividad de servicio**, el sistema envía datos hacia el sistema del medio de pago. **SwimLane**: sistema.

Séptimo, navegar hacia "el sistema del medio de pago procesa los datos enviados".	13 1 6	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de servicio**, el sistema envía datos hacia el sistema del medio de pago. Destino: **Actividad de servicio**, el sistema del medio de pago procesa los datos enviados. **SwimLane**: sistema del medio de pago.
Octavo, el sistema valida el pago.	13 1 6	**Flujo de Secuencia o de Mensaje.** Origen: **Actividad de servicio** el sistema del medio de pago procesa los datos enviados. Destino: **Actividad de servicio**, el sistema valida el pago. **SwimLane**: sistema. **Actividad de Servicio**: el sistema valida el pago.
Noveno, si, "el sistema valida el pago", entonces navegar hacia "el sistema registra la compra". Sino, navegar hacia "el cliente escribe el número del medio de pago".	10 11 13	**Compuerta XOR**: ¿el sistema valida el pago? **Flujo de Secuencia o de Mensaje.** Origen: **Actividad de servicio**, el sistema valida el pago. Destino: **Compuerta XOR**: ¿el sistema valida el pago? **Flujo de Secuencia o de Mensaje.** Origen: **Compuerta XOR**: ¿el sistema valida el pago? Destino: **Actividad de servicio**, el sistema registra la compra. **Flujo de Secuencia o de Mensaje.** Origen: **Compuerta XOR**: ¿el sistema valida el pago? Destino: **Actividad de usuario**, el cliente escribe el número del medio de pago.

Tabla 26 – Tabla de Extracción de Elementos BPMN

El resultado del "*Proceso de Extracción de Elementos BPMN*" son los elementos BPMN identificados. A continuación, mostramos los mismos sin repeticiones:

SwimLanes:
- cliente
- medio de pago
- sistema
- sistema del medio de pago

Evento Inicial:
- el cliente presionó el botón con la imagen de compra

Sub-Procesos:
- el cliente trabaja con el pago de la compra
- el cliente se identifica con el sistema

Actividades de usuario:
- el cliente escribe el número del medio de pago
- el cliente escribe el mes de caducidad del medio de pago
- el cliente escribe el año de caducidad del medio de pago
- el cliente escribe el código de verificación del medio de pago
- el cliente presiona el botón con la imagen de pago

Actividad de servicio:

el sistema del medio de pago procesa los datos enviados
el sistema valida la compra
el sistema registra la compra

Compuerta XOR:

¿el sistema valida la compra?

Flujos de Secuencia o de Mensaje:

Para exponer los mismos se utilizó una tabla (ver tabla 27) que ayuda a organizar y visualizar de una manera amigable el origen y el destino cada "Flujo de Secuencia" identificado.

Flujos de Secuencias o de Mensajes	
Origen	**Destino**
el cliente trabaja con el pago de la compra	el cliente se identifica con el sistema
el cliente presionó el botón con la imagen de compra	el cliente escribe el número del medio de pago
el cliente escribe el número del medio de pago	el cliente escribe el mes de caducidad del medio de pago
el cliente escribe el mes de caducidad del medio de pago	el cliente escribe el año de caducidad del medio de pago
el cliente escribe el año de caducidad del medio de pago	el cliente escribe el código de verificación del medio de pago
el cliente escribe el código de verificación del medio de pago	el cliente presiona el botón con la imagen de pago
el cliente presiona el botón con la imagen de pago	el sistema envió datos hacia el medio de pago
el sistema envió datos hacia el medio de pago	el sistema del medio de pago procesa los datos enviados
el sistema del medio de pago procesa los datos enviados	el sistema valida la compra
el sistema valida la compra	¿el sistema valida la compra?
¿el sistema valida la compra?	el sistema registra la compra
¿el sistema valida la compra?	el cliente escribe su número de medio de pago

Tabla 27 – Flujos de Secuencia y de Mensaje identificados en el Mockup

Luego de haber realizado éste paso, es posible inferir el o los procesos de negocios involucrados utilizando la notación BPMN. La derivación de modelo BPMN consiste principalmente en descubrir el "Flujo Normal" (concepto mencionado en la sección "2.2.3.6 – Conectores") del proceso de negocio.

3.6.1.4 Proceso de Identificación del Flujo Normal

Después de haber trabajado en el "*Proceso de Extracción de Elementos BPMN*" el objetivo es derivar el diagrama BPMN implícito contenido dentro del Mockup.

El "*Proceso de Identificación del Flujo Normal*" consiste en analizar cada uno de los elementos BPMN identificados en el paso anterior. Desde el análisis de las relaciones que existen entre los elementos BPMN identificados se infiere el "Flujo Normal" del BP.

Para trabajar en el "*Proceso de Identificación del Flujo Normal*" se deben seguir una serie de seis pasos que se describen a continuación:

Primero
En primer lugar, el equipo filtra el Evento Inicial contenido en el resultado del "Proceso de División de Anotaciones EUGEBP y Extracción de Elementos BPMN": *el cliente presionó el botón con la imagen de compra*.

Segundo
En segundo lugar, el equipo filtra del conjunto de oraciones aquellas que contienen adjetivos ordinales:

- Primero, el cliente escribe el número del medio de pago.
- Segundo, el cliente escribe el mes de caducidad del medio de pago.
- Tercero, el cliente escribe el año de caducidad del medio de pago.
- Cuarto, el cliente escribe el código de verificación del medio de pago.
- Quinto, el cliente presiona el botón con la imagen de pago.
- Sexto, Trigger: el sistema envió datos hacia el medio de pago.
- Séptimo, navegar hacia "el sistema del medio de pago procesa los datos enviados".
- Octavo, el sistema valida el pago.
- Noveno, si, "el sistema valida el pago", entonces navegar hacia "el sistema registra la compra". Sino, navegar hacia "el cliente escribe su número de medio de pago".

Los adjetivos ordinales nos proporcionan el orden en el que se debe ejecutar el BP.

Tercero
En tercer lugar, el equipo extrae todos los "Objetos de Flujo (Eventos, Actividades, Compuertas) BPMN" contenidos en las oraciones filtradas del paso dos:

- **Actividades de usuario**:
  el cliente escribe el número del medio de pago
  el cliente escribe el mes de caducidad del medio de pago
  el cliente escribe el año de caducidad del medio de pago
  el cliente escribe el código de verificación del medio de pago
  el cliente presiona el botón con la imagen de pago
- **Actividad de servicio**:
  el sistema del medio de pago procesa los datos enviados
  el sistema valida el pago
  el sistema registra la compra
- **Compuerta XOR:**
  ¿el sistema valida el pago?

Cuarto
En cuarto lugar, el equipo identifica los "SwimLanes BPMN" contenidos en las oraciones filtradas del paso dos:

- **SwimLanes**:
  cliente
  sistema
  sistema del medio de pago

Quinto
En quinto lugar, el equipo filtra desde "**Flujos de Secuencias y de Mensajes**" detectados en el paso "**Proceso de Extracción de Elementos BPMN**" solo aquellos flujos que contengan (en el origen o en el destino) aquellos "Objetos de Flujo" filtrados en el paso tres (ver tabla 28):

Flujos de Secuencias o de Mensajes	
Origen	**Destino**
el cliente presionó el botón con la imagen de compra	*el cliente escribe el número del medio de pago*
el cliente escribe el número del medio de pago	*el cliente escribe el mes de caducidad del medio de pago*
el cliente escribe el mes de caducidad del medio de pago	*el cliente escribe el año de caducidad del medio de pago*
el cliente escribe el año de caducidad del medio de pago	*el cliente escribe el código de verificación del medio de pago*
el cliente escribe el código de verificación del medio de pago	*el cliente presiona el botón con la imagen de pago*
el cliente presiona el botón con la imagen de pago	*el sistema envía datos hacia el medio de pago*
el sistema envía datos hacia el medio de pago	*el sistema del medio de pago procesa los datos enviados*
el sistema del medio de pago procesa los datos enviados	*el sistema valida el pago*
el sistema valida el pago	*¿el sistema valida el pago?*
¿el sistema valida el pago?	*el sistema registra la compra*
¿el sistema valida el pago?	*el cliente escribe su número de medio de pago*

Tabla 28 – Flujos Normales de Secuencias o de Mensajes

Sexto
El equipo de analistas, de acuerdo a todos los elementos BPMN identificados en los pasos anteriores puede inferir un diagrama de procesos de negocios BPMN. El diagrama BPMN inferido desde las anotaciones EUGEBP se muestra a continuación (ver figura 39):

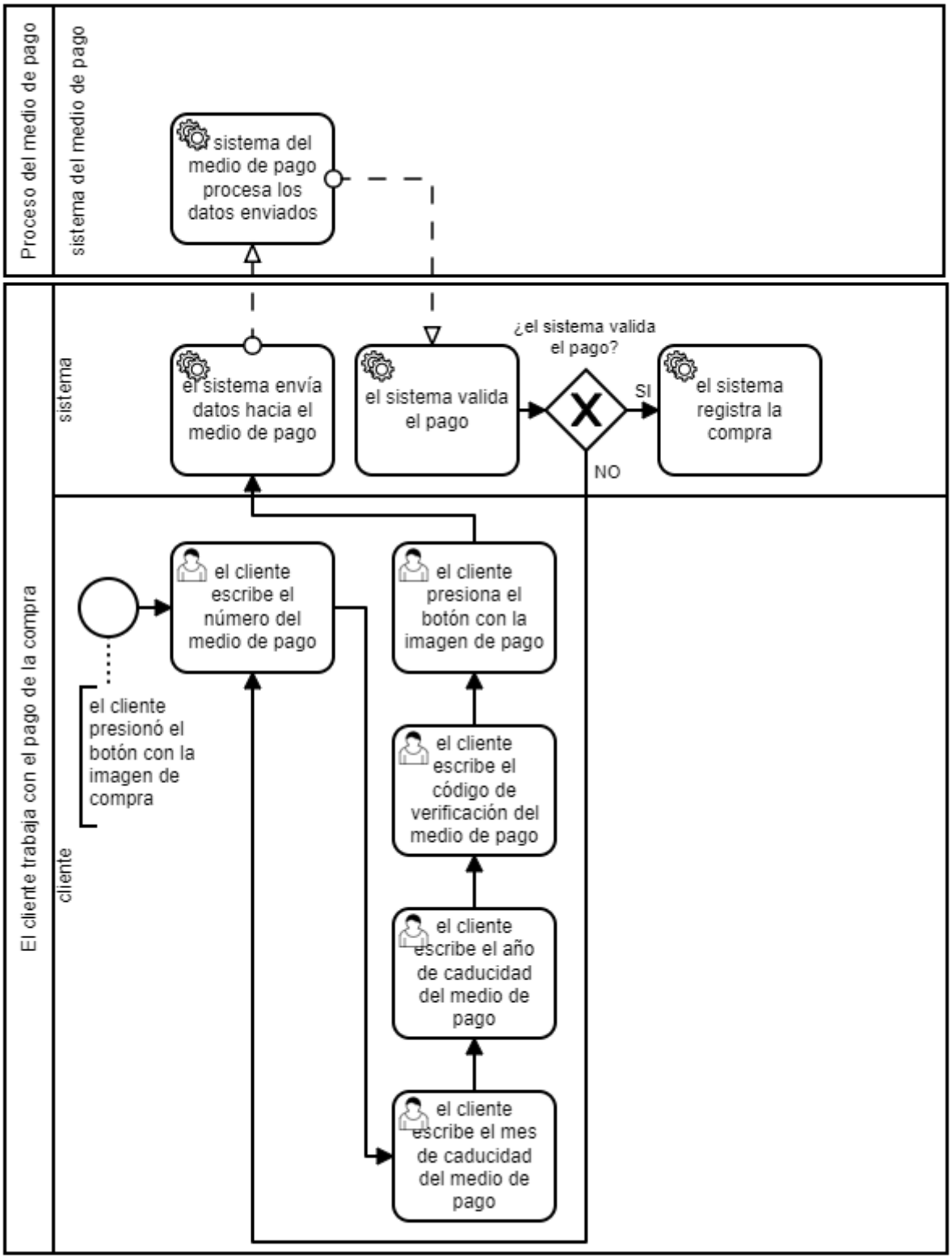


Figura 39 – Diagrama BPMN inferido desde las anotaciones EUGEBP

Notar que todos los elementos BPMN que componen el diagrama BPMN precedente fueron extraídos desde las anotaciones EUGEBP correspondientes.

Para determinar el "Evento final" aplicamos la Regla 9. Si aplicamos lo mencionado, el diagrama BPMN resultante quedaría de la siguiente manera (ver figura 40):

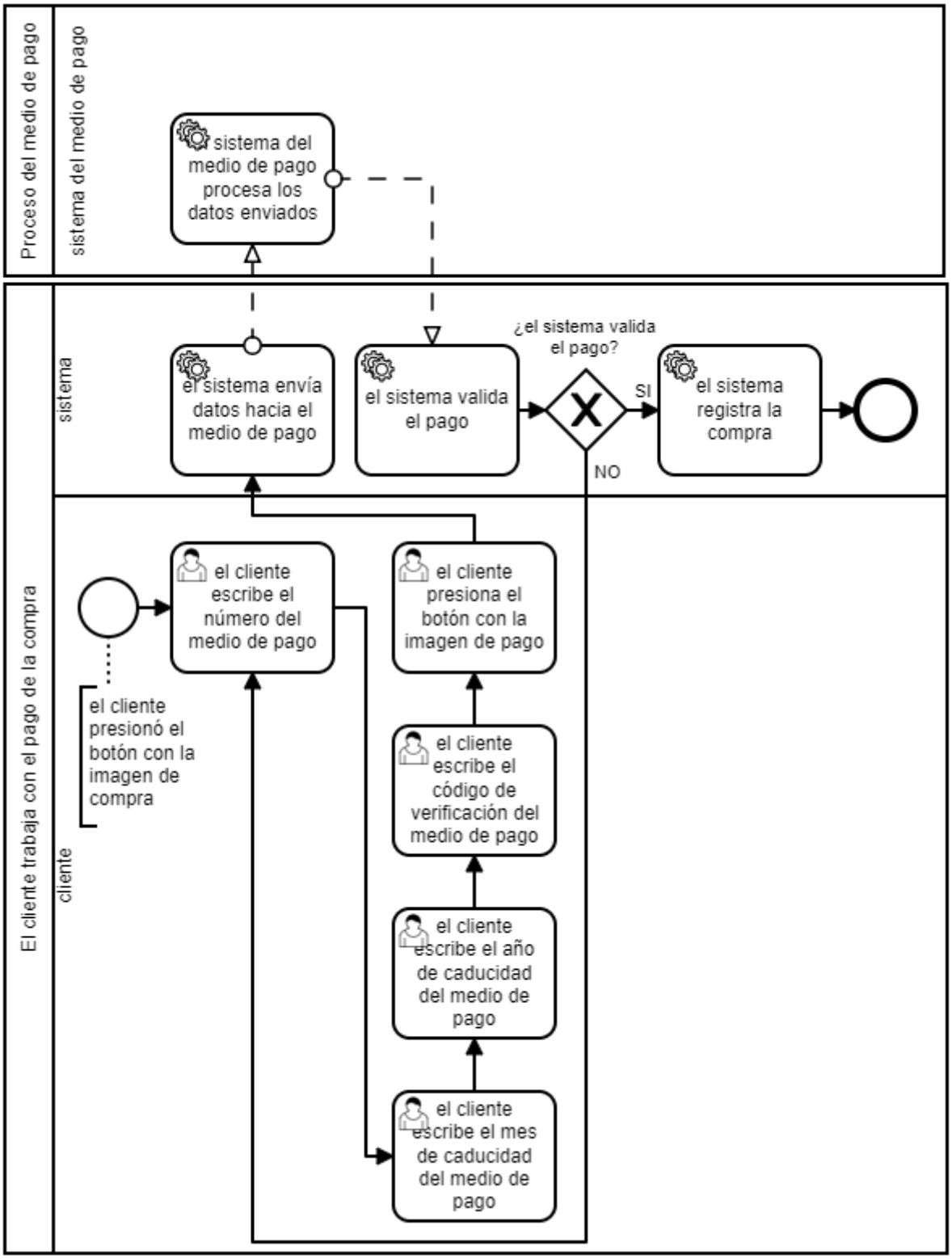


Figura 40 – Diagrama BPMN inferido desde las anotaciones EUGEBP final

El resultado del "*Proceso de Identificación del Flujo Normal*" es el diagrama BPMN derivado desde las anotaciones EUGEBP que se corresponden con el "Flujo Normal".

3.7 Identificación del Proceso de Negocio Superior

En la sección 2.2.3.1.2 habíamos notado la figura de "Proceso Superior" y "Sub-proceso". Dichos conceptos van a ser referenciados en la presente sección.

Del "Proceso de Identificación del Flujo Principal del BP" anterior, podemos notar que, la derivación del diagrama BPMN vinculado al "*Mockup, el cliente trabaja con el pago de la compra, vista (6)*" se conecta al Sub-Proceso denominado "*el cliente se identifica con el sistema*". Además, podemos observar que existe una anotación que quedó fuera del "Flujo Principal" derivado. Dicha anotación es la siguiente (ver tabla 29):

Oración EUGEBP	**Nº Regla EUGEBP**	**Elemento BPMN**
Navegar hacia "Mockup, el cliente se identifica con el sistema, vista (1)".	12	**Flujo de Secuencia.** Origen: **Sub-Proceso** "el cliente trabaja con el pago de la compra". Destino: **Sub-Proceso** "el cliente se identifica con el sistema".

Tabla 29 – Fragmento del "Proceso de Extracción de Elementos BPMN"

Sin embargo, sabemos que dicha anotación EUGEBP permite inferir un sub-proceso denominado el "*el cliente se identifica con el sistema*".

Si se trabajara en la derivación del diagrama BPMN vinculado al "*Mockup, el cliente se identifica con el sistema, vista (1)*" (ver Anexo 1) llegaríamos a la conclusión de que el mismo se conecta a dos sub-procesos denominados: "*el cliente trabaja con el carro de compra*" y "*el cliente registra sus datos*".

Luego, si se trabajara en la derivación del diagrama BPMN vinculado a la anotación "*Mockup, el cliente trabaja con el carro de compra, vista (2)*" llegaríamos a la conclusión de que el mismo ésta conectado al sub-proceso denominado "*el cliente trabaja con el pago de la compra*".

Si realizamos una representación BPMN de todos los sub-procesos mencionados hasta el momento podemos observar lo siguiente (ver figura 41):

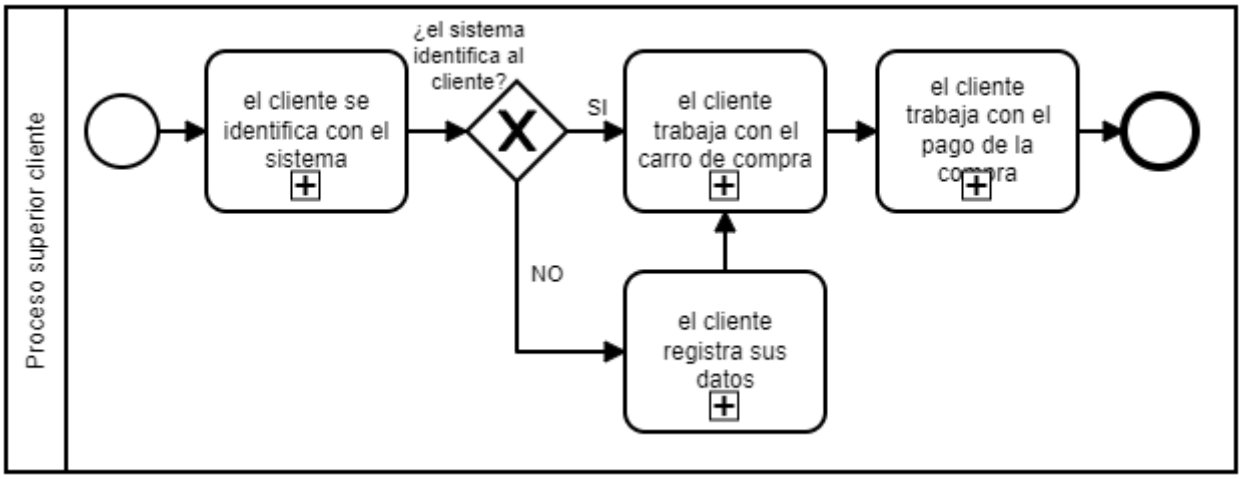


Figura 41 – Diagrama BPMN Superior inferido desde las anotaciones EUGEBP

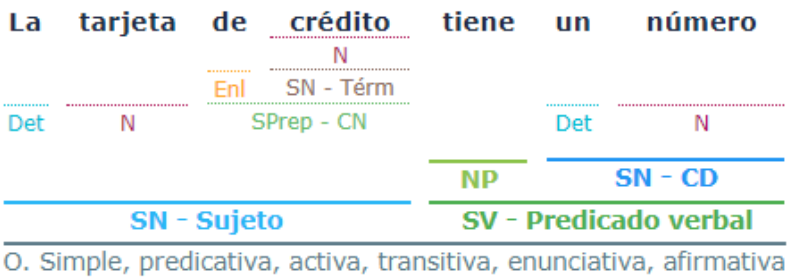
Es decir que si derivamos todos los diagramas BPMN correspondientes desde los Mockups anotados con EUGEBP y filtramos los Sub-Procesos junto con las conexiones de los mismos es posible producir una visión macro del "Proceso de Negocio Superior" que los contiene. Es decir que es posible analizar en mayor profundidad esté aspecto. Solo se hace mención al "Proceso de Negocio Superior", pero el mismo no va a ser analizado en la presente tesis.

3.8 Análisis de Oraciones Simples con NLP

Para el contexto de la tesis es importante destacar que todas las anotaciones, aplicadas sobre los Mockups del ejemplo, fueron analizadas con las siguientes herramientas de análisis de lenguaje natural: "Sintaxis.org"[13] y "Stanford CoreNLP"[14]. Dichas herramientas permiten verificar que las anotaciones son oraciones simples gramaticalmente correctas. En éste sentido una herramienta de análisis de oraciones NLP podría trabajar adecuadamente con ellas.

A continuación, mostramos el resultado del procesamiento del lenguaje natural, obtenido desde las herramientas "Sintaxis.org" y "Stanford CoreNLP" (ver figuras 42 y 43), aplicado a una oración cualquiera como por ejemplo "*La tarjeta de crédito tiene un número*":

[13] https://sintaxis.org/analizador/
[14] https://corenlp.run/



Se trata de una oración simple que según el tipo de predicado es predicativa, activa, transitiva y según la actitud del hablante se clasifica en enunciativa, afirmativa

La oración está formada por:

- Sujeto - (Sintagma nominal)
  - determinante: ***La***
  - núcleo: ***tarjeta***
  - complemento del núcleo - (Sintagma preposicional)
    - enlace: ***de***
    - término - (Sintagma nominal)
      - núcleo: ***crédito***
- Predicado verbal - (Sintagma verbal)
  - Núcleo del predicado: ***tiene***
  - Complemento directo - (Sintagma nominal)
    - determinante: ***un***
    - núcleo: ***número***

Figura 42 – Analizador “sintaxis.org” aplicado a una oración EUGEBP

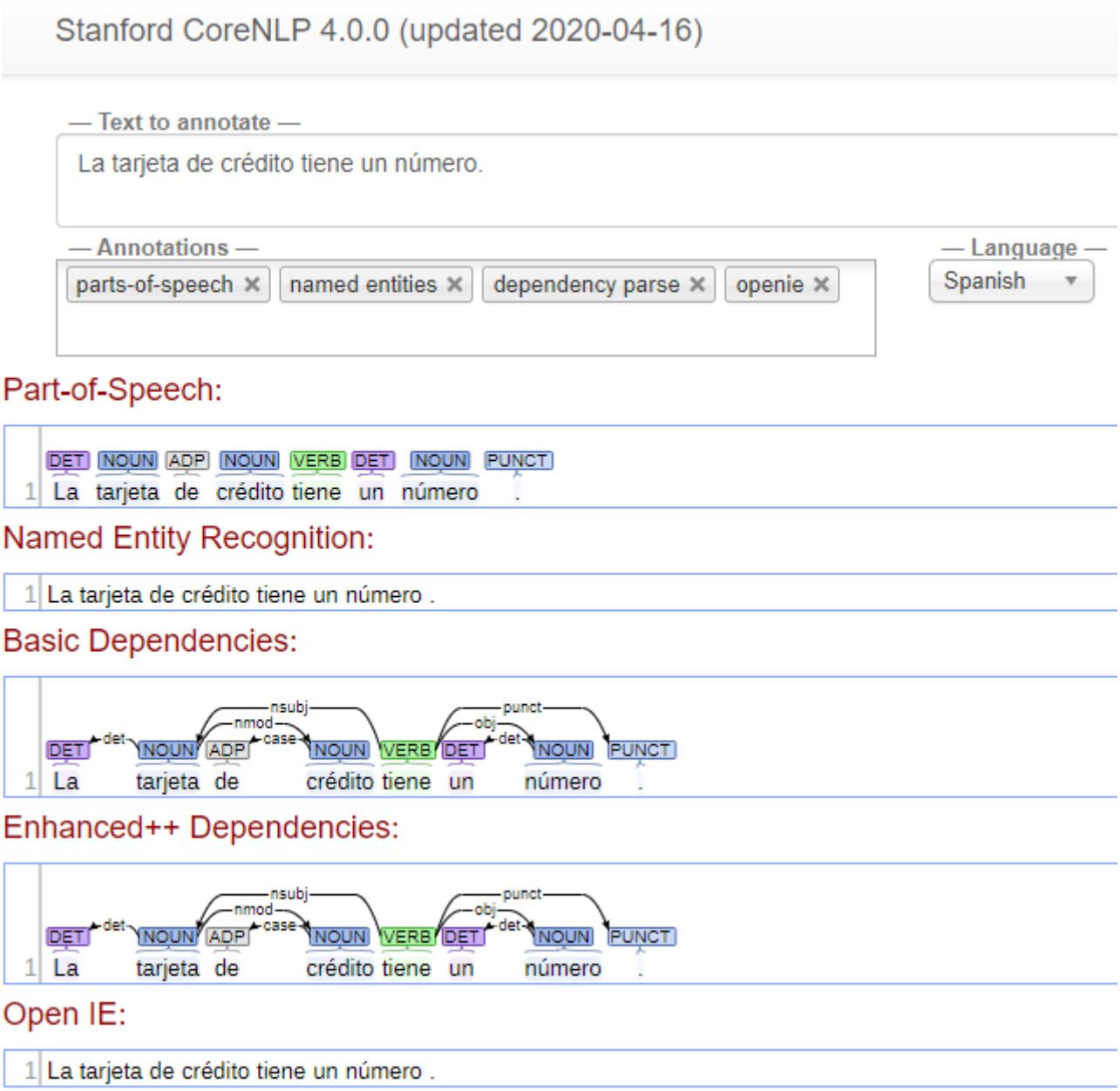


Figura 43 – Analizador “Stanford CoreNLP” aplicado a una oración EUGEBP

En el siguiente capítulo se desarrolla en mayor profundidad la idea de analizar oraciones simples, gramaticalmente correctas, utilizando herramientas de software diseñadas para tal fin.

Capítulo 4 Herramientas para el análisis de oraciones aplicadas a EUGEBP

4.1 Introducción

"PEG.js" es una herramienta que permite realizar un análisis gramatical simple y rápido sobre un texto cualquiera. "Stanford CoreNLP" es una herramienta que permite el procesamiento del lenguaje natural con un mayor nivel de detalle. Este último proporciona una tecnología que permite analizar las palabras y marcar las estructuras de las oraciones en términos de dependencias sintácticas.

Es posible enmarcar la sintaxis asociada a las "Reglas de redacción EUGEBP" a través del uso de "PEG.js". Con dicha herramienta podemos controlar la estructura gramatical de las anotaciones realizadas sobre los Mockups. Si la notación no cumple con las "Reglas de redacción EUGEBP" podemos informar éste hecho automáticamente. Esto sirve como un control superficial de la anotación. Sin embargo, para realizar un análisis profundo de cada oración, necesitamos valernos de una herramienta más robusta. En éste contexto, es posible aplicar "Stanford CoreNLP" para lograr un análisis exhaustivo (sujeto, verbo y objeto) de cada oración.

Las tecnologías mencionadas pueden colaborar con la automatización de algunos de los pasos (*Proceso de Producción de Anotaciones EUGEBP*, *Proceso de División de Anotaciones EUGEBP* y, *Proceso de Extracción de Elementos BPMN)* que componen la "Técnica EUGEBP". Recodemos que EUGEBP se apoya en un conjunto de "Reglas de redacción" que poseen una estructura gramatical predeterminada. Dicha gramática organiza las oraciones de las anotaciones y las palabras de las oraciones de una manera conveniente que luego permite la extracción de elementos BPMN. En éste capítulo investigamos si dicha tarea de análisis y extracción es posible de automatizar.

A continuación, realizamos una breve presentación de las herramientas mencionadas.

4.2 PEG.js

PEG.js es un generador de analizadores de sintaxis simple para JavaScript. Permite producir analizadores rápidos con un excelente informe de errores. Puede usarse, por ejemplo, para procesar datos complejos o lenguajes de computación y para construir intérpretes de sintaxis fácilmente.

Sus principales características son:

- Sintaxis gramatical simple y expresiva.
- Integra análisis léxico y sintáctico.
- Se puede usar desde su navegador, desde la línea de comando o mediante la API de JavaScript.
- Funciona en todos los navegadores actuales (Internet Explorer 8+, Edge, Firefox, Chrome, Safari, Opera).

Para generar un analizador con "PEG.js" se debe escribir una gramática que describa la entrada esperada y luego, se debe especificar lo que el analizador debe devolver (utilizando acciones semánticas en partes coincidentes de la entrada). El analizador generado en sí mismo es un objeto JavaScript con una API simple.

Escribir en "PEG.js" consiste en escribir patrones o reglas a partir de una regla principal. Por ejemplo, nuestra primera regla podría ser la siguiente:

*contadorDePalabras = palabra**

“PEG.js” está muy emparentado con las expresiones regulares. Se parece mucho a ellas, pero aquí "*palabra*" representa otra regla. En expresiones regulares, "*" significa que el token precedente (para nuestro ejemplo “*palabra*” es el token) puede estar presente 0 o muchas veces dentro de la cadena de entrada (0-n). En cambio, aquí “*palabra*” hace referencia a otra regla y “PEG.js” espera que se defina. En este contexto se define la regla “*palabra*” de la siguiente manera:

palabra = letra+

Ahora, como las expresiones regulares, estamos haciendo uso del operador “+”. El operador “+” indica que el token debe estar presente al menos una vez, pero puede repetirse muchas veces (1 - n). Entonces, hemos resuelto el problema de la regla “*palabra*”, pero creamos otra regla que no está definida "*letra*". En consecuencia, se define la regla “*letra*” de la siguiente manera:

letra = [a-zA-Z]

El operador “[]” significa que cualquier carácter dentro de los corchetes puede coincidir. Es como un conjunto. “*a-z*” literalmente significa caracteres desde la “a” a la “z” en minúscula. “*A-Z*” literalmente significa caracteres desde la “A” a la “Z” en mayúscula. En resumen, la regla “*letra*” indica que se puede introducir cualquier carácter en minúscula o mayúscula.

En éste punto “PEG.js” acepta las reglas escritas y la mismas constituyen nuestro “analizador”. Lo único que nos falta definir sería nuestra salida. Para contar las palabras, debemos poder hacer referencia a ellas. Para hacer referencia a algo, utilizamos la siguiente sintaxis en “PEG.js”:

contadorDePalabras = p:palabra { return p.length; }*

La sintaxis "label:" permite acceder al texto coincidente real dentro de su “PEG.js”.

Nuestro primer analizador de prueba entonces es el siguiente:

```
contadorDePalabras = p:palabra* { return p.length; }
palabra = letra+
letra = [a-zA-Z]
```

Hasta aquí realizamos una breve presentación de “PEG.js”.

4.3 Stanford CoreNLP

“Stanford CoreNLP” es un analizador de sintaxis probabilístico desarrollado por la Universidad de Stanford que se basa en seleccionar el mejor análisis según aquel que sea más probable a partir de un conjunto de ejemplos analizados correctamente por lingüistas. Proporciona un sistema de redes neuronales completo que toma un texto como entrada y realiza sobre el mismo un conjunto de tareas de análisis sintáctico, desde la tokenización y la segmentación de oraciones, hasta el etiquetado de POS y el análisis de dependencia [55].

“Stanford CoreNLP” además proporciona un kit de herramientas extensible que posibilita un análisis central del lenguaje natural. Este juego de herramientas es bastante utilizado, tanto en la comunidad de investigación de NLP como también entre usuarios comerciales y gubernamentales de tecnología de código abierto de NLP. El kit posee un diseño simple y accesible, interfaces directas, incluye componentes de análisis robustos y de buena calidad[56]. Algunas de las herramientas del kit son: el etiquetador de parte del discurso (POS - part-of-speech), el reconocedor de entidad con nombre (NER - named entity recognizer), el analizador de dependencias (DependencyParser), entre otras.

El analizador de Stanford es un analizador estadístico. Un analizador de lenguaje natural es un programa que resuelve la estructura gramatical de las oraciones, por ejemplo, qué grupos de palabras van juntas (como "frases") y qué palabras son el "*sujeto*" u "*objeto*" de un "*verbo*". Los analizadores probabilísticos usan el conocimiento del lenguaje obtenido de las oraciones analizadas a mano para tratar de producir el análisis más probable de nuevas oraciones. Estos analizadores estadísticos aún cometen algunos errores, pero generalmente funcionan bastante bien. Su desarrollo fue uno de los mayores avances en el procesamiento del lenguaje natural en la década de 1990.

En el siguiente ejemplo (ver figura 44) podemos ver el kit de herramientas Stanford CoreNLP en funcionamiento:

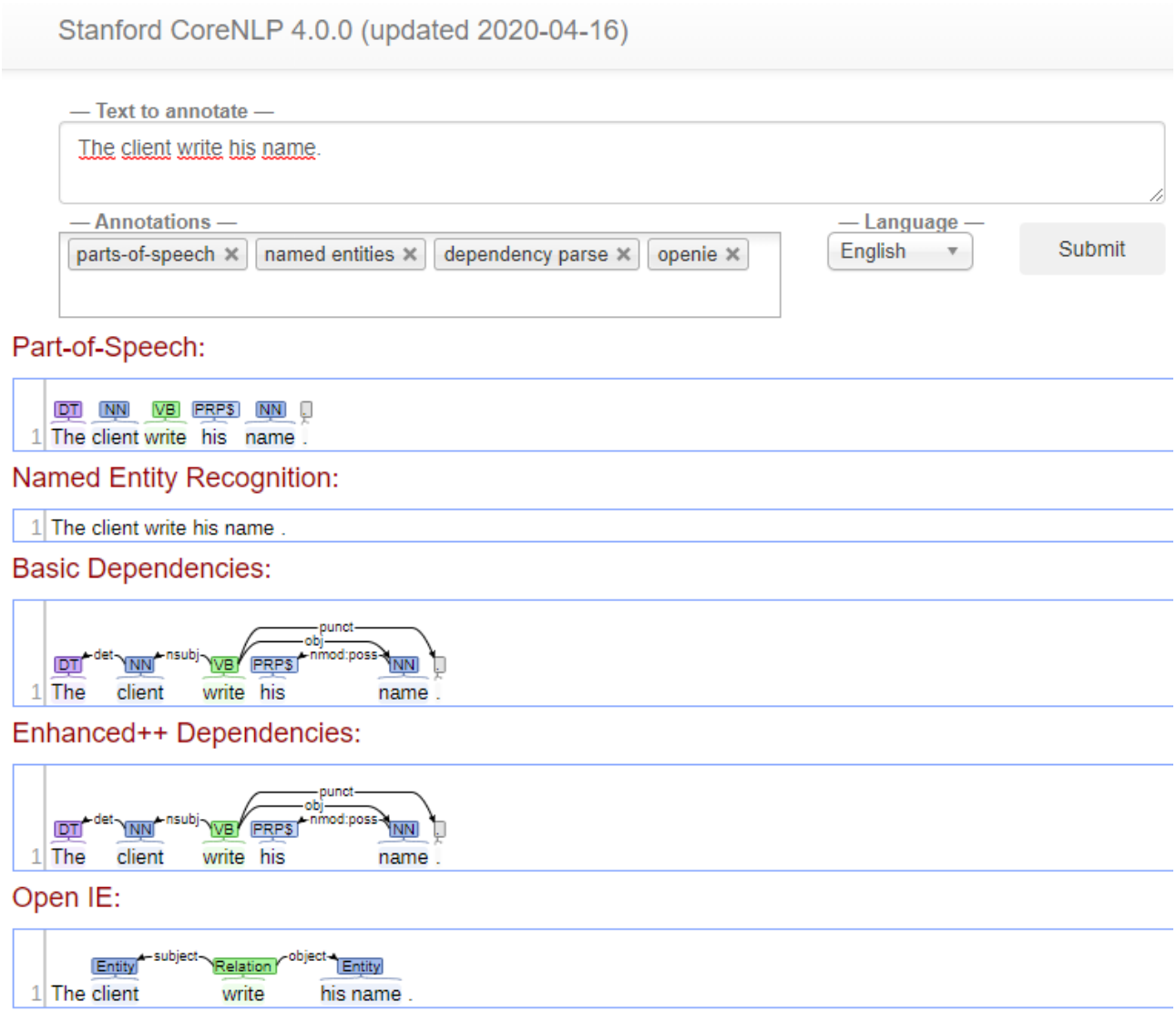


Figura 44 – Stanford CoreNLP aplicado a la oración "The client write his name."

Además de proporcionar un analizador de inglés, el analizador ha sido adaptado para trabajar con otros idiomas. También se incluye un analizador chino basado en el "Treebank chino", un analizador alemán basado en el "Corpus Negra" y analizadores árabes basados en el "Penn Arabic Treebank". El analizador también se ha utilizado para otros idiomas, como el italiano, el búlgaro y el portugués. Stanford Parser version 3.9.2 incluye modelos para árabe, chino, inglés, francés, alemán y español.

Hasta aquí realizamos una breve presentación de "Stanford CoreNLP".

4.4 Gramática “PEG.js” aplicada a EUGEBP

En la presente sección mostramos de qué manera “PEG.js” puede ser aplicado a EUGEBP como soporte. La idea es analizar las anotaciones, que los analistas junto a los stakeholders producen, a través de “PEG.js” y determinar si dicha anotación cumple con las reglas de producción de anotaciones de EUGEBP.

El siguiente analizador “PEG.js” permite analizar, las anotaciones EUGEBP introducidas por los stakeholders y los analistas. La misma se aplica a la “*Regla 4: [Sujeto] + [Verbo en tiempo presente] + [Objeto]*”.

Ejemplo: El cliente *[Sujeto]* escribe *[Verbo en tiempo presente]* su nombre *[Objeto]*.

Analizador “PEG.js” correspondiente a la “Regla 4”:

```
anotacion = oracionSimple
oracionSimple = strOS: oracionSimpleEnTiempoPresente {
        return{
        oracion:{
        customData: strOS.toString().replace(/['"]+/g, '').split(',').join('')
      },
      elementoBPMN:"actividadTiempoPresente"
   }
}
oracionSimpleEnTiempoPresente = sujetoOracion unEspacio verboEnTiempoPresente unEspacio
objetoOracion posfijoOracion
sujetoOracion = articulo unEspacio sustantivo
articulo = ("El" / "el" / "La" / "la" / "Los" / "los" / "Las" / "las" / "Un" / "un" / "Una" / "una" /
"Unos" / "unos" / "Unas" / "unas")
sustantivo = palabra
verboEnTiempoPresente = palabra
objetoOracion = oracion
posfijoOracion = "."
oracion = p:(palabra espacio?)*
palabra = letra+
letra = [a-zA-ZÀ-ÿ\u00f1\u00d1]
numero = [0-9]
espacio = [ \t\n\r]*
unEspacio = " "
```

El analizador “PEG.js” precedente puede verificarse en línea[15]. A continuación (ver figura 45), mostramos la correcta verificación del analizador desarrollado:

[15] https://pegjs.org/online

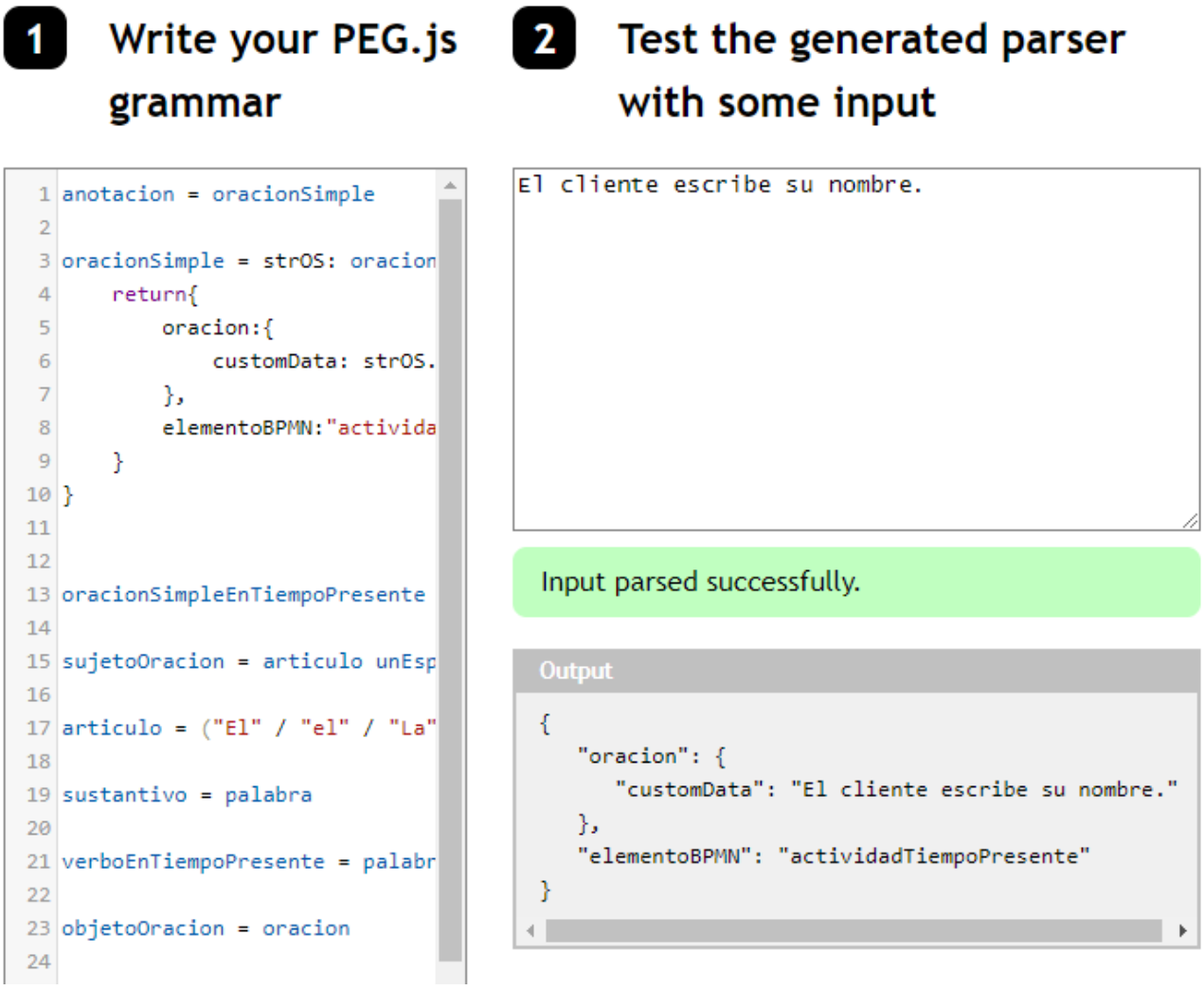


Figura 45 – Verificación del analizador “PEG.js” desarrollado para la “Regla 4 EUGEBP”

En el “Apéndice 2” se exponen un conjunto de analizadores (correspondientes a las reglas de producción de anotaciones 5, 6, 7, 8, 10, y 12) desarrollados para ayudar a la verificación de la “Producción de Anotaciones EUGEBP”.

De acuerdo al ejemplo precedente notamos que es posible desarrollar un analizador para cada una de las reglas que componen EUGEBP. Sin embargo, con los mismos solo es posible analizar y descartar aquellas anotaciones u oraciones que no cumplen con las estructuras básicas de escrituras de las mismas. En éste contexto es necesario utilizar alguna herramienta que nos permita analizar en mayor profundidad cada oración.

4.5 Stanford CoreNLP aplicada al análisis de oraciones

Existe mucha literatura vinculada al análisis de oraciones utilizando herramientas NLP Una de tales herramientas es Standford CoreNLP [56][57] (existen otras como Spacy, Freeling, Gate, NLTK). Por defecto, Stanford CoreNLP espera y procesa textos en inglés. Pero, Stanford CoreNLP fue diseñado desde el principio para trabajar con múltiples lenguajes humanos y tiene cuidado con cosas como diferentes codificaciones de caracteres. Se han desarrollado componentes para varios idiomas principales y los mismos están a disposición en forma de paquetes de idiomas (archivos jar). La siguiente figura (ver figura 46) resume el soporte actual de idiomas extranjeros Stanford CoreNLP.

ANNOTATOR	AR	ZH	EN	FR	DE	ES
Tokenize / Segment	✓	✓	✓	✓		✓
Sentence Split	✓	✓	✓	✓	✓	✓
Part of Speech	✓	✓	✓	✓	✓	✓
Lemma			✓			
Named Entities		✓	✓	✓	✓	✓
Constituency Parsing	✓	✓	✓	✓	✓	✓
Dependency Parsing		✓	✓	✓	✓	
Sentiment Analysis			✓			
Mention Detection		✓	✓			
Coreference		✓	✓			
Open IE			✓			

Figura 46 - Soporte de idiomas Stanford CoreNLP

Para ejecutar Stanford CoreNLP en un idioma compatible, es necesario incluir el ".jar" de modelos para ese idioma en su CLASSPATH. Los ".jars" para cada idioma se pueden encontrar en la página oficial [16]. La siguiente figura (ver figura 47) muestra los diferentes ".jars" existentes:

LANGUAGE	MODEL JAR	VERSION
Arabic	download	4.0.0
Chinese	download	4.0.0
French	download	4.0.0
German	download	4.0.0
Spanish	download	4.0.0

Figura 47 - ".jars" de modelos de idiomas Stanford CoreNLP

Se puede construir y correr un "pipeline" en Java de la siguiente manera (ver figura 48):

[16] https://stanfordnlp.github.io/CoreNLP/human-languages.html

```
String text = "La Universidad de Stanford se encuentra en Palo Alto.";
StanfordCoreNLP pipeline = new StanfordCoreNLP("spanish");
CoreDocument doc = pipeline.processToCoreDocument(text);
```

Figura 48 – Declaración de un “pipeline” en Java con Stanford CoreNLP

Las dependencias de Stanford proporcionan una representación de las relaciones gramaticales entre palabras en una oración. Han sido diseñadas para ser fácilmente entendidas y efectivamente utilizadas por personas que desean extraer relaciones textuales. Las dependencias de Stanford (SD – Stanford Dependencies) son trillizas: nombre de la relación, gobernador y dependiente. Las dependencias estándares para la oración “*Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas*”, se muestran a continuación, así como también dos representaciones gráficas: las dependencias estándares (colapsadas y propagadas) y la representación de dependencias básica en la que cada palabra en la oración (excepto el encabezado de la oración) depende de otra palabra (sin colapsar, sin propagación)[56] (ver figura 49).

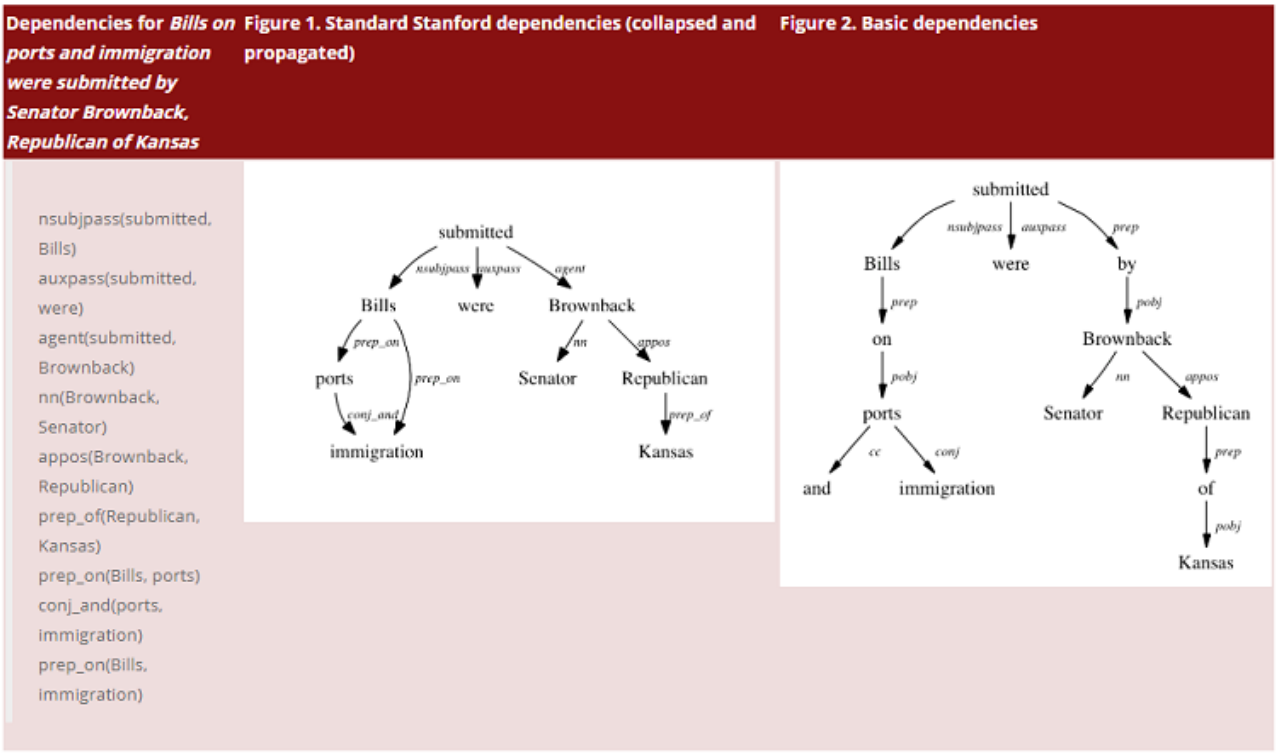


Figura 49- Dependencias Estándares y Representación de Dependencias Básicas

La herramienta “Interface Annotator” es una interfaz para agregar anotaciones a un anotador parcialmente anotado. Opera explícitamente sobre los objetos de anotación. Los Annotators deben ser entregados a un AnnotatorPipeline para hacer canalizaciones de anotaciones, y, por lo tanto, los implementadores de esta interfaz deben estar diseñados para jugar bien con otros Anotadores. Cuando se utiliza ésta herramienta se debe indicar explícitamente qué anotaciones se asume que existen en la anotación (como parse, POS tag, etc.), qué claves se espera (ver, por ejemplo, las de CoreAnnotations), y qué anotaciones agregarán (o modificarán).

La herramienta “DependencyParser” es parte de “StanfordParser”. Un analizador de dependencia (dependency parser) analiza la estructura gramatical de una oración, estableciendo relaciones entre las palabras del tipo "cabeza" y las palabras que modifican esas cabezas. La figura (ver figura 50) a

continuación muestra un análisis de dependencia de una oración corta. La flecha de la palabra "*moving*" hacia a la palabra "*faster*" indica que "*faster*" modifica a "*moving*", y la etiqueta "*advmod*" asignada a la flecha describe la naturaleza exacta de la dependencia.

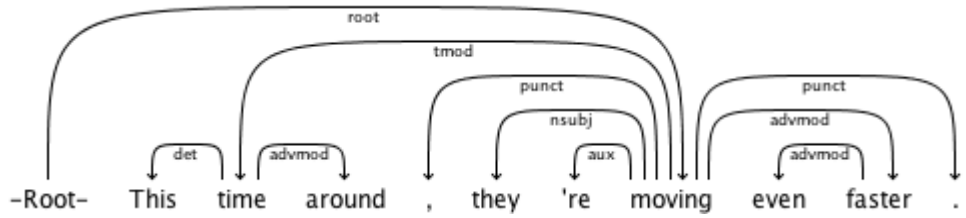


Figura 50 - Analizador de dependencias "DependencyParser" aplicado al análisis de una oración

La herramienta "DependencyParser" proporciona un analizador de dependencia sintáctico rápido de oraciones en lenguaje natural. Genera salidas basadas en dependencias, de la siguiente manera: Dependencias básicas sin colapsar, guardadas en BasicDependenciesAnnotation
Dependencias mejoradas guardadas en EnhancedDependenciesAnnotation

"Dependency parsing" es la tarea de extraer un análisis de dependencias de una oración. El mismo representa su estructura gramatical y define las relaciones entre las palabras y palabras "principales", que modifican a las primeras. "DependencyParser" define un analizador de dependencia basado en un clasificador alimentado por una red neuronal. La red neuronal acepta entradas de representación distribuidas: representaciones densas y continuas de palabras, su parte de etiquetas de voz y las etiquetas que conectan palabras en un análisis de dependencia parcial. Este analizador también se puede usar mediante programación. La forma más fácil de preparar el analizador sintáctico con un modelo previamente entrenado es llamar a loadFromModelFile (String). Luego de debe llamar a predict (edu.stanford.nlp.util.CoreMap) en la instancia del analizador devuelto para obtener nuevos análisis.

Hasta aquí se presentó en un mayor detalle la herramienta "Stanford CoreNLP" mostrando que es posible analizar oraciones en mayor profundidad oraciones en lenguaje natural a través del uso de la misma.

4.5.1 Stanford CoreNLP DependencyParser aplicado a EUGEBP

En la presente sección proponemos utilizar la herramienta "Stanford CoreNLP DependencyParser" para dar soporte al "*Proceso de División de Anotaciones EUGEBP*" y al "*Proceso de Extracción de Elementos BPMN*". A través del uso de la herramienta mencionada es posible automatizar el análisis de las anotaciones EUGEBP realizadas sobre los Mockups.

Las anotaciones EUGEBP, en esencia, son oraciones simples en lenguaje natural y sobre las mismas podemos aprovechar y aplicar todo el potencial de "Stanford CoreNLP DependencyParser". En la sección "3.4" habíamos mencionado una serie de buenas prácticas para producir anotaciones EUGEBP. Además, mencionamos que el uso de las mismas colabora con la identificación posterior de elementos BPMN. Una vez que los Mockups fueron anotados con EUGEBP lo procedente es analizar cada una de las anotaciones de acuerdo a las "Reglas de redacción EUGEBP". En las actividades "*Proceso de División de Anotaciones EUGEBP*" y "*Proceso de Extracción de Elementos BPMN*", se separan las anotaciones en oraciones y se analiza cada una de ellas para luego extraer, manualmente, el o los elementos BPMN que contienen las oraciones. Para colaborar con dichas actividades se investigó si era posible responder a las siguientes preguntas:

- ¿Es posible determinar si una oración contiene sujeto, verbo y objeto?
- ¿Es posible determinar el tiempo verbal de una oración?
- ¿Es posible determinar sin una oración es gramaticalmente correcta para el contexto EUGEBP?

Se menciona que lo investigado se acotó al lenguaje español. Sin embargo, es posible extender la misma hacia otros lenguajes (ingles, chino, u otros soportados por Standford CoreNLP) pero no es el objetivo de la presente tesis.

A continuación, se presenta el producto de la investigación mencionada.

4.5.2 Extracción del sujeto, verbo y objeto de una oración

Sabemos que uno de los componentes de StandFord CoreNLP es el analizador de dependencias "DependencyParser". El mismo proporciona un análisis sintáctico completo de la oración, incluida la representación de los componentes y de las dependencias que existe entre ellos, basada en un analizador probabilístico [56]. "DependencyParser" es un analizador de dependencias que analiza la estructura gramatical de una oración, estableciendo relaciones entre las palabras del tipo "cabeza" y las palabras que modifican esas cabezas. La figura (ver figura 51) a continuación muestra un análisis de dependencia de una oración simple en español (*La tarjeta de crédito tiene un número*). La flecha de la palabra "*tiene*" hacia a la palabra "*tarjeta*" indica que "*tiene*" modifica a "*tarjeta*", y la etiqueta "*nsubj*" asignada a la flecha describe la naturaleza exacta de la dependencia.

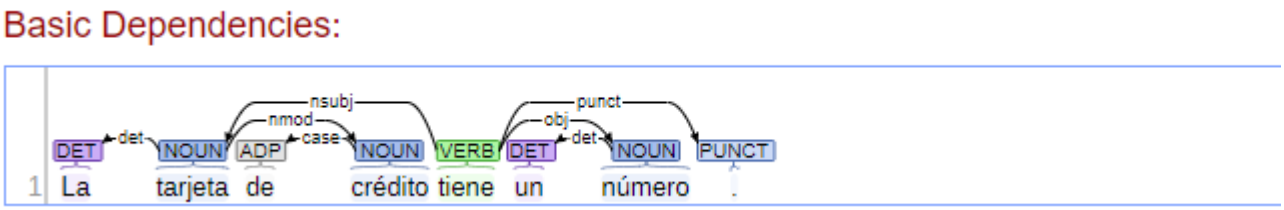


Figura 51 – Análisis de una oración con DependencyParser

A través del uso "DependencyParser" es posible extraer de una oración en lenguaje natural [58] tres elementos relevantes para EUGEBP:

- El verbo de la oración. (el elemento "root" o el elemento principal del árbol de dependencias)
- El sujeto de la oración. (el elemento "nsubj" o el elemento sujeto del árbol de dependencias)
- El objeto de la oración. (el elemento "obj" o el elemento objeto del árbol de dependencias)

Entonces, es posible extraer el sujeto, el verbo y el sujeto de una oración a través de "DependencyParser". Sin embargo, nos está faltando determinar en qué tiempo verbal (pasado o presente según el contexto EUGEBP) se encuentra el verbo.

A continuación, se presenta cómo se puede determinar el tiempo de un verbo con Stanford CoreNLP.

4.5.3 Determinación del tiempo de un verbo

Existe literatura vinculada a la determinación del tiempo verbal de un verbo [57]. Stanford CoreNLP, a través del análisis de las partes del discurso (POS) permite obtener el código "Penn TreeBank" [59] que permite determinar el tiempo del verbo [60]. Las etiquetas "Penn TreeBank" VB, VBD, VBG, VBN permiten identificar los tiempos presente y pasado de un verbo en lenguaje inglés y en general las mismas suelen ser suficientes para dicho propósito. Para el lenguaje español, dentro de Stanford CoreNLP existe un conjunto de etiquetas similar a "Penn TreeBank" denominado "DEFT Spanish TreeBank". El mismo es una versión simplificada del conjunto de etiquetas usado en el "Corpus

AnCora 3.0[17]". El conjunto de etiquetas predeterminado de AnCora tiene cientos de diferentes etiquetas extremadamente precisas. Stanford CoreNLP redujo el conjunto de etiquetas a 85 etiquetas, un tamaño más manejable que aún permite una cantidad útil de precisión. Las etiquetas están diseñadas para seguir siendo compatibles con el estándar EAGLES[18]. El conjunto de etiquetas que hace referencia a los verbos se muestra a continuación (ver tabla 30).

Verbos		
va00000	Verb (unknown)	*should*
vag0000	Verb (auxiliary, gerund)	*habiendo*
vaic000	Verb (auxiliary, indicative, conditional)	*habría*, *habríamos*
vaif000	Verb (auxiliary, indicative, future)	*habrá*, *habremos*
vaii000	Verb (auxiliary, indicative, imperfect)	*había*, *habíamos*
vaip000	Verb (auxiliary, indicative, present)	*ha*, *hemos*
vais000	Verb (auxiliary, indicative, preterite)	*hubo*, *hubimos*
vam0000	Verb (auxiliary, imperative)	*haya*
van0000	Verb (auxiliary, infinitive)	*haber*
vap0000	Verb (auxiliary, participle)	*habido*
vasi000	Verb (auxiliary, subjunctive, imperfect)	*hubiera*, *hubiéramos*, *hubiese*
vasp000	Verb (auxiliary, subjunctive, present)	*haya*, *hayamos*
vmg0000	Verb (main, gerund)	*dando*, *trabajando*
vmic000	Verb (main, indicative, conditional)	*daría*, *trabajaríamos*
vmif000	Verb (main, indicative, future)	*dará*, *trabajaremos*
vmii000	Verb (main, indicative, imperfect)	*daba*, *trabajábamos*
vmip000	Verb (main, indicative, present)	*da*, *trabajamos*
vmis000	Verb (main, indicative, preterite)	*dio*, *trabajamos*
vmm0000	Verb (main, imperative)	*da*, *dé*, *trabaja*, *trabajes*, *trabajemos*
vmn0000	Verb (main, infinitive)	*dar*, *trabjar*
vmp0000	Verb (main, participle)	*dado*, *trabajado*
vmsi000	Verb (main, subjunctive, imperfect)	*diera*, *diese*, *trabajáramos*, *trabajésemos*
vmsp000	Verb (main, subjunctive, present)	*dé*, *trabajemos*
vsg0000	Verb (semiauxiliary, gerund)	*siendo*

[17] http://clic.ub.edu/corpus/en
[18] https://web.archive.org/web/20160325024315/http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html

vsic000	Verb (semiauxiliary, indicative, conditional)	*sería*, *serían*
vsif000	Verb (semiauxiliary, indicative, future)	*será*, *seremos*
vsii000	Verb (semiauxiliary, indicative, imperfect)	*era*, *éramos*
vsip000	Verb (semiauxiliary, indicative, present)	*es*, *son*
vsis000	Verb (semiauxiliary, indicative, preterite)	*fue*, *fuiste*
vsm0000	Verb (semiauxiliary, imperative)	*sea*, *sé*
vsn0000	Verb (semiauxiliary, infinitive)	*ser*
vsp0000	Verb (semiauxiliary, participle)	*sido*
vssf000	Verb (semiauxiliary, subjunctive, future)	*Fuere*
vssi000	Verb (semiauxiliary, subjunctive, imperfect)	*fuera*, *fuese*, *fuéramos*
vssp000	Verb (semiauxiliary, subjunctive, present)	*sea*, *seamos*

Tabla 30 – Fragmento del DEFT Spanish TreeBank vinculado a los verbos

De acuerdo a la tabla precedente, para el lenguaje español es posible determinar el tiempo de un verbo (presente o pasado) utilizando las etiquetas adecuadas del "DEFT Spanish TreeBank" proporcionado por Stanford CoreNLP.

Entonces, a través del análisis de las partes del discurso (POS), Stanford CoreNLP permite obtener el código "Penn TreeBank" necesario para determinar el tiempo de un verbo. Para el ejemplo que se muestra a continuación (ver figura 52), la etiqueta "VERB" puede ser analizada en un detalle más minucioso y de esa manera es posible determinar en qué tiempo verbal se encuentra la misma.

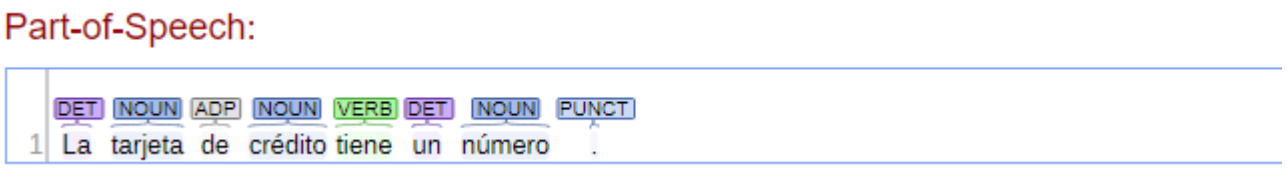


Figura 52 – Obtención de las "Partes del Discurso (POS)" a través de Stanford CoreNLP

Hasta aquí podemos afirmar que es posible determinar el tiempo de un verbo con Stanford CoreNLP

4.5.4 Determinación de una oración gramaticalmente correcta para EUGEBP

Para que una oración en lenguaje español sea gramaticalmente correcta, dentro del contexto de las "Reglas de redacción EUGEBP", la misma debe cumplir con las siguientes características:

- La anotación debe ser una "oración simple", gramaticalmente correcta para el idioma en cuestión (en nuestro caso el español) y debe estar compuesta por las categorías más generales de la "oración simple": el "Sujeto", el "Verbo" y el "Objeto". Dichas categorías existen como unidades sintácticas comprensibles para cualquier ser humano[54].

- Estilísticamente debe comenzar en mayúscula y terminar con un punto.
- El tiempo verbal del verbo debe estar en presente o pretérito prefecto simple.

Entonces, de acuerdo a las dos secciones anteriores, a través del uso de Stanford CoreNLP si es posible automatizar las condiciones que debe cumplir una oración, dentro del contexto EUGEBP, para que la misma sea gramaticalmente correcta.

A continuación, se presenta un Pseudocódigo Stanford CoreNLP diseñado para colaborar con las actividades "Proceso de Producción de Anotaciones EUGEBP", "Proceso de División de Anotaciones EUGEBP" y "Proceso de Extracción de Elementos BPMN".

4.5.5 Pseudocódigo Stanford CoreNLP

A través de Standford CoreNLP, entonces, si es posible automatizar la producción y el análisis de las anotaciones EUGEBP realizadas sobre los Mockups. Para colaborar colaborar con las actividades "*Proceso de Producción de Anotaciones EUGEBP*", "*Proceso de División de Anotaciones EUGEBP*" y "*Proceso de Extracción de Elementos BPMN*" se procedió a investigar si era posible responder a las siguientes preguntas de forma automática:

- ¿Es posible determinar si una oración contiene sujeto, verbo y objeto?
- ¿Es posible determinar el tiempo verbal de una oración?
- ¿Es posible de terminar sin una oración es gramaticalmente correcta para el contexto EUGEBP?

La respuesta a cada una de las preguntas realizadas, luego de haber realizado una breve investigación (4.5.1 - 4.5.4) es afirmativa. Es decir que utilizando Stanford CoreNLP:

- Si es posible determinar si una oración contiene sujeto, verbo y objeto.
- Si es posible determinar el tiempo verbal de una oración.
- Si es posible determinar sin una oración es gramaticalmente correcta para el contexto EUGEBP.

A continuación, presentamos un Pseudocódigo que podría colaborar con las actividades "*Proceso de Producción de Anotaciones EUGEBP*", "*Proceso de División de Anotaciones EUGEBP*" y "*Proceso de Extracción de Elementos BPMN*".

```
Importar Standford CoreNLP DependencyParser
Oraciones = ExtraerTodasLasOracionesDesdeLasAnotacionesEUGEBP //colección de oraciones
Para cada Oracion en Oraciones
  Sujeto = OracionExtraerSujeto
  Verbo = OracionExtraerVerbo
  Objeto = OracionExtraerObjeto
  oraciónCorrecta = False
  Si existe Sujeto, Verbo y Objeto Entonces
    TiempoVerbal = VerboExtraerTiempoVerbal
    Si TiempoVerbal es Presente o Pretérito Perfecto
      oraciónCorrecta = True
    FinSi
  FinSi
  Si (oraciónCorrecta = True) Entonces
    ElementosBPMNParcial = OracionExtraerElementosBPMN//colección de elementos BPMN
    Por cada ElementoBPMN en ElementosBPMNParcial
      ElementosBPMNFinal.Agregar(ElementoBPMN) //colección de elementos BPMN
    FinPor
  FinSi
FinPara
```

En “Anexo 3” se presenta una clase JAVA que implementa un fragmento del pseudocódigo precedente junto con una ejecución del mismo.

Notar que para el pseudocódigo precedente se supone que existe una función denominada “*OracionExtraerElementosBPMN*” que analiza una oración, gramaticalmente correcta para EUGEBP, y devuelve una colección de elementos BPMN contenidos en la oración. No se describe dicha función, pero se menciona que se posee información suficiente como para determinar un tipo de elemento BPMN si se trabaja de acuerdo a lo descripto en la sección 3.6.1.3.

Además, el pseudocódigo precedente solo es un fragmento que se puede complementar con otro que utilice “PEG.js” por ejemplo. Es decir que desde la utilización de varios fragmentos que colaboren con la “*Técnica EUGEBP*”, es posible constituir una base sólida de una aplicación que de soporte a la misma.

Lo que se presentó en ésta sección fue la factibilidad de hacer posible una automatización desde un punto de vista central (análisis automático de oraciones) pero que puede expandirse hacia una aplicación de software real. Sin embargo, no es el propósito de la presente tesis desarrollar una aplicación y por la tanto solo se hace mención al hecho de que es posible hacerla.

Capítulo 5 Evaluación de EUGEBP

En éste capítulo presentamos una evaluación preliminar de "EUGEBP". Para realizar dicha evaluación nos basamos en las directrices propuestas por Wholin et al. [61]. En este contexto, primero, en el ámbito del alcance, se definió el objetivo del experimento. Luego, en el ámbito de la planificación, se definieron las hipótesis y las variables del experimento. A continuación, se procedió a definir las métricas y los materiales para la evaluación. Luego, se abordó el diseño del experimento, la selección de sujetos, la instrumentación y los métodos de recolección utilizados en el experimento. A continuación, se realizó un análisis de los resultados y su interpretación. Finalmente, se consideraron las amenazas a la validez de la evaluación.

5.1 Evaluación Cuantitativa

De acuerdo a lo sugerido por Goal-Question-Metric (GQM) [61] se definió el objetivo del experimento de la siguiente manera:

Evaluar la experiencia de sujetos que estuvieron expuestos a un conjunto de Mockups con anotaciones EUGEBP y un diagrama BPMN derivado de cada uno en el contexto de la Ingeniería de Software.

Después de definir el objetivo, se definió un conjunto de preguntas que nos permitió responder, desde un punto de vista operativo, si el objetivo puede ser alcanzado o no. Aprovechamos los conceptos de precisión y recuperación [62] del campo de investigación de recuperación de información y los adaptamos a nuestro experimento para medir la calidad de las respuestas.

Nuestras principales preguntas de investigación (RQ) fueron:

RQ1: ¿Están de acuerdo los analistas con el diagrama BPMN derivado y sus elementos ?;
RQ2: ¿Los analistas encuentran útil el diagrama BPMN derivado?

En esta evaluación nos enfocamos en reunir las primeras ideas sobre el enfoque para establecer el marco básico de un experimento formal en el futuro.

5.2 Diseño y ejecución de la evaluación

Para responder a las preguntas de investigación, se diseñó un experimento al azar, en el que se solicitó a un conjunto de sujetos que analizaran un grupo de Mockups y, que los mismos, confirmaran la exactitud del diagrama BPMN derivado. En el contexto de este trabajo se elaboraron un grupo de Mockups principales junto a sus correspondientes diagramas BPMN derivados. En base a las preguntas de investigación, se produjeron diferentes objetos experimentales. Luego, se proporcionó a los participantes una serie de formularios de Google[19] (Google Forms) que contenían lo siguiente:

a) Formulario Demográfico: Un cuestionario que contenía preguntas demográficas y preguntas vinculadas a la experiencia profesional de los individuos participantes en diferentes campos de la Ingeniería de Software. Se deja en claro que la información recopilada es confidencial.
b) Formulario de Inicio de Sesión: Se proporcionó un Mockup anotado para el caso de uso de un "Inicio de Sesión del Sistema" junto con un correspondiente diagrama BPMN derivado. En relación a los elementos presentados (Mockup y Diagrama BPMN), también se proporcionó un cuestionario para reunir información asociada a la representatividad del diagrama BPMN

[19] https://forms.gle/vYWGamyKDKoyoNvw6
https://forms.gle/sGzvUHxXEMkJ7Dpj7

derivado. Lo que buscó comprobar fue, si los elementos BPMN, participantes, eventos, actividades, compuertas, flujos BPMN en general eran correctos o no.

c) Formulario de Búsqueda de Productos: Un conjunto de Mockups vinculados a la búsqueda de productos, donde el cliente navega por el sistema para llegar a un producto y un diagrama BPMN correspondiente. El cuestionario asociado a los elementos presentados (Mockups y Diagrama BPMN), es similar al presentado en el punto b).
d) Formulario de Muestra de Productos y Pago: Un conjunto de Mockups vinculados a la muestra de productos, donde el cliente puede ver los productos que va a comprar y el pago de la compra con tarjeta de crédito. Además, se proporcionan los diagramas BPMN correspondientes. El cuestionario asociado a los elementos presentados (Mockups y Diagramas BPMN), estaba vinculado a reunir información general o macro asociada a la representatividad de los diagramas BPMN derivados. En éste último caso los Mockups se analizaron desde un punto de vista holístico.
e) Formulario Final: Un cuestionario que contenía preguntas vinculadas al cansancio de los sujetos asociado a la realización del experimento, y a la recopilación de información útil sobre fortalezas, debilidades y mejoras para el futuro de EUGEBP.

Los sujetos participantes fueron profesionales del área de la Ingeniería de Software: 11 con el grado de licenciados, y 5 de ellos con el grado de doctores. Todos provenientes de diferentes compañías e instituciones de software. En promedio, tenían 15 años de experiencia en programación y seis años en tareas de análisis de requerimientos.
Los sujetos participantes pertenecen a diferentes organizaciones en la que practican activamente la Ingeniería de Software en Argentina.
El protocolo del experimento fue ejecutado de la misma manera por todos los sujetos. En primer lugar, recibieron una breve introducción sobre EUGEBP y sobre el material que debía usarse durante el experimento. El material y el resultado del experimento están disponibles en línea en la web [20].

5.3 Evaluación de los resultados e implicaciones

La encuesta nos ayudó a evaluar la experiencia de los sujetos que utilizan nuestro enfoque. A continuación, respondemos las preguntas de investigación esbozadas al principio de esta sección.

RQ1: ¿Están de acuerdo los analistas con el diagrama BPMN derivado y sus elementos?
Evaluamos cuatro aspectos principales del diagrama BPMN: swim lanes, actividades, compuertas y flujos de secuencia. En primer lugar, analizamos las respuestas de los sujetos para los casos de uso de inicio de sesión, búsqueda de productos y pago. El 90% de los sujetos estuvo de acuerdo con la exactitud de los diagramas BPMN. Confirmaron que los swim lanes, las actividades, las compuertas y el flujo del proceso, mostrados en los diagramas eran correctos. Solo un sujeto no estaba de acuerdo con el diagrama BPMN derivado para los tres casos. Después de revisar la pregunta abierta para capturar la razón por la cual no estaba de acuerdo, desafortunadamente nos dimos cuenta de que rechazó el diagrama BPMN no solo por el diagrama sino también porque no estaba de acuerdo con la definición funcional del enfoque (que no es el enfoque de este trabajo).

RQ2: ¿Los analistas encuentran útil el diagrama BPMN derivado?
Con respecto al punto de vista cualitativo del enfoque, capturamos las siguientes fortalezas y debilidades desde las encuestas realizadas los sujetos:

[20] Objetos del experimento y resultados: https://tinyurl.com/rkvl3rt

Fortalezas:

- ✓ La derivación es sencilla si los Mockups están completamente anotados.
- ✓ Es fácil identificar los eventos del proceso.
- ✓ Los puntos fuertes son la facilidad con la que se pueden establecer requerimientos para trabajar junto con el cliente.
- ✓ Fácil comprensión.
- ✓ Representación gráfica.
- ✓ Los Mockups como herramienta gráfica permiten la participación de los stakeholders a través de anotaciones para identificar elementos BPMN.
- ✓ Los Mockups son intuitivos, ya que son aproximaciones visuales del sistema final.
- ✓ Independientemente del mecanismo específico utilizado para anotar el Mockup y derivar el diagrama BPMN, entiendo que para los usuarios participantes en la sesión de diseño es más fácil evaluar el flujo sobre el Mockup que utilizando las anotaciones. Por otro lado, para aquellos que diseñan / desarrollan software, el uso de diagramas BPMN es más efectivo / claro. Mantener ambos artefactos sincronizados es un reto interesante.

Debilidades:

- ✓ Se necesita capacitación.
- ✓ La debilidad, dependerá de la capacidad del analista para manejar el lenguaje natural de las anotaciones sobre los Mockups. Eso podría eliminar la claridad o la confusión, como casi cualquier herramienta de captura de requerimientos. Pero esto es precisamente porque es un método práctico e intuitivo.
- ✓ Práctica tediosa. Se necesita experiencia para estructurar coloquialmente las anotaciones de acuerdo al catálogo EUGEBP.
- ✓ Es necesario realizar muchas anotaciones y más precisas, lo que afecta la legibilidad del Mockup.

Desde el punto de vista del sujeto, el enfoque ha beneficiado la comunicación de los requerimientos de forma natural. Sin embargo, existen oportunidades para mejorar. Se podría por ejemplo proporcionar una herramienta que apoye el enfoque. O, se podrían minimizar las anotaciones.

Capítulo 6 Conclusiones

En éste capítulo final se muestran una serie de conclusiones vinculadas a los capítulos precedentes. Básicamente se presentan apreciaciones, generales y particulares, que son resultado del desarrollo de la tesis en sí misma. Además, en algunas de las mismas se dejan abiertas posibilidades de continuar o de derivar futuras investigaciones.

A continuación, se presentan las conclusiones del trabajo desarrollado.

6.1 Conclusiones generales

La totalidad de los autores consultados, en el ámbito de los procesos de negocio, coinciden en la importancia de los mismos dentro de las organizaciones. Mejorar la manera en que las cosas se hacen para el beneficio de todos los integrantes de una organización es el centro de la atención. Cuanto más tiempo se invierte en éste objetivo más maduros, repetibles y escalables son las operaciones de una organización [21]. Introducir procesos que permitan entrar en un círculo virtuoso de mejora continua a través del tiempo, es el desafío en el que se encuentran todas las organizaciones [19]. El término "mejora" puede tener diferentes significados: reducir costos, tiempos de ejecución, tasas de error, obtener ventajas competitivas, entre otros. Sin embargo, el termino más bien se refiere a administrar: cadenas enteras de eventos, actividades y decisiones, de manera tal que la misma agregue valor a la organización [4].

Modelar un proceso de negocio significa comprender el mismo y compartir dicha comprensión con las personas que participan en él a diario. Los participantes de un proceso de negocio generalmente realizan actividades bastante especializadas y difícilmente se enfrentan a toda la complejidad del mismo. En éste contexto, los modelos de procesos de negocio nos ayudan a comprender, identificar y prevenir problemas dentro de los mismos [4]. El estándar “Business Process Model and Notation (BPMN)” tiene como objetivo principal proporcionar una notación que sea fácil de comprender para todas las personas que intervienen en un proceso de negocio y, crea un puente entre el diseño del proceso y la implementación del mismo [23]. Es decir que facilita la comunicación rápida, fluida y expresiva con cualquier socio del negocio (cliente, consultor, proveedor, etc.) que conozca y comprenda el estándar. Institutos de capacitación, universidades, y empresas consultoras invierten recursos para formar profesionales en dicha notación [19]. Con más de 100 símbolos, BPMN es un lenguaje bastante nutrido. Sin embargo, conocer un subconjunto de esos símbolos permite cubrir muchas de sus necesidades de modelado básicas.

El “Natural Language Processing – Procesamiento del Lenguaje Natural (NLP)” es el estudio del modelado matemático y computacional de varios aspectos del lenguaje y el desarrollo de una amplia gama de sistemas. La investigación en NLP es altamente interdisciplinaria e involucra conceptos de informática, lingüística e inteligencia artificial, entre otros. El idioma (hablado y escrito) es fundamental para todos los aspectos de la comunicación de las personas y los sistemas NLP actualmente juegan un papel crucial en dicha comunicación. Los sistemas de procesamiento de texto y comprensión de mensajes, por ejemplo, son útiles para extraer información de textos y formatearla de varias maneras para su aprovechamiento posterior [5]. El análisis del lenguaje natural, proporcionado en forma de texto, es el campo que se puede explorar en mayor profundidad.

Dado que el campo de los modelos de procesos de negocio contiene mucha información en lenguaje natural (escrito y hablado), el mismo califica como un campo de aplicación de NLP. El rol del lenguaje natural, compuesto por su dimensión sintáctica y semántica, dentro de los modelos de proceso es un campo de investigación de mucha importancia [24]. Muchos autores coinciden en que la generación de modelos de procesos de negocio a partir del procesamiento de textos en lenguaje natural se puede lograr teniendo presente una serie de requisitos sintácticos y semánticos que el texto proporcionado

debe cumplir [6][7][8][9][10]. Este contexto, con base en el estudio del procesamiento del lenguaje natural, existe un precedente que propone reglas de mapeo para identificar elementos de procesos de negocio en textos [7].

El Mockup o Wireframe es un prototipo de interfaz de usuario final (un bosquejo rápido). El mismo permite descubrir y definir requerimientos, que no son del ámbito de la interfaz de usuario, en un lenguaje natural de fácil comprensión. Y colabora con la etapa de elicitación de requerimientos [49]. El enfoque "Mockup Driven Development - Desarrollo Dirigido por Mockups (MockupDD)" [15] tiene como artefacto principal al Mockup y enriquece al mismo con serie de etiquetas. Dichas etiquetas ayudan a especificar cuestiones específicas de navegación o de manipulación de datos. Luego, la semántica de las etiquetas puede ser usada para: descubrir y mapear requerimientos, y especificar como la aplicación final debe comportarse. Con la participación de los usuarios finales.

La técnica denominada "End User Grammar – Gramática de usuario Final" (EUG) propone una notación coloquial (catálogo de anotaciones) y amigable para describir datos, navegación, requisitos de interacción y de negocios sobre Mockups [20]. La misma depende de MockupDD y se centra en mejorar la recopilación de requerimientos mediante anotaciones fáciles de usar. Cada anotación realizada sobre él Mockup es una definición coloquial estructurada legible para cualquier usuario final y no presenta ningún concepto técnico que pueda limitar su entendimiento.

El presente trabajo de tesis combina todos los conceptos mencionados en los párrafos anteriores:

- La importancia de conocer los procesos de negocio de una organización y el valor agregado que estos proporcionan a la misma (BP).
- El impacto que tiene modelar un proceso de negocio para los integrantes de una organización, utilizando una notación estándar común a todos los usuarios (BPMN).
- La aplicación del análisis y el procesamiento del lenguaje natural a los textos vinculados a los procesos de negocio (NLP).
- El rol del lenguaje natural dentro de los procesos de negocio (Lingüistica).
- La producción de texto en lenguaje natural, sobre bosquejos de las interfaces de usuario, con oraciones simples (EUG).
- La identificación de elementos pertenecientes a los procesos de negocios en textos en lenguaje natural (Enfoque semiautomático).

El resultado de la combinación mencionada precedentemente es la Técnica EUGEBP.

6.2 Conclusiones particulares

El campo de investigación de los procesos de negocio contiene una vasta extensión de contenidos. El estándar BPMN es solo uno de los mismos y está compuesto a su vez de un amplio conjunto de símbolos como para ser abordado en un solo trabajo de tesis. Por éste motivo, en la sección "3.3 Limitaciones: Elementos BPMN no considerados" se aclara que existe una acotación. Sin embargo, en la sección "3.2 Elementos BPMN a identificar" se especifica cuáles son los símbolos BPMN a considerar y los mismos permiten cubrir las necesidades de modelado propuestas por la tesis.

Uno de los trabajos de investigación que tuvo una fuerte incidencia en la presente tesis fue el "enfoque semiautomático para identificar elementos de BP en textos en lenguaje natural" [7]. El mismo propone analizar el texto existente en las organizaciones para colaborar con la identificación de los elementos BPMN implícitos en el texto y por ende en la organización. Dicho análisis se apoya en el campo de investigación del procesamiento del leguaje natural (NLP). Mediante la aplicación del mismo se identifican los elementos BPMN necesarios para conformar luego, los diagramas BPMN que los contienen.

Otro de los trabajos de investigación que influyó de manera decisiva, en el desarrollo de la tesis, fue la técnica "EUG" [20]. La misma es una gramática coloquial y amigable diseñada para describir: datos, navegación, requisitos de interacción y de negocios, sobre los Mockups. Propone un enfoque que se centra en mejorar la recopilación de requerimientos mediante anotaciones fáciles de usar y depende de MockupDD.

La técnica EUGEBP, al igual que EUG depende del enfoque MockupDD y es una gramática amigable, pero está diseñada para descubrir procesos de negocios. De allí toma la idea de producir anotaciones sobre los Mockups, pero adopta una gramática para dicho propósito que sienta su base en los conceptos ligados al procesamiento del lenguaje natural (NLP). Se apoya en los mismos y los explota a su favor.

EUGEBP es una técnica que propone lo siguiente:

i) un conjunto de reglas para la producción de anotaciones sobre sobre los Mockups,
ii) un conjunto de pasos que se apoya en las reglas mencionadas y colabora con la derivación de diagramas BPMN desde los Mockups.

La técnica no analiza el texto existente en las organizaciones, vinculado a los BP de la misma, sino que ayuda a producirlo. El "*Proceso de Producción Anotaciones EUGEBP*" se realiza guiado por el conjunto de reglas propuesto (sección "3.5"). Dicha producción debe ser trabajada por los stakeholders y los analistas del equipo de desarrollo. Los primeros son quienes conocen los procesos de la organización. Y, los segundos son quienes pueden traducir el conocimiento en oraciones simples que pueden ser aprovechadas para descubrir los procesos de la organización.

La "Técnica EUGEBP" se desarrolla a través de un ejemplo. El mismo muestra el proceso de derivación manual de un diagrama BPMN a partir de un Mockup comentado con anotaciones EUGEBP (ver figura 53). Dicho proceso es manual, es realizado por el equipo de desarrollo, y se basa en la comprensión de oraciones simples. Cualquier ser humano, desde muy temprana edad posee la capacidad de comprender oraciones simples gramaticalmente correctas. La técnica EUGEBP se apoya fuertemente en ésta capacidad de las personas. Por ende, la interpretación y comprensión de las oraciones, llevada a cabo por el equipo de analistas requiere un mínimo esfuerzo por parte de los mismos.

El propósito de la técnica se resume visualmente en la siguiente imagen:

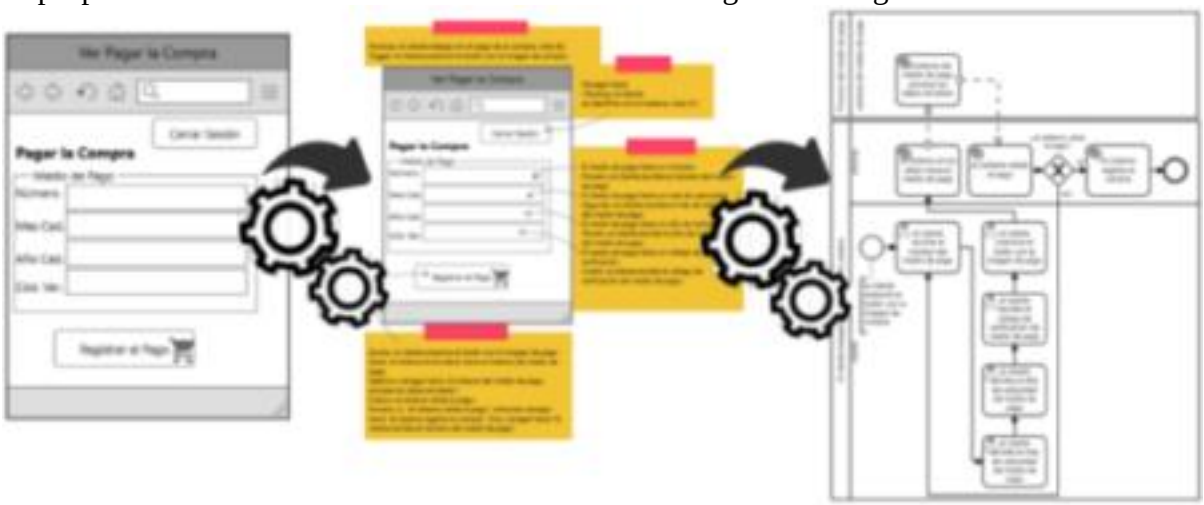


Figura 53 – Técnica EUGEBP

6.3 Futuras líneas de investigación

En la tesis se mencionó que en las primeras etapas del proceso de desarrollo de software el conocimiento sobre los BP de la organización puede ser limitado o nulo. Y que, además, en ocasiones los integrantes del equipo de desarrollo desconocen los mismos. La técnica EUGEBP está diseñada para colaborar con la fase del "Levantamiento del Proceso" y ayuda a los analistas a conocer en mayor detalle los BP de la organización en los que interviene la aplicación. EUGEBP colabora con la comprensión de los BP. Y dicha colaboración puede explotarse aún más si se amplía con una futura investigación lo presentado en la sección "3.7 Identificación del Proceso de Negocio Superior".

Luego de haber aplicado anotaciones EUGEBP sobre un Mockup la tarea de identificar los elementos BPMN y de derivar su correspondiente diagrama BPMN es sencilla de realizar. Las anotaciones EUGEBP facilitan dicha derivación. Sin embargo, no todos los elementos BPMN que existen en dicha notación son cubiertos por la tesis. Los elementos restantes podrían ser cubiertos en una futura ampliación de la investigación.

Las tareas de: "Producción de Anotaciones EUGEBP", "División de Anotaciones EUGEBP" y "Extracción de Elementos BPMN" pueden ser realizadas por el equipo de analistas manualmente. Sin embargo, las mismas también pueden ser asistidas y complementadas con herramientas NLP (por ejemplo, Stanford CoreNLP). Esta última mención es una idea que se investigó con mayor profundidad en capítulo 4 y en el mismo queda sentada una fuerte base técnica que demuestra que la automatización de dichas tareas es factible de realizar y puede ampliarse en mayor profundidad en una futura investigación.

6.4 Conclusión Final

En la presente tesis hemos presentado un enfoque novedoso para derivar diagramas BPMN desde Mockups anotados con EUGEBP. Con éste enfoque utilizamos una forma de especificar requerimientos fácil y amigable para usuario final que además permite identificar elementos BPMN y los diagramas BPMN que los contienen. Las anotaciones son oraciones simples (gramaticalmente correctas) en lenguaje natural que son legibles tanto para el analista como para el usuario final. Las mismas pueden procesarse utilizando herramientas NLP tales como Stanford CoreNLP. Describimos los pasos más importantes de nuestro enfoque utilizando un ejemplo simple pero ilustrativo. Luego, realizamos una evaluación para apreciar los beneficios de nuestro enfoque. El resultado preliminar permite comprender los temas y el enfoque efectivo para detectar modelos de proceso de negocio.
Planeamos extender el soporte de diferentes elementos BPMN no cubiertos en este trabajo. Además, planeamos una evaluación más formal si este enfoque utiliza software empírico.

Anexo 1 – Derivación del diagrama BPMN vinculado al "Mockup, el cliente se identifica con el sistema, vista (1)"

Paso 1) Proceso de Producción de Anotaciones EUGEBP
Ver figura 16

Paso 2) Proceso de División de Anotaciones EUGEBP

Oración EUGEBP
Mockup, el cliente se identifica con el sistema, vista (1).
Trigger: El cliente presionó el botón de acceso.
El cliente tiene un correo electrónico.
Primero, el cliente escribe su correo electrónico.
El cliente tiene una contraseña.
Segundo, el cliente escribe su contraseña.
Tercero, el cliente presiona el botón de verificación de identidad.
Cuarto, navegar hacia "el sistema identifica el cliente".
El cliente presionó el botón de autenticación.
Quinto, si "el sistema identifica al cliente", entonces, navegar hacia "Mockup, el cliente trabaja con el carro de compra, vista (5)". Sino, navegar hacia "el cliente escribe su correo electrónico".
Navegar hacia "Mockup, el cliente registra sus datos, vista (2)".

Tabla 31 – Resultado de la división de las anotaciones EUGEBP en oraciones

Paso 3) Proceso de Extracción de Elementos BPMN

Oración EUGEBP	**Nº Regla EUGEBP**	**Elemento BPMN**
Mockup, el cliente se identifica con el sistema, vista (1).	5 1 2	**Sub-proceso**: el cliente se identifica con el sistema. **SwimLane**: cliente. **SwimLane**: sistema.
Trigger: el cliente presionó el botón de acceso.	8 1	**Evento Inicial**: el cliente presionó el botón de acceso. **SwimLane**: cliente.
El cliente tiene un correo electrónico.	3	**SwimLane**: cliente.
Primero, el cliente escribe su correo electrónico.	13 1	**Flujo de Secuencia**. Origen: **Evento Inicial,** el cliente presionó el botón de acceso. Destino: **Actividad de usuario**, el cliente escribe su correo electrónico. **SwimLane**: cliente.
El cliente tiene una contraseña.	3	**SwimLane**: cliente.
Segundo, el cliente escribe su contraseña.	13 1	**Flujo de Secuencia**. Origen: **Actividad de usuario**, el cliente escribe su correo electrónico. Destino: **Actividad de usuario**, el cliente escribe su contraseña. **SwimLane**: cliente.

Tercero, el cliente presiona el botón de verificación de identidad.	13 1	**Flujo de Secuencia**. Origen: **Actividad de usuario**, el cliente escribe su correo contraseña. Destino: **Actividad de usuario**, el cliente presiona el botón de verificación de identidad. **SwimLane**: cliente.
Cuarto, navegar hacia "el sistema identifica el cliente".	13 1 2	**Flujo de Secuencia**. Origen: **Actividad de usuario:** el cliente presiona el botón de verificación de identidad. Destino: **Actividad de servicio,** el sistema identifica al cliente. **SwimLane**: cliente. **SwimLane**: sistema.
Quinto, si "el sistema identifica al cliente", entonces, navegar hacia "Mockup, el cliente trabaja con el carro de compra, vista (5)". Sino, navegar hacia "el cliente escribe su correo electrónico".	10 12 14	**Flujo de Secuencia**: Origen: **Actividad de servicio,** el sistema identifica al cliente. Destino: **Compuerta XOR,** ¿el sistema identifica al cliente? **Flujo de Secuencia**: Origen: **Compuerta XOR,** ¿el sistema identifica al cliente? Destino: **Sub-proceso**, el cliente trabaja con el carro de compra. **Flujo de Secuencia**: Origen: **Compuerta XOR,** ¿el sistema identifica al cliente? Destino: **Actividad de usuario**, el cliente escribe su correo electrónico.
Navegar hacia "Mockup, el cliente registra sus datos, vista (2)".	11	**Flujo de Secuencia:** Origen: **Sub-proceso**, el cliente se identifica con el sistema. Destino: **Sub-proceso**, el cliente registra sus datos.

Tabla 32 – Tabla de Extracción de Elementos BPMN

El resultado del "*Proceso de Extracción de Elementos BPMN*" son los elementos BPMN identificados. A continuación, mostramos los mismos sin repeticiones:

SwimLanes:
- cliente
- sistema

Evento Inicial:
- el cliente presionó el botón con la imagen de acceso

Sub-Procesos:
- el cliente trabaja con el carro de compra
- el cliente registra sus datos

Actividades de usuario:
- el cliente escribe su correo electrónico
- el cliente escribe su correo contraseña
- el cliente presiona el botón de verificación de identidad

Actividad de servicio:
- el sistema identifica al cliente

Compuerta XOR:
¿el sistema identifica al cliente?

Flujos de Secuencia o de Mensaje:

Flujos de Secuencia o de Mensaje	
Origen	**Destino**
el cliente presionó el botón de acceso	el cliente escribe su correo electrónico
el cliente escribe su correo electrónico	el cliente escribe su contraseña
el cliente escribe su contraseña	el cliente presiona el botón de verificación de identidad
el cliente presiona el botón de verificación de identidad	el sistema identifica al cliente
el sistema identifica al cliente	¿el sistema identifica al cliente?
¿el sistema identifica al cliente?	el cliente trabaja con el carro de compra
¿el sistema identifica al cliente?	el cliente escribe su correo electrónico
el cliente se identifica con el sistema	el cliente registra sus datos

Tabla 33 – Flujos de Secuencia o de Mensaje identificados en el Mockup

Paso 4) Proceso de Identificación del Flujo Normal

Primero
Se filtra el primer Evento Inicial.

Evento Inicial: el cliente presionó el botón con la imagen de acceso.

Segundo
Se filtran las oraciones que contienen adjetivos ordinales:

- Primero, el cliente escribe su correo electrónico.
- Segundo, el cliente escribe su contraseña.
- Tercero, el cliente presiona el botón de verificación de identidad.
- Cuarto, navegar hacia “el sistema identifica el cliente”.
- Quinto, si “el sistema identifica al cliente”, entonces, navegar hacia “Mockup, el cliente trabaja con el carro de compra, vista (5)”. Sino, navegar hacia “el cliente escribe su correo electrónico”.

Tercero
En tercer lugar, el equipo extrae todos los “Objetos de Flujo BPMN” contenidos en las oraciones filtradas del paso dos:

- **Actividad de Usuario:**
  el cliente escribe su correo electrónico
  el cliente escribe su contraseña.
  e cliente presiona el botón de verificación de identidad
- **Actividad de Servicio:**
  el sistema identifica el cliente
- **Subproceso:**
  el cliente trabaja con el carro de compra
- **Compuerta XOR:**
  ¿el sistema identifica el cliente?

Cuarto
“SwimLanes BPMN” contenidos en las oraciones filtradas:

- **SwimLanes**:
  cliente
  sistema

Quinto

El equipo filtra desde "**Flujos de Secuencias y de Mensajes**" aquellos flujos que contengan (en el origen o en el destino) aquellos "Objetos de Flujo" filtrados en el paso tres (ver tabla 34):

Flujos de Secuencia o de Mensaje	
Origen	**Destino**
el cliente presionó el botón de acceso	*el cliente escribe su correo electrónico*
el cliente escribe su correo electrónico	*el cliente escribe su contraseña*
el cliente escribe su contraseña	*el cliente presiona el botón de verificación de identidad*
el cliente presiona el botón de verificación de identidad	*el sistema identifica al cliente*
el sistema identifica al cliente	*¿el sistema identifica al cliente?*
¿el sistema identifica al cliente?	*el cliente trabaja con el carro de compra*
¿el sistema identifica al cliente?	*el cliente escribe su correo electrónico*

Tabla 34 – Flujos Normales de Secuencia o de Mensaje identificados en el Mockup

Sexto

El equipo infiere el diagrama de procesos de negocios BPMN (ver figura 54):

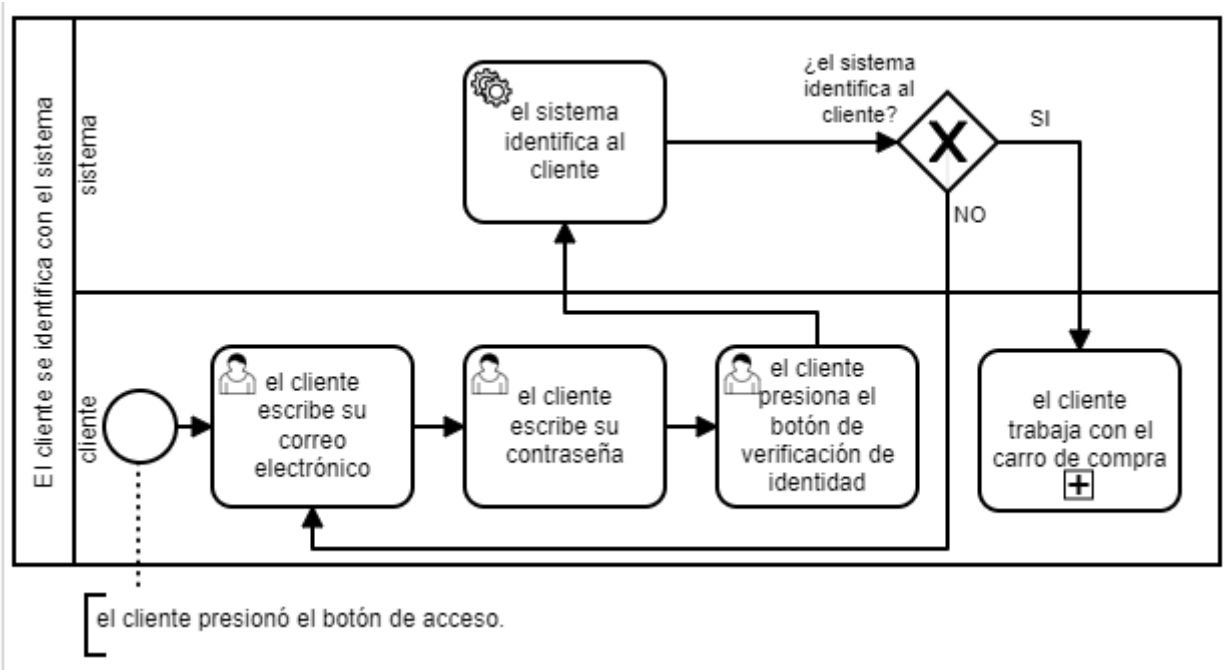


Figura 54 – Diagrama BPMN inferido desde las anotaciones EUGEBP

El "Evento Final" simplemente lo agregamos al final. (ver figura 55):

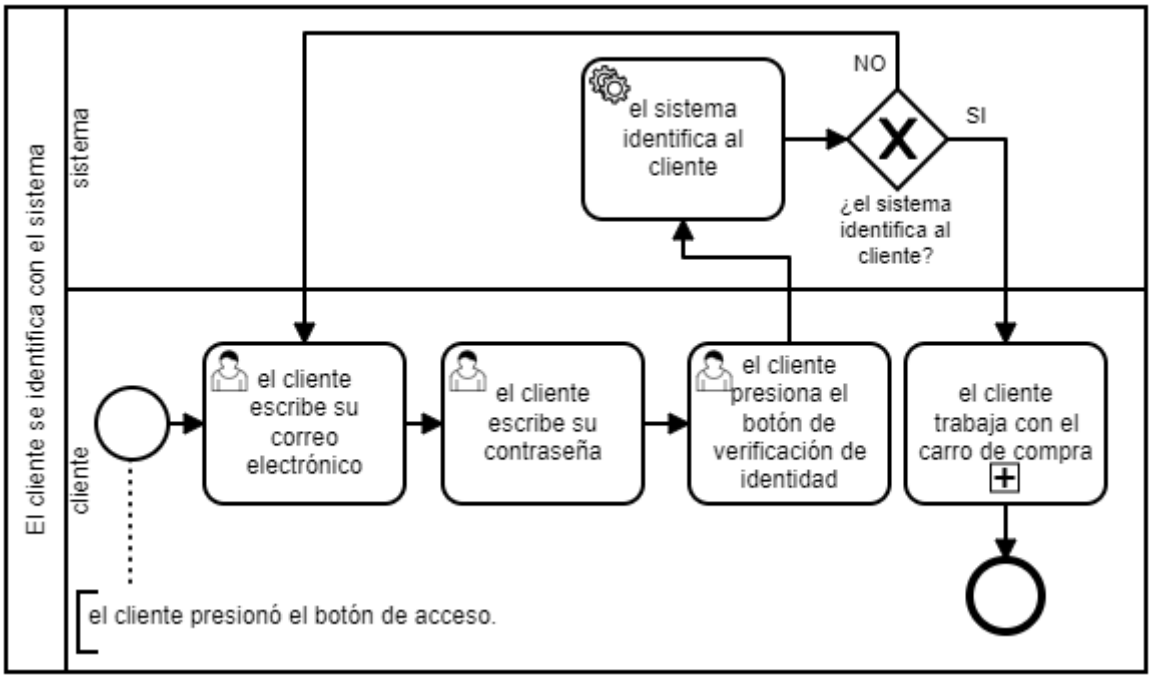


Figura 55 – Diagrama BPMN inferido desde las anotaciones EUGEBP final

Anexo 2 – Analizadores PEG.js

Regla 5: [Sujeto] + [Verbo en tiempo presente] + [Objeto].

```
anotacion = actitividadMockup
actitividadMockup = prefijoOracion espacio strOS: oracionSimple prefijoVista strN: numero posfijoVista {
        return{
        vista:{
        numero: strN
      },
        oracionSimple:{
        customData: strOS.toString().split(",").join("")
      },
      elementoBPMN:"actitividadMockup"
   }
}
prefijoOracion = "Mockup,"
prefijoVista = ", vista ("
posfijoVista = ")."
oracionSimple = p:(palabra espacio?)*
palabra = letra+
letra = [a-zA-Z0-9]
numero = [0-9]
espacio = (" " / "\n" / "\r")*
```

Regla 6: E/el "sistema" [Verbo en tiempo presente] + [Objeto].

```
anotacion = actividadElSistemaVerboPObjeto
actividadElSistemaVerboPObjeto = strOS: oracionSimple {
        return{
        oracionSimple:{
        customData: strOS.toString().replace(/['"]+/g, '').split(",").join("")
      },
      elementoBPMN:"actividadElSistemaVerboPObjeto"
   }
}
oracionSimple = prefijoNucleo unEspacio comilla sistema comilla unEspacio restoOracion posfijoRestoOracion
prefijoNucleo = ("El" / "el")
posfijoRestoOracion = "."
comilla = "\""
sistema = "sistema"
restoOracion = p:(palabra espacio?)*
palabra = letra+
letra = [a-zA-ZÀ-ÿ\u00f1\u00d1]
numero = [0-9]
espacio = [ \t\n\r]*
unEspacio = " "
```

Regla 7: [Sujeto] + [Verbo en pretérito perfecto simple] + [Objeto].

```
anotacion = oracionSimple
oracionSimple = strOS: oracionSimpleEnTiempoPasado {
        return{
        oracion:{
        customData: strOS.toString().replace(/['"]+/g, '').split(',').join('')
      },
      elementoBPMN:"actividadTiempoPresente"
   }
}
oracionSimpleEnTiempoPasado = sujetoOracion unEspacio verboEnTiempoPasado unEspacio objetoOracion posfijoOracion
sujetoOracion = articulo unEspacio sustantivo
articulo = ("El" / "el" / "La" / "la" / "Los" / "los" / "Las" / "las" / "Un" / "un" / "Una" / "una" / "Unos" / "unos" / "Unas" / "unas")
sustantivo = palabra
verboEnTiempoPasado = palabra
objetoOracion = oracion
posfijoOracion = "."
oracion = p:(palabra espacio?)*
palabra = letra+
letra = [a-zA-ZÀ-ÿ\u00f1\u00d1]
numero = [0-9]
espacio = [ \t\n\r]*
unEspacio = " "
```

Regla 8: Trigger: [Sujeto] + [Verbo en pretérito perfecto simple] + [Objeto].

```
anotacion = triggerOracion
triggerOracion = prefijoOracion unEspacio strOS: oracionSimple {
        return{
        oracion:{
        customData: strOS.toString().replace(/['"]+/g, '').split(',').join('')
      },
      elementoBPMN:"eventoTrigguer"
   }
}
oracionSimple = oracion posfijoOracion
prefijoOracion = "Trigger:"
posfijoOracion = "."
oracion = p:(palabra espacio?)*
palabra = letra+
letra = [a-zA-ZÀ-ÿ\u00f1\u00d1
numero = [0-9]
espacio = [ \t\n\r]*
unEspacio = " "
```

Regla 10: [Palabra de exclusión] + “[Actividad/Evento]”, + [Conector de consecuencia], + navegar hacia “[Actividad/Evento]”. Sino, navegar hacia “[Actividad/Evento]”.

```
anotacion = parrafoSimple
parrafoSimple = strOS: parrafoSimpleXOR
parrafoSimpleXOR = palabraDeExclusion unaComa unEspacio unaComilla
strOS:actividadEventoP unaComilla unaComa unEspacio conectorDeConsecuencia unaComa
unEspacio navegarHacia unEspacio unaComilla actividadEvento unaComilla unPunto unEspacio
sino unaComa unEspacio unaComilla actividadEvento unaComilla unPunto {
        return{
        oracion:{
        customData: "¿"+strOS.toString().replace(/['"]+/g, '').split(',').join('')+"?"
      },
      elementoBPMN:"compuertaXOR"
   }
}
palabraDeExclusion = "Si"
unEspacio = " "
unaComa = ","
unaComilla = "\""
unPunto = "."
actividadEventoP = oracionSimpleEnTiempoPresente
actividadEvento = oracionSimpleEnTiempoPresente
conectorDeConsecuencia = ("como consecuencia" / "como resultado" / "entonces" / "luego" / "por
consiguiente")
navegarHacia = "navegar hacia"
sino = "Sino"
verboEnTiempoPasado = palabra
objetoOracion = oracion
posfijoOracion = "."
oracion = p:(palabra espacio?)*
palabra = letra+
letra = [a-zA-ZÀ-ÿ\u00f1\u00d1
numero = [0-9]
espacio = [ \t\n\r]*
oracionSimpleEnTiempoPresente = sujetoOracion unEspacio verboEnTiempoPresente unEspacio
objetoOracion
sujetoOracion = articulo unEspacio sustantivo
articulo = ("El" / "el" / "La" / "la" / "Los" / "los" / "Las" / "las" / "Un" / "un" / "Una" / "una" /
"Unos" / "unos" / "Unas" / "unas")
sustantivo = palabra
verboEnTiempoPresente = palabra
```

Regla 12: Navegar hacia “[Actividad]”.

```
etiqueta = "mn:" espacio mockupNombre / "nh:" espacio navegarHacia
mockupNombre = nombreMockup: palabras guion vistaNumero:referenciaVista {
   return{
      tag:{
         widget: {
            customData: nombreMockup.toString().split(",").join("")+ vistaNumero.join("")
         },
         expresion: 'nombreMockup'
      }
   }
 }
navegarHacia = navigate: ("Navegar hacia" espacio) destino: mockupNombre{
   return{
      tag:{
         widget: {
            customData: navigate.toString().split(",").join("")+ destino.tag.widget.customData,
            link: true
         },
         expresion: 'navegacion'
      }
   }
}
palabras = p:(palabra espacio?)*
guion = "-" espacio
referenciaVista = "Vista (" numero ")"
palabra = letra+
letra = [a-zA-Z0-9]
numero = [0-9]
espacio = (" " / "\n" / "\r")*
```

Anexo 3 - Clase JAVA, que hace uso de Stanford CoreNLP DependencyParser

```
import edu.stanford.nlp.ling.*;
import edu.stanford.nlp.io.IOUtils;
import edu.stanford.nlp.pipeline.*;
import edu.stanford.nlp.semgraph.*;
import edu.stanford.nlp.trees.*;
import edu.stanford.nlp.util.Pair;
import java.io.IOException;
import java.util.*;
public class BasicPipelineExample2 {
	public static String text =	"El cliente escribe su nombre. " +
					"El sistema identifica al cliente. ";
	public static void main(String[] args) throws IOException {
	    // configuro las propiedades del pipeline
	    Properties props = new Properties();
	    // configuro las propiedades del pipeline en español
	    props.load(IOUtils.readerFromString("StanfordCoreNLP-spanish.properties"));
	                // configuro una propiedad para un anotador
	       props.setProperty("coref.algorithm", "neural");
	    // configuro una lista de anotadores para correr
	    props.setProperty("annotators", "tokenize,ssplit,pos,parse,depparse");
	    StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
	    // creao un objeto documento
	    CoreDocument document = new CoreDocument(text);
	    // anoto el documento
	    pipeline.annotate(document);
	    // obtengo el texto de una oración
	    String sentenceText = document.sentences().get(0).text();
	    System.out.println("Ejemplo: oración ");
	    System.out.println(sentenceText);
	    // obtengo el objeto oración
	    CoreSentence sentence = document.sentences().get(0);
	    // analizo las dependencias para la oración
	    SemanticGraph dependencyParse = sentence.dependencyParse();
	    System.out.println("Ejemplo: análisis de dependencia");
	    System.out.println(dependencyParse);
	    Boolean poseeVerbo = false;
	    Boolean poseeSujeto = false;
	    Boolean poseeObjeto = false;
	    IndexedWord root = dependencyParse.getFirstRoot();
	    String strParteDeLaOracion = null;
	    String strPalabra = null;
	    String strEtiqueta = root.tag().toString().trim();
	    switch(strEtiqueta)
	    {
	       case "VERB": //VERBO DE LA ORACION
	              poseeVerbo = true;
	              strPalabra = root.value();
	              strParteDeLaOracion = "VERBO DE LA ORACION: "+strPalabra;
	              System.out.println(strParteDeLaOracion);//VERBO DE LA ORACION
	              break;
	    }
	    List<Pair<GrammaticalRelation,IndexedWord>> ramasHijas =
dependencyParse.childPairs(root);
	    String strGrammaticalRelation = null;
	    for(Pair<GrammaticalRelation,IndexedWord> rama : ramasHijas) {
	       strParteDeLaOracion = null;
	       strPalabra = rama.second.word();
	       strGrammaticalRelation = rama.first.toString();
	        switch(strGrammaticalRelation)
```

```
                {
                    case "nsubj": //SUJETO DE LA ORACION
                     poseeSujeto = true;
                     strParteDeLaOracion = "SUJETO DE LA ORACION: "+strPalabra;
                        break;
                    case "obj": //OBJETO DE LA ORACION
                     poseeObjeto = true;
                     strParteDeLaOracion = "OBJETO DE LA ORACION: "+strPalabra;
                        break;
                    case "punct": //FINAL DE LA ORACION
                     strParteDeLaOracion = "FINAL DE LA ORACION: "+strPalabra;
                        break;
                    default:
                        System.out.println("no match");
                }
                System.out.println(strParteDeLaOracion);
            }
            if(poseeVerbo && poseeSujeto && poseeObjeto) {
               System.out.println("SI es una oración simple gramaticalmente correcta.");
            }
            else {
               System.out.println("NO es una oración simple gramaticalmente correcta.");
            }
        }
}
```

El producto de la ejecución de las precedentes líneas de código es el siguiente:

```
[main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator tokenize
[main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator ssplit
[main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator pos
[main] INFO edu.stanford.nlp.tagger.maxent.MaxentTagger - Loading POS tagger from
edu/stanford/nlp/models/pos-tagger/spanish-ud.tagger ... done [0.5 sec].
[main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator parse
[main] INFO edu.stanford.nlp.parser.common.ParserGrammar - Loading parser from
serialized file edu/stanford/nlp/models/srparser/spanishSR.beam.ser.gz ... done [23.6
sec].
[main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator depparse
[main] INFO edu.stanford.nlp.parser.nndep.DependencyParser - Loading depparse model:
edu/stanford/nlp/models/parser/nndep/UD_Spanish.gz ...
[main] INFO edu.stanford.nlp.parser.nndep.DependencyParser - Done reading from disk ...
Time elapsed: 3.3 sec
[main] INFO edu.stanford.nlp.parser.nndep.Classifier - PreComputed 99990, Elapsed Time:
25.935 (s)
[main] INFO edu.stanford.nlp.parser.nndep.DependencyParser - Initializing dependency
parser ... done [29.2 sec].
Ejemplo: oración
El cliente escribe su nombre.
Ejemplo: análisis de dependencia
-> escribe/VERB (root)
  -> cliente/NOUN (nsubj)
    -> El/DET (det)
  -> nombre/NOUN (obj)
    -> su/DET (det)
  -> ./PUNCT (punct)
VERBO DE LA ORACION: escribe
SUJETO DE LA ORACION: cliente
OBJETO DE LA ORACION: nombre
FINAL DE LA ORACION: .
SI es una oración simple gramaticalmente correcta.
```

Referencias

[1] U. Dayal, M. Hsu, and R. Ladin, “Business Process Coordination: State of the Art, Trends, and Open Issues,” Sep. 2001.

[2] P. Harmon, “The future of business process management,” *Bus. Process Chang.*, pp. 441–455, 2019.

[3] V. Kale and V. Kale, “Business Process Modeling and Notation,” *Enterp. Process Manag. Syst.*, no. January, pp. 257–275, 2019.

[4] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, “Fundamentals of business process management.” 2018.

[5] A. K. JOSHI, “Natural Language Processing,” *Science (80-.).*, vol. 253, no. 5025, pp. 1242–1249, 1991.

[6] F. Pittke, H. Leopold, and J. Mendling, “Automatic detection and resolution of lexical ambiguity in process models (extended abstract),” *Lect. Notes Informatics (LNI), Proc. - Ser. Gesellschaft fur Inform.*, vol. P252, no. 6, pp. 526–544, 2016.

[7] R. C. B. Ferreira, L. H. Thom, and M. Fantinato, “A Semi-automatic Approach to Identify Business Process Elements in Natural Language Texts,” no. April, pp. 250–261, 2017.

[8] N. M. Maruai, M. S. M. Ali, M. H. Ismail, S. A. Z. S. Shaikh, M. Shirakashi, and S. Muhamad, “Comparative study on energy extraction from vibrating square cylinder,” *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 8, pp. 2581–2587, 2017.

[9] E. E.V.a, M.-R. P.b, H. C.a, D. R.a, and S. C.a, “Automatic process model discovery from textual methodologies: An archaeology case study,” *Proc. - Int. Conf. Res. Challenges Inf. Sci.*, vol. 2015-June, no. June, pp. 19–30, 2015.

[10] J. Mendling, H. A. Reijers, and W. M. P. van der Aalst, “Seven process modeling guidelines (7PMG),” *Inf. Softw. Technol.*, vol. 52, no. 2, pp. 127–136, 2010.

[11] B. Maqbool *et al.*, *Information Science and Applications 2018*, vol. 514. Springer Singapore, 2019.

[12] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, “Empirical Study of Agile Software Development Methodologies,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 40, no. 1, pp. 1–6, 2015.

[13] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano, *On the effectiveness of screen mockups in requirements engineering: Results from an internal replication*. 2010.

[14] J. J. Garrett, *The Elements of User Experience: User-Centered Design for the Web and Beyond*, 2nd ed. USA: New Riders Publishing, 2010.

[15] J. M. Rivero, J. Grigera, G. Rossi, E. Robles Luna, F. Montero, and M. Gaedke, “Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering,” *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 670–687, 2014.

[16] J. M. Rivero, E. R. Luna, J. Grigera, and G. Rossi, “Improving user involvement through a model-driven requirements approach,” *2013 3rd Int. Work. Model. Requir. Eng. MoDRE 2013 - Proc.*, pp. 20–29, 2013.

[17] R. R. Young, “The requirements engineering handbook.” Artech House, Boston, 2004.

[18] M. Dumas, M. La Rosa, J. Mendling, R. Mäesalu, H. A. Reijers, and N. Semenenko, “Understanding Business Process Models: The Costs and Benefits of Structuredness,” in *CAiSE*, 2012.

[19] J. Freund, B. Rücker, and B. Hitpass, *BPMN 2.0: Manual de Referencia y Guía Práctica*. 2011.

[20] M. Urbieta, N. Torres, J. M. Rivero, and G. Rossi, “Improving Mockup-based Requirement Specification Quality with End-User Annotations,” pp. 1–14.

[21] S. A. White and D. Miers, *BPMN guía de referencia y modelado : comprendiendo y utilizando BPMN : desarrolle representaciones grá ficas de procesos de negocios, que sean rigurosas pero al mismo tiempo de fácil comprensión*. Lighthouse Point, Fla.: Future Strategies Inc., 2010.

[22] Y. G. OJEDA and E. V. GARCÍA, “Guía Para La Identificación Y Análisis De Los Procesos De La Universidad De Málaga,” *Univ. Malaga*, p. 40, 2008.
[23] Object Management Group (OMG), “Business Process Model and Notation , V1.1,” no. January, p. 294, 2008.
[24] H. Leopold, *Natural Language in Business Process Models*, vol. 45, no. 8. Springer, 2013.
[25] M. B. Maria Antònia Martí, Mariona Taulé and and L. Màrquez, “AnCora: Multilingual an d Multilevel Annotate d Corpora,” vol. 2006, no. 1, pp. 2–6, 2006.
[26] G. Miller, C. Fellbaum, J. Kegl, and K. Miller, “WordNet: An Electronic Lexical Reference System Based on Theories of Lexical Memory,” *Rev. québécoise Linguist.*, vol. 17, no. 2, pp. 181–212, Mar. 1988.
[27] Q. Y. B, M. Ge, and M. Helfert, “Data Quality Problems in TPC-DI Based,” vol. 1, no. August, pp. 57–73, 2018.
[28] A. Ghose, G. Koliadis, and A. Chueng, “Process discovery from model and text artefacts,” *Proc. - 2007 IEEE Congr. Serv. Serv. 2007*, pp. 167–174, 2007.
[29] A. Sinha and A. Paradkar, “Use cases to process specifications in business process modeling notation,” *ICWS 2010 - 2010 IEEE 8th Int. Conf. Web Serv.*, pp. 473–480, 2010.
[30] A. Gangopadhyay, “Conceptual modeling from natural language functional specifications,” *Artif. Intell. Eng.*, vol. 15, no. 2, pp. 207–218, 2001.
[31] C. Science and M. Shores, “natural language specifications Sentcnce--,” *English*, no. 98, 1999.
[32] I. S. Bajwa and M. A. Choudhary, “to UML Class Models,” pp. 224–237, 2012.
[33] D. K. Deeptimahanti and M. A. Babar, “An automated tool for generating UML models from natural language requirements,” *ASE2009 - 24th IEEE/ACM Int. Conf. Autom. Softw. Eng.*, pp. 680–682, 2009.
[34] D. K. Deeptimahanti and R. Sanyal, “Semi-automatic generation of UML models from natural language requirements,” *Proc. 4th India Softw. Eng. Conf. 2011, ISEC’11*, pp. 165–174, 2011.
[35] P. More and R. Phalnikar, “Generating UML Diagrams from Natural Language Specifications,” *Int. J. Appl. Inf. Syst.*, vol. 1, no. 8, pp. 19–23, 2012.
[36] J. Becker, P. Bergener, D. Breuker, and M. Räckers, “An empirical assessment of the usefulness of weakness patterns in business process redesign,” *ECIS 2012 - Proc. 20th Eur. Conf. Inf. Syst.*, 2012.
[37] E. Knauss and D. Lübke, “Using the friction between business processes and use cases in SOA requirements,” *Proc. - Int. Comput. Softw. Appl. Conf.*, pp. 601–606, 2008.
[38] M. Ehrig, A. Koschmider, and A. Oberweis, “Measuring similarity between semantic business process models,” *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 67, pp. 71–80, 2007.
[39] J. Becker, D. Breuker, P. Delfmann, H. A. Dietrich, and M. Steinhorst, “Identifying business process activity mappings by optimizing behavioral similarity,” *18th Am. Conf. Inf. Syst. 2012, AMCIS 2012*, vol. 1, pp. 41–49, 2012.
[40] R. Knackstedt, D. Kuropka, O. Müller, and A. Polyvyanyy, “An ontology-based service discovery approach for the provisioning of product-service bundles,” *16th Eur. Conf. Inf. Syst. ECIS 2008*, 2008.
[41] V. Gacitua-Decar and C. Pahl, “Automatic business process pattern matching for enterprise services design,” *Serv. 2009 - 5th 2009 World Congr. Serv.*, no. PART 2, pp. 111–118, 2009.
[42] F. Meziane, N. Athanasakis, and S. Ananiadou, “Generating Natural Language specifications from UML class diagrams,” *Requir. Eng.*, vol. 13, no. 1, pp. 1–18, 2008.
[43] H. Dalianis, “A method for validating a conceptual model by natural language discourse generation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 593 LNCS, pp. 425–444, 1992.
[44] B. Lavoie, O. Rambow, and E. Reiter, “The ModelExplainer,” *Demonstr. Notes Int. Nat. Lang. Gener. Work.*, pp. 9–12, 1996.
[45] F. Friedrich, J. Mendling, and F. Puhlmann, “Process Model Generation from Natural

Language Text BT - Advanced Information Systems Engineering," 2011, pp. 482–496.
[46] S. Fan, Z. Hua, V. C. Storey, and J. L. Zhao, "A process ontology based approach to easing semantic ambiguity in business process modeling," *Data Knowl. Eng.*, vol. 102, pp. 57–77, 2016.
[47] A. Ravid and D. M. Berry, "A method for extracting and stating software requirements that a user interface prototype contains," *Requir. Eng.*, vol. 5, no. 4, pp. 225–241, 2000.
[48] K. S. Mukasa and H. Kaindl, "An integration of requirements and user interface specifications," *Proc. 16th IEEE Int. Requir. Eng. Conf. RE'08*, pp. 327–328, 2008.
[49] A. Rashid, D. Meder, J. Wiesenberger, and A. Behm, "Visual requirement specification in end-user participation," *First Int. Work. Multimed. Requir. Enineering, MeRE'06*, 2006.
[50] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano, "On the effectiveness of screen mockups in requirements engineering: Results from an internal replication," *ESEM 2010 - Proc. 2010 ACM-IEEE Int. Symp. Empir. Softw. Eng. Meas.*, no. July 2014, 2010.
[51] K. Schneider, "Generating fast feedback in requirements elicitation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4542 LNCS, pp. 160–174, 2007.
[52] Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, and V. López-Jaquero, "USIXML: A language supporting multi-path development of User Interfaces," *Lect. Notes Comput. Sci.*, vol. 3425, pp. 200–220, 2005.
[53] K. Schwaber and J. Sutherland, "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework," *SCRUM Inc.*, p. 224, 2010.
[54] S. López Ornat, "La formación de la oración simple: las omisiones sintáctica (S-V-0) en la adquisición del español," *Estud. Psicol.*, vol. 41, pp. 41–72, 1990.
[55] P. Qi, T. Dozat, Y. Zhang, and C. D. Manning, "Universal dependency parsing from scratch," *CoNLL 2018 - SIGNLL Conf. Comput. Nat. Lang. Learn. Proc. CoNLL 2018 Shar. Task Multiling. Parsing from Raw Text to Univers. Depend.*, pp. 160–170, 2018.
[56] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," pp. 55–60, 2015.
[57] Y. Feng, Q. Deng, and D. Yu, "BLCUNLP: Corpus Pattern Analysis for Verbs Based on Dependency Chain," no. SemEval, pp. 325–328, 2015.
[58] D. Chen and C. Manning, "A Fast and Accurate Dependency Parser using Neural Networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing ({EMNLP})*, 2014, pp. 740–750.
[59] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," pp. 63–70, 2000.
[60] R. Huang, I. Cases, D. Jurafsky, C. Condoravdi, and E. Riloff, "Distinguishing past, on-going, and future events: The eventstatus corpus," *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, no. 1, pp. 44–54, 2016.
[61] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*, vol. 9783642290. Springer Publishing Company, Incorporated, 2012.
[62] C. D. Manning, "Intro to Information Retrieval," *Inf. Retr. Boston.*, no. c, pp. 1–18, 2009.