

**IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA RECOMENDADOR
GAMIFICADO PARA LA ELECCIÓN DE LIBROS MEDIANTE ARQUITECTURA
POR COMPONENTES**

**XENIA VIVIANA CADENA DÍAZ
NELSON ARLEY CARANTÓN GALEANO**



**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN INGENIERÍA DE SOFTWARE
BOGOTÁ D.C.
2017**

**IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA RECOMENDADOR
GAMIFICADO PARA LA ELECCIÓN DE LIBROS MEDIANTE ARQUITECTURA
POR COMPONENTES**

XENIA VIVIANA CADENA DÍAZ

20171099004

NELSON ARLEY CARANTÓN GALEANO

20171099005

ANTEPROYECTO

INGENIERO FERNANDO MARTÍNEZ RODRÍGUEZ

Director



UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

ESPECIALIZACIÓN EN INGENIERÍA DE SOFTWARE

BOGOTÁ D.C.

2017

CONTENIDO

INTRODUCCIÓN.....	5
1. TÍTULO Y DEFINICIÓN DEL TEMA DE INVESTIGACIÓN.....	7
2. ESTUDIO DEL PROBLEMA DE INVESTIGACIÓN	8
2.1 PLANTEAMIENTO DEL PROBLEMA.....	8
2.2 FORMULACIÓN DEL PROBLEMA.....	9
2.3 SISTEMATIZACIÓN DEL PROBLEMA.....	9
3. OBJETIVOS DE LA INVESTIGACIÓN	10
3.1 OBJETIVO GENERAL.....	10
3.2 OBJETIVOS ESPECÍFICOS	10
4. JUSTIFICACIÓN DE LA INVESTIGACIÓN	11
4.1 JUSTIFICACIÓN PRÁCTICA	11
5. HIPÓTESIS DE TRABAJO	12
6. MARCO DE REFERENCIA.....	13
6.1 MARCO TEÓRICO.....	13
6.1.1 Sistema Recomendador.....	13
6.1.2 Gamificación	14
6.1.3 Arquitectura por Componentes [14].....	15
6.1.4 Spring [15]	18
6.1.3 Metodología Scrum.....	20
6.2 ESTADO DEL ARTE.....	23
7. ASPECTOS METODOLÓGICOS.....	26
7.1 TIPO DE ESTUDIO	26
7.2 MÉTODO DE INVESTIGACIÓN	26
7.3 FUENTES Y TÉCNICAS PARA LA RECOLECCIÓN DE LA INFORMACIÓN	27
7.4 TRATAMIENTO DE LA INFORMACIÓN	28
8. ALCANCES, LIMITACIONES Y RESULTADOS ESPERADOS	29
8.1 ALCANCES	29
8.2 LIMITACIONES	29
8.3 RESULTADOS ESPERADOS.....	30

9. TABLA DE CONTENIDO DEL PROYECTO	31
9.1 CAPÍTULOS, OBJETIVOS Y SUBTEMAS DE LA INVESTIGACIÓN.....	31
10. CRONOGRAMA DE TRABAJO	33
10.1 TABLA DE ACTIVIDADES	33
10.2 DIAGRAMA DE GANTT	34
11. PRESUPUESTO	35
11.1 COSTOS POR SERVICIOS PERSONALES.....	35
11.2 GASTOS GENERALES	35
11.3 COSTO TOTAL DEL PROYECTO	36
BIBLIOGRAFÍA.....	37

INTRODUCCIÓN

Con el pasar de los años, el mundo Web ha presentado un crecimiento exponencial hablando de sitios y documentos Web y el uso del internet. Es por esto que el manejo de la gran cantidad de información que existe en este mundo, es cada vez más complicado, sobre todo a la hora de buscar información. Es aquí donde han ido consolidándose herramientas que ayudan a disminuir la sobrecarga de información; entre estas herramientas existen los SR (Sistemas Recomendadores), los cuales filtran los ítems de la información a partir de técnicas que identifican los principales atributos que más se acomodan a las preferencias de los usuarios, recomendaciones que se generan a partir de gustos de otros usuarios o del contenido de la información. [1]

La estructura de estos sistemas, plantea que deben tener unas entradas y unas salidas, debe manejar un método (basado en contenido y colaborativos) [2,3] que genere las recomendaciones (salidas) y un grado de personalización (no personalizados, personalización efímera y personalización persistente). Cabe destacar que los métodos no son excluyentes y se pueden usar en el mismo sistema para generar recomendaciones. [4]

Para plantear el sistema también hay que conocer la gamificación que en este contexto puede ser planteada como un proceso de interacción de las dinámicas y mecánicas de juego en una página web, un servicio, una comunidad online, portal de contenido o compañía de marketing con el fin de lograr participación e implicación del usuario. [5] Con la gamificación se aplican elementos y reglas de juegos con el fin de captar y atrapar a los usuarios, trabajadores y clientes. La finalidad es generar compromiso, fomentar conductas deseadas y resolver problemas. Éste busca sobre todo resolver los problemas que plantean las tareas más aburridas de la vida cotidiana. [6]

Existen una gran variedad de SR en el mundo Web, Netflix por ejemplo le brinda al usuario recomendaciones de series o películas a partir del contenido al que ya haya accedido anteriormente el mismo usuario, método basado en contenido. Tinder por ejemplo recomienda a sus usuarios usando el método colaborativo, recomendando posibles parejas basado en el perfil del usuario. Pero no hay que quedarse ahí, sistemas como el buscador de Google implementan un SR híbrido para arrojar el mejor resultado en la búsqueda, cuando busca el contenido de las páginas web basándose en la búsqueda y para clasificar esos resultados según la geolocalización de la persona que realiza la consulta. [7] También existen SR como Waze que se apoyan de la gamificación para incentivar la fidelización en el usuario, este sistema genera una calificación por kilómetro recorrido la cual le da un personaje en la aplicación, lo que incentiva al usuario a usarla mientras conducen, haciendo que el mismo usuario alimente las bases de datos del sistema. Esto da una idea a lo que es un SR gamificado.

El objetivo principal en este proyecto es la implementación de un prototipo de Sistema Recomendador Gamificado para generarle al usuario la mejor recomendación de libros para su próxima lectura. Se quiere que el sistema implemente un híbrido con los métodos de recomendación ya mencionados y además gamificar la aplicación a partir de un sistema de puntos, lo que logrará la fidelización del usuario y que sea éste el que ayude a alimentar al sistema.

1. TÍTULO Y DEFINICIÓN DEL TEMA DE INVESTIGACIÓN

El título planteado para el proyecto incluye las dos metodologías que se van a usar, el objetivo de la aplicación, la arquitectura que se va usar en el desarrollo de la aplicación y por último se aclara que lo que se va a implementar es un prototipo ya que se va a probar con algo nuevo; el tema de investigación como lo dice el título es sistema de recomendación junto con gamificación. El título es el siguiente:

IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA RECOMENDADOR
GAMIFICADO PARA LA ELECCIÓN DE LIBROS MEDIANTE ARQUITECTURA
POR COMPONENTES.

2. ESTUDIO DEL PROBLEMA DE INVESTIGACIÓN

2.1 PLANTEAMIENTO DEL PROBLEMA

Un “sistema recomendador” analiza y procesa información histórica de los usuarios (edad, compras previas, calificaciones...), de los productos o de los contenidos (marcas, modelos, precios, contenidos similares...) y la transforma en conocimiento accionable, es decir, predice qué producto puede ser interesante para el usuario y para la empresa. Los recomendadores, además, tienen cierto nivel de autonomía a la hora de presentar las recomendaciones al usuario final. Estos sistemas se han hecho muy populares en los últimos años. Entre los ámbitos más extendidos se encuentran las recomendaciones de productos en tiendas online, películas, vídeos, música, libros, productos o recomendaciones de perfiles a los que seguir en redes sociales.

Existen sistemas recomendadores de libros que le indican al usuario los libros que podría leer, basado en calificaciones de otras personas; esto genera un problema, las calificaciones no siempre representan lo que el usuario quiere y además podrían presentarse ambigüedades a la hora de generar una recomendación. No existen sistemas populares que recomienden libros basados en más características a parte de las calificaciones. Hasta el momento no se ha encontrado una aplicación que tenga en cuenta el historial de lecturas del usuario y que además combine estas características con el historial de perfiles similares a los del usuario; pero esto no es todo, usar aplicaciones que no son llamativas y que no estén en un ambiente gamificado que incentiven a la persona a seguir usando la plataforma, se convierte en un problema ya que esto lleva a que el sistema se vuelva obsoleto ante las demás aplicaciones que incentiven estas actividades.

Es decir, un sistema recomendador de libros que no recomiende con base a un historial de lectura, perfiles similares, calificaciones y que además combine la estrategias lúdicas para que sea el propio usuario el que alimente el sistema; puede llevar al usuario a aburrirse de la aplicación por las recomendaciones poco acertadas y porque no le interesen más cosas en el sistema.

2.2 FORMULACIÓN DEL PROBLEMA

¿Cómo recomendar a un usuario la mejor opción de libro a leer y que a su vez sea incentivado mediante estrategias lúdicas a apoyar la alimentación del sistema y que lo siga usando?

2.3 SISTEMATIZACIÓN DEL PROBLEMA

- ¿Cuál arquitectura es óptima para un sistema recomendador que permita la escalabilidad y el mejor uso del sistema?
- ¿Cuál es el mejor algoritmo de recomendación para un sistema que recomiende libros?
- ¿Qué metodología es útil para que apoye al sistema recomendador a que los usuarios sigan usando la aplicación?

3. OBJETIVOS DE LA INVESTIGACIÓN

3.1 OBJETIVO GENERAL

Implementar un prototipo de software recomendador mediante el uso de arquitectura moderna y gamificación, para incentivar la lectura en las personas de todas las edades.

3.2 OBJETIVOS ESPECÍFICOS

- Diseñar una arquitectura por componentes mediante la ingeniería de software para la solidez del sistema recomendador.
- Implementar una métrica para el sistema mediante recomendación basada en contenido y colaborativa, que se retroalimente, para que genere la recomendación que mejor se adapte al usuario.
- Desarrollar un sistema de puntos mediante el uso de la gamificación para motivar al usuario a usar la aplicación y a ayudar con la alimentación del sistema.

4. JUSTIFICACIÓN DE LA INVESTIGACIÓN

4.1 JUSTIFICACIÓN PRÁCTICA

Algunas personas toman la iniciativa de empezar a leer un libro ya sea por investigar o por entretenimiento, pero se les presenta el problema más común para el mundo de los lectores ¿Qué libro leer? Existen herramientas que basadas en calificaciones de otros usuarios le recomiendan a la persona el libro que podría leer, pero estas herramientas no siempre son las más efectivas. Las calificaciones que dan otros usuarios podrían no ser iguales al pensamiento del usuario que consulta. Además de esto, las bases de datos suelen no aumentar su información y los resultados empiezan a repetirse. Por otra parte las herramientas que realizan estas recomendaciones no son muy empleadas porque no incentivan al usuario a interactuar con ellas.

Es por esto que es importante unir la recomendación basada en contenido junto con la colaborativa y plantear una métrica que se retroalimente para que genere una lista de libros recomendados basados en el historial de lecturas del usuario y el historial de otras personas con su mismo perfil, también se presenta la necesidad de combinar la gamificación para que el sistema incentive el empleo de la aplicación y además la plataforma sea agradable para el usuario, gracias a estrategias lúdicas que lo enganchen y lo vuelvan parte esencial para alimentar el sistema.

5. HIPÓTESIS DE TRABAJO

Los Sistemas Recomendadores, surgen para facilitar la toma de decisiones en temas en los que las opciones son numerosas y variadas. Estos sistemas son utilizados para indicar información, productos, y otros ítems a los usuarios que los visitan, tratan de automatizar aspectos de un modelo de búsqueda de información completamente distinto al tradicional [8]. Con base a esto y lo planteado para el sistema se afirma qué:

- Un sistema recomendador basado en el historial de lectura del usuario y su perfil, es más eficiente que aquellos que se basan en calificaciones de otros usuarios.
- La gamificación es una estrategia útil para incentivar a los usuarios a seguir usando la aplicación y alimentar el sistema.
- Las métricas que se retroalimentan por las decisiones que toman los usuarios, son más acertadas para arrojar recomendaciones a las personas.
- La arquitectura por componentes es útil para proyectos que se van a migrar a otro ambiente, ya que realiza la arquitectura de forma modular, lo que le permite migrar de forma eficiente a otras plataformas.

6. MARCO DE REFERENCIA

6.1 MARCO TEÓRICO

6.1.1 Sistema Recomendador

Un SR está asociado con un conjunto de ítems $I = \{ i_1, \dots, i_n \}$ y su objetivo es recomendar a los usuarios ítems de I que les puedan ser de interés. Por ejemplo, se podría diseñar un SR para la recomendación de películas; de hecho a lo largo de la literatura hemos encontrado numerosos ejemplos de SR de películas, tales como Film-Conseil, MovieFinder, Reel o MetaLens entre otros. En el caso de un SR para la recuperación de documentos los ítems serían los documentos almacenados. [12]

La implementación de técnicas para el desarrollo de los SR está íntimamente relacionada con el tipo de información que se vaya a utilizar. Una primera fuente de información a tener en cuenta es el tipo de ítems con los que vamos a trabajar. Habrá situaciones en las que únicamente conozcamos un identificador de cada ítem. Por ejemplo en el caso de la recomendación de películas solemos conocer únicamente el título. En otras situaciones, dispondremos de más información sobre los ítems, a través de una serie de atributos. En el caso de la recomendación de películas, podrían ser el año en que se hizo la película, el género, el director, protagonistas, etc. En el caso de recuperación de documentos, la información con la que contamos serían los términos índice usados en su representación. En general, cuanto más sofisticada es la representación de los ítems mejor se puede desarrollar la actividad de los SR. [12]

En cualquier caso, algunos aspectos que debemos considerar sobre las recomendaciones en el diseño de SR son [12]:

- Representación de las recomendaciones: Los contenidos de una evaluación pueden venir dados por un único bit (recomendado o no) o por comentarios de texto sin estructurar.
- Expresión de las recomendaciones: Las recomendaciones pueden ser introducidas de forma explícita o bien de forma implícita.
- Aspectos de identificación de la fuente: Las recomendaciones pueden realizarse de forma anónima, identificando la fuente, o bien usando un pseudónimo.
- Forma de agregar las evaluaciones: Se refiere a cómo vamos a ir agregando las evaluaciones disponibles sobre los ítems de cara a generar las recomendaciones. Más adelante entraremos en más detalle sobre este aspecto que es de suma importancia de cara a clasificar los SR.
- Uso de las recomendaciones: Las recomendaciones se pueden usar de distintas formas. Por ejemplo, se podrían mostrar los ítems en forma de lista ordenada según las recomendaciones de cada uno, o a la hora de visualizar los ítems que se muestre también su recomendación.

6.1.2 Gamificación

Cuando se habla de gamificación, se habla de “aplicar estrategias (pensamientos y mecánicas) de juegos en contextos no jugables, ajenos a los juegos, con el fin de que las personas adopten ciertos comportamientos”. Con la gamificación se aplican elementos y reglas de juegos con el fin de captar y atrapar a los usuarios, trabajadores y clientes. Su finalidad es la de generar compromiso, fomentar

conductas deseadas y resolver problemas. La gamificación busca sobre todo resolver los problemas que plantean las tareas más aburridas de la vida cotidiana. [13]

6.1.3 Arquitectura por Componentes [14]

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado. El estilo de arquitectura basado en componentes tiene las siguientes características:

- Es un estilo de diseño para aplicaciones compuestas de componentes individuales.
- Pone énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen interfaces bien definidas.
- Define una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades.

Principios Fundamentales

Un componente es un objeto de software específicamente diseñado para cumplir con cierto propósito. Los principios fundamentales cuando se diseña un componente es que estos deben ser:

- Reusable. Los componentes son usualmente diseñados para ser utilizados en escenarios diferentes por diferentes aplicaciones, sin embargo, algunos componentes pueden ser diseñados para tareas específicas.
- Sin contexto específico. Los componentes son diseñados para operar en diferentes ambientes y contextos. Información específica como el estado de los datos deben ser pasadas al componente en vez de incluirlos o permitir al componente acceder a ellos.
- Extensible. Un componente puede ser extendido desde un componente existente para crear un nuevo comportamiento.
- Encapsulado. Los componentes exponen interfaces que permiten al programa usar su funcionalidad. Sin revelar detalles internos, detalles del proceso o estado.
- Independiente. Los Componentes están diseñados para tener una dependencia mínima de otros componentes. Por lo tanto los componentes pueden ser instalados en el ambiente adecuado sin afectar otros componentes o sistemas.

Beneficios

Los siguientes son los principales beneficios del estilo de arquitectura basado en componentes:

- Facilidad de Instalación. Cuando una nueva versión esté disponible, usted podrá reemplazar la versión existente sin impacto en otros componentes o el sistema como un todo.
- Costos reducidos. El uso de componentes de terceros permite distribuir el costo del desarrollo y del mantenimiento.

- **Facilidad de desarrollo.** Los componentes implementan un interface bien definida para proveer la funcionalidad definida permitiendo el desarrollo sin impactar otras partes del sistema.
- **Reusable.** El uso de componentes reutilizables significa que ellos pueden ser usados para distribuir el desarrollo y el mantenimiento entre múltiples aplicaciones y sistemas.
- **Mitigación de complejidad técnica.** Los componentes mitigan la complejidad por medio del uso de contenedores de componentes y sus servicios. Ejemplos de servicios de componentes incluyen activación de componentes, gestión de la vida de los componentes, gestión de colas de mensajes para métodos del componente y transacciones.

Ejemplos

Tipos comunes de componentes usados en aplicaciones incluyen:

- Componentes de interfaz de usuario, como grillas, botones, etc., generalmente conocidos como “controles”.
- Componentes de ayuda que exponen un conjunto específico de funciones usados por otros componentes.
- Componentes que se no se usan con mucha frecuencia o son intensivos en recursos y deben ser actividades usando una aproximación de solo en el momento justo (Just in Time (JIT)). Estos son comunes en escenarios de componentes distribuidos o en componentes remotos.

- Componentes encolados, aquellos cuyos métodos pueden ser ejecutados de forma asíncrona usando colas de mensajes del tipo almacenamiento, entrega.

6.1.4 Spring [15]

Spring es un *framework* alternativo al *stack* de tecnologías estándar en aplicaciones JavaEE. Nació en una época en la que las tecnologías estándar JavaEE y la visión "oficial" de lo que debía ser una aplicación Java Enterprise tenían todavía muchas aristas por pulir. Los servidores de aplicaciones eran monstruosos devoradores de recursos y los EJB eran pesados, inflexibles y era demasiado complejo trabajar con ellos. En ese contexto, Spring popularizó ideas como la inyección de dependencias o el uso de objetos convencionales (POJOs) como objetos de negocio, que suponían un soplo de aire fresco. Estas ideas permitían un desarrollo más sencillo y rápido y unas aplicaciones más ligeras. Eso permitió que de ser un *framework* inicialmente diseñado para la capa de negocio pasara a ser un completo *stack* de tecnologías para todas las capas de la aplicación.

Las ideas "innovadoras" que en su día popularizó Spring se han incorporado en la actualidad a las tecnologías y herramientas estándar. Así, ahora mismo no hay una gran diferencia entre el desarrollo con Spring y el desarrollo JavaEE "estándar", o al menos no tanta como hubo en su día. No obstante, Spring ha logrado aglutinar una importante comunidad de desarrolladores en torno a sus tecnologías y hoy por hoy sigue constituyendo una importante alternativa al estándar que merece la pena conocer. En la actualidad, las aportaciones más novedosas de Spring se centran en los campos de Big Data/NoSQL, HTML5/móviles y aplicaciones sociales.

Básicamente, la mayor diferencia práctica que podemos encontrar hoy en día entre desarrollar con Spring y con JavaEE estándar es la posibilidad de usar un servidor web convencional al estilo Tomcat para desplegar la aplicación. Las tecnologías JavaEE más sofisticadas requieren del uso de un servidor de aplicaciones, ya que los APIs los implementa el propio servidor, mientras que Spring no es más que un conjunto de librerías portables entre servidores. En otras palabras, usando JavaEE estándar, nos atamos al servidor de aplicaciones y usando Spring nos atamos a sus APIs. Eso sí, los desarrolladores de Spring se han preocupado bastante de armonizar con el estándar en la medida de lo posible, por ejemplo dando la posibilidad de usar anotaciones estándar aun con implementaciones propias por debajo. La idea es obstaculizar lo menos posible una posible portabilidad a JavaEE, idea que es de agradecer en un mundo en que todos los fabricantes intentan de una forma u otra mantener un público cautivo.

Hay una abundante bibliografía sobre Spring, aunque la documentación del propio proyecto es excelente y bastante exhaustiva, pudiéndose utilizar perfectamente no solo como manual de referencia sino como tutorial detallado. La hemos tomado como referencia básica para la elaboración de estos apuntes.

Desde un punto de vista genérico, Spring se puede ver como un soporte que nos proporciona tres elementos básicos:

Servicios enterprise: podemos hacer de manera sencilla que un objeto sea transaccional, o que su acceso esté restringido a ciertos roles, o que sea accesible de manera remota y transparente para el desarrollador, o acceder a otros muchos servicios más, sin tener que escribir el código de manera manual. En la mayoría de los casos solo es necesario anotar el objeto.

Estereotipos configurables para los objetos de nuestra aplicación: podemos anotar nuestras clases indicando por ejemplo que pertenecen a la capa de negocio o de acceso a datos. Se dice que son configurables porque podemos

definir nuestros propios estereotipos "a medida": por ejemplo podríamos definir un nuevo estereotipo que indicara un objeto de negocio que además sería cacheable automáticamente y con acceso restringido a usuarios con determinado rol.

Inyección de dependencias:: ya hemos visto este concepto cuando se hablaba de CDI de JavaEE. La inyección de dependencias nos permite solucionar de forma sencilla y elegante cómo proporcionar a un objeto cliente acceso a un objeto que da un servicio que este necesita. Por ejemplo, que un objeto de la capa de presentación se pueda comunicar con uno de negocio. En Spring las dependencias se pueden definir con anotaciones o con XML.

6.1.3 Metodología Scrum

Scrum es una metodología de desarrollo ágil que se caracteriza por [16]:

- Tener estrategias de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Anteponer la calidad del resultado y el conocimiento tácito en equipos auto organizados, que en la calidad de los procesos empleados.
- Trabajar las diferentes fases del desarrollo paralelamente, en lugar de realizarlas secuencialmente o en cascada.

Roles Principales

- Product Owner: representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

- **Scrum Master:** El Scrum es facilitado por un Scrum Master, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El Scrum Master no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El Scrum Master se asegura de que el proceso Scrum se utiliza como es debido. El Scrum Master es el que hace que las reglas se cumplan.
- **Equipo de desarrollo:** El equipo tiene la responsabilidad de entregar el producto. Es recomendable un pequeño equipo de 2 a 8 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc.).

Sprint

Es el período en el cual se lleva a cabo cada ciclo en sí. Es recomendado que la duración de los Sprint sea constante y definida por el equipo con base en su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo.

Reuniones en Scrum

- Daily Scrum

Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama daily standup o Stand-up meeting. El Scrum tiene unas guías específicas:

- La reunión comienza puntualmente a su hora.
- Todos son bienvenidos, pero sólo los involucrados en el proyecto pueden hablar.

- La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- La reunión debe ser a la misma hora todos los días.
- Durante la reunión, cada miembro del equipo contesta a tres preguntas:
 - ¿Qué hice ayer?
 - ¿Qué voy hacer mañana?
 - ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

- **Reunión de Planificación del Sprint**

Al inicio de cada ciclo de Sprint (cada 15 o 30 días), se lleva a cabo una reunión de planificación del Sprint. Se pretende:

- Seleccionar qué trabajo se hará.
- Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que llevará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.
- Realizarse esta planificación en ocho horas como tiempo límite.

Al final del ciclo Sprint se hacen la reunión de revisión del Sprint.

- **Reunión de Revisión del Sprint (Sprint Review Meeting)**

- Revisar el trabajo que fue completado y no completado
- Presentar el trabajo completado a los interesados (también conocido como demo)
- El trabajo incompleto no puede ser demostrado
- Cuatro horas como límite

Etapas Principales [17]

1.- Pre-juego

Planificación: Definición de una nueva versión basada en la pila actual, junto con una estimación de coste y agenda. Si se trata de un nuevo sistema, esta fase abarca tanto la visión como el análisis. Si se trata de la mejora de un sistema existente comprende un análisis de alcance más limitado. Arquitectura: Diseño de la implementación de las funcionalidades de la pila. Esta fase incluye la modificación de la arquitectura y diseño generales.

2.- Juego

Desarrollo de sprints: Desarrollo de la funcionalidad de la nueva versión con respeto continuo a las variables de tiempo, requisitos, costo y competencia. La interacción con estas variables define el final de esta fase. El sistema va evolucionando a través de múltiples iteraciones de desarrollo o sprints.

3.- Post-juego

Preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la versión.

6.2 ESTADO DEL ARTE

No se encontró ningún aplicativo que mescle un sistema recomendador para la lectura de libros con un sistema de gamificación.

- **thefussylibrarian.com** : Entre los puntos a destacar de este aplicativo, envía recomendaciones diarias al correo electrónico, entre los puntos que no tiene son que se basa en la recomendación enfocada por contenido, además la base de datos de libros es estática y es manejada por los administradores del aplicativo y por última desventaja se tiene que el sitio solo tiene soporte para el idioma inglés y libros en este idioma.

- **quelibroleo.com** : Este aplicativo tiene un enfoque más de buscador de libros que de recomendador, pero la parte de recomendador es basada en recomendación colaborativa ya que los usuarios colocan libros que quieren que otras personas lean por esto no es muy eficaz el sistema ya que no garantiza que recomiende libros enfocados en cada usuario, el sitio está en español, también tiene el problema es que los libros que alimentan el sistema son administrados por el creador del sitio.

- **ebrolis.com** : El aplicativo como novedad tiene que aparte de recomendador hace de puente con las librerías para comprar el libro o con Amazon.com para libros virtuales, en cuanto al sistema recomendador es basado en contenido ya que según los libros que vaya poniendo en el carrito o haya comprado, el aplicativo está en varios idiomas, igual que los otros aplicativos tienen la base de datos restringida por la alimentación del administrador.

- **boolino.es**: Es un aplicativo muy similar a *quelibroleo.com* ya que se enfoca más en un buscador de libros la recomendación colaborativa y también vende libros, la base de datos no se alimenta por los usuarios, tiene varios idiomas para seleccionar y libros de todo tipo. Pero no deja de ser una tienda de libros con un buscador.

- **whatshouldireadnext.com** : Este es el único aplicativo de los evaluados que cuenta con un sistema que se alimenta de los mismos usuarios, la recomendación es colaborativa, pero es solo está en inglés, también cuenta con una base de datos de libros pequeña ya que su comunidad es pequeña, no hay un sistema que atraiga nuevos usuarios o que los antiguos sigan colaborando, además que no es obligatorio registrar libros.

Como se pudo deducir ningún a casi ningún aplicativo que está entre los más usados utiliza la recomendación híbrida, además casi todos se enfocan en la recomendación colaborativa eso quiere decir no se enfocan en lo que consume el usuario, tampoco se encontró un aplicativo que utilice gamificación.

7. ASPECTOS METODOLÓGICOS

7.1 TIPO DE ESTUDIO

El tipo de estudio señala el nivel de profundidad con el cual el investigador busca abordar el objeto de conocimiento. Existen 3 tipos de estudio, el exploratorio, el descriptivo y el explicativo. [9]

En este proyecto se definió que el tipo de estudio es el exploratorio, ya que se requiere familiarizarse con los sistemas recomendadores gamificados y es el punto de partida para el método de recomendación que propondrá el equipo.

7.2 MÉTODO DE INVESTIGACIÓN

El método de investigación es el procedimiento riguroso, formulado de una manera lógica, que el investigador debe seguir en la adquisición del conocimiento. Existen varios métodos de investigación entre los que se destacan: método de observación, método inductivo, método deductivo, método de análisis, método de síntesis, entre otros [9]

El método de investigación del proyecto es el método de observación, porque se está partiendo desde la experiencia personal de los autores, y se ha hablado con personas vinculadas al objeto principal del proyecto y además se propone un método de recomendación a partir de la experiencia.

7.3 FUENTES Y TÉCNICAS PARA LA RECOLECCIÓN DE LA INFORMACIÓN

Las fuentes son hechos o documentos a los que acude el investigador y que le permiten obtener información, ya sean fuentes primarias que son las que acuden a técnicas de recolección de información y fuentes secundarias cuando la información se recopila de fuentes escritas como textos, revistas, documentos, prensa entre otros. Las técnicas son los medios empleados para recolectar la información entre los que están la observación, encuestas, cuestionarios, entrevistas y sondeos. [9]

Para este proyecto se considera que es un trabajo teórico-práctico por lo que es necesaria la información primaria recogida directamente por los autores, para lo cual se definió como técnica de recolección la encuesta.

Una encuesta es una investigación realizada sobre una muestra de sujetos representativa de un colectivo más amplio, utilizando procedimientos estandarizados de interrogación con intención de obtener mediciones cuantitativas de una gran variedad de características objetivas y subjetivas de la población. [10]

Se definió la encuesta ya que se necesita establecer las preferencias a la hora de elección de libros para los valores iniciales de los que va a empezar a usar la métrica planteada en la aplicación para el método de recomendación. La idea es realizar esta encuesta inicial y cuando se tenga el prototipo hacer que las personas lo prueben y luego realizar una encuesta de satisfacción para realizar mejoras futuras a la aplicación.

7.4 TRATAMIENTO DE LA INFORMACIÓN

El tratamiento de la información es la determinación de los procedimientos para la codificación y tabulación de la información para el recuento, clasificación y ordenación de la información en tablas o cuadros. [9]

La técnica que se va a implementar es la técnica de análisis cuantitativo ya que se van describir las variables para caracterizarlas una por una, se usarán gráficos de barras para presentar la información y poder determinar los valores de cada variable de la métrica planteada. También es importante determinar el tamaño de la muestra que se va a seleccionar, ya que es un paso importante en cualquier estudio de investigación, se debe justificar convenientemente de acuerdo al planteamiento del problema, la población, los objetivos y el propósito de la investigación. [11]

8. ALCANCES, LIMITACIONES Y RESULTADOS ESPERADOS

8.1 ALCANCES

Los alcances que se proponen son los siguientes:

- Propuesta de métrica para los métodos de recomendación.
- Aplicación de la métrica en la recomendación de los libros.
- Uso de recomendación basada en contenido y recomendación colaborativa.
- Diseño de sistema de puntos.
- La aplicación se desarrollará bajo una arquitectura por componentes, mediante metodología de desarrollo Scrum y framework Spring.
- La aplicación será web.

8.2 LIMITACIONES

- La aplicación será un prototipo.
- La base de datos inicial será la obtenida por los autores.
- Los usuarios iniciales serán los obtenidos en el cálculo de la muestra para poder realizar el análisis final de los resultados.
- Los autores interactuarán con un framework nuevo (Spring) ya que esto conlleva al aprendizaje y más tiempo a la hora del desarrollo.
- No se podrá interactuar con todas las redes sociales para compartir los puntos de los usuarios. Se iniciará con Facebook y dependiendo de los tiempos se podrá experimentar con más.

8.3 RESULTADOS ESPERADOS

Se espera que con la unión de estas dos metodologías el usuario pueda con la aplicación realizar las siguientes actividades:

- Registrarse en el sistema y guardar un perfil propio.
- Ingresar con usuario y contraseña a la plataforma.
- Solicitar recomendación de libros basado por lo menos en los 3 libros que ingresó al momento de registrarse.
- Consultar la lista de libros recomendados y de cada uno seleccionar si ya lo leyó o si lo va a leer.
- Consultar su perfil en donde encontrará lo siguiente:
 - a) Modificar datos del perfil
 - b) Ingresar libro nuevo
 - c) Consultar libros de la base de datos sin generar recomendación y agregar comentarios.
 - d) Compartir estado de puntos en redes sociales.
 - e) Consultar lista de libros leídos y realizar modificaciones.
 - f) Consultar lista de libros por leer y realizar modificaciones.

9. TABLA DE CONTENIDO DEL PROYECTO

9.1 CAPÍTULOS, OBJETIVOS Y SUBTEMAS DE LA INVESTIGACIÓN

La tabla de contenido que se propones inicialmente es:

Resumen

Introducción

1. Planteamiento del problema

1.2 Descripción del problema

1.3 Formulación del problema

1.4 Sistematización del problema

2. Justificación

3. Objetivos

3.1 Objetivo General

3.2 Objetivos Específicos

4. Marco Referencial

4.1 Marco Teórico

4.2 Estado del Arte

5. Planificación

5.1 Diseño de BackLog

5.2 Evaluación y control adecuado de los riesgos

6. Diseño y Desarrollo

6.1 Diseño de Arquitectura

6.2 Diseño Métrica Propuesta para el Sistema Recomendador

6.3 Diseño y Desarrollo de la persistencia de datos

6.4 Diseño y Desarrollo de Sistema recomendador

6.5 Diseño y Desarrollo del Sistema de Puntos

6.6 Diseño y Desarrollo de Módulos adicionales del sistema

7. Cierre

7.1 Integración

7.2 Pruebas del Sistema

7.3 Resultados

8. Conclusiones

9. Trabajos Futuros

10. Bibliografía

ANEXOS

Anexo 1. Documentación técnica

10. CRONOGRAMA DE TRABAJO

10.1 TABLA DE ACTIVIDADES

# Tarea	Nombre Tarea	Fecha Inicio	Fecha Final	Num Dias
1	Planificación	05/29/17	06/04/17	7
1.1	Diseño BackLog	05/29/17	06/01/17	4
1.2	Evaluación y Control de riesgos	06/02/17	06/04/17	3
2	Diseño y Desarrollo	06/05/17	10/15/17	133
2.1	Diseño de Arquitectura	06/05/17	06/11/17	7
2.2	Diseño Métrica para el Sistema Recomendador	06/12/17	07/09/17	28
2.2.1	Diseño Encuestas para determinación de Métrica	06/12/17	06/18/17	7
2.2.2	Implementación de Encuestas	06/19/17	06/25/17	7
2.2.3	Interpretación de Datos Encuestas	06/26/17	07/02/17	7
2.2.4	Planteamiento de Métrica	07/03/17	07/09/17	7
2.3	Diseño y Desarrollo de la Persistencia de Datos	07/10/17	07/23/17	14
2.3.1	Diseño Cadena Lógica de Negocio	07/10/17	07/13/17	4
2.3.2	Construcción de Tablas - Modelo DF	07/14/17	07/16/17	3
2.3.3	Diseño Diagrama Bases de Datos	07/17/17	07/20/17	4
2.3.4	Diseño Modelo de Definición de Datos	07/21/17	07/23/17	3
2.4	Diseño y Desarrollo de Modulo Recomendador	07/24/17	08/27/17	35
2.4.1	Diseño y Desarrollo Método Basado en Contenido	07/24/17	08/02/17	10
2.4.2	Diseño y Desarrollo Método Colaborativo	08/03/17	08/13/17	11
2.4.3	Diseño e integración Método híbrido	08/14/17	08/27/17	14
2.5	Diseño y Desarrollo del Módulo de Gamificación	08/28/17	09/17/17	21
2.5.1	Diseño y Desarrollo de Sistema de Puntos	08/28/17	09/10/17	14
2.5.2	Diseño y Desarrollo de módulo de Redes Sociales	09/11/17	09/17/17	7
2.6	Diseño y Desarrollo de Módulos extras	09/18/17	10/15/17	28
2.6.1	Diseño y Desarrollo de Interfaz Gráfica	09/18/17	09/24/17	7
2.6.2	Diseño y Desarrollo de Módulo de Registros y Login	09/25/17	10/01/17	7
2.6.3	Diseño y Desarrollo de Módulo de Información	10/02/17	10/08/17	7
2.6.4	Diseño y Desarrollo de Módulo de Perfil de Usuario	10/09/17	10/15/17	7
3	Cierre	10/16/17	11/30/17	46
3.1	Integración	10/16/17	10/22/17	7
3.2	Pruebas del Sistema	10/23/17	10/29/17	7
3.3	Documentación	10/30/17	11/29/17	31
3.4	Entrega y Sustentación	11/30/17	11/30/17	1

Tabla 1. Tabla Actividades Cronograma (Autores)

10.2 DIAGRAMA DE GANTT

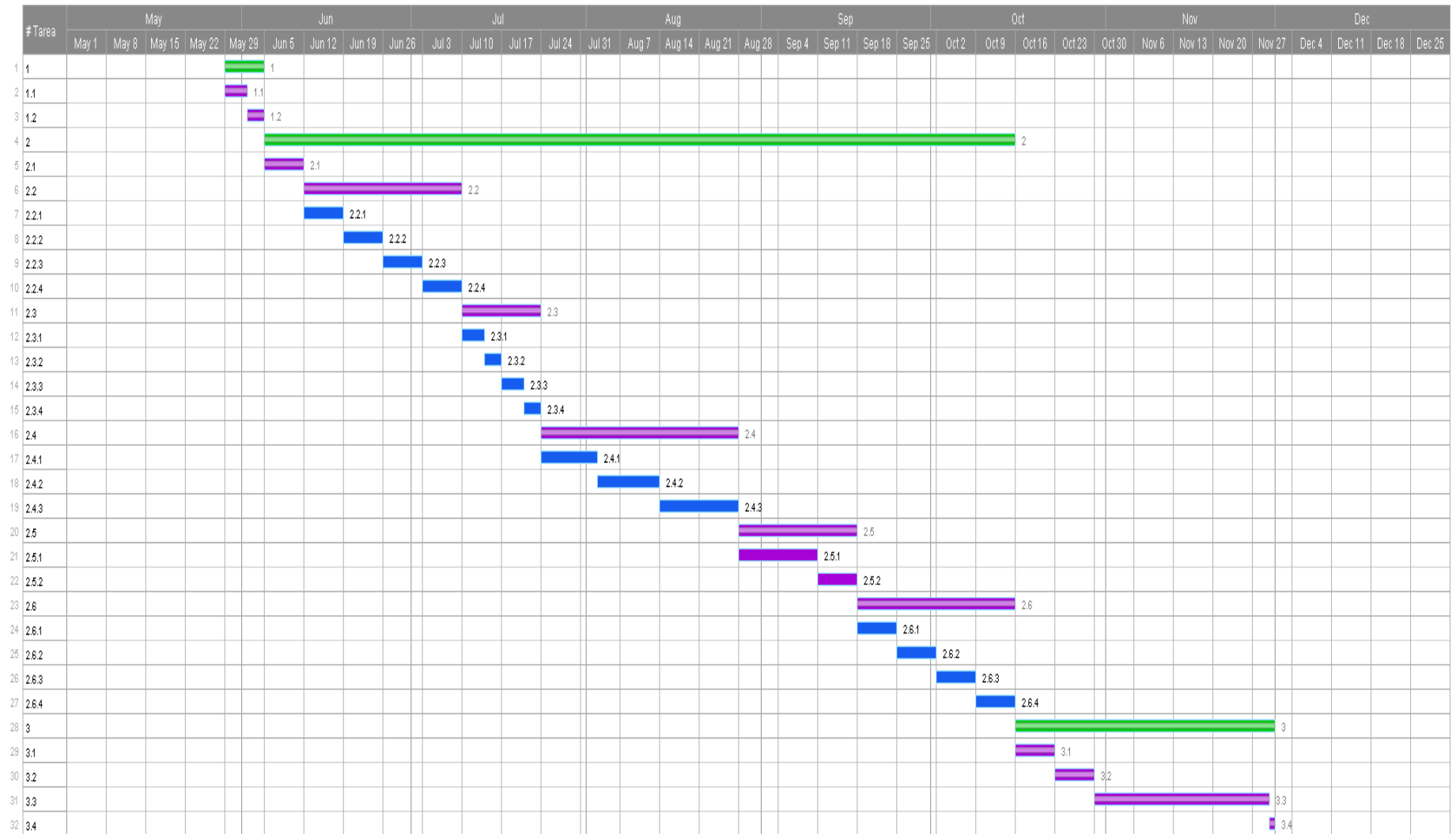


Figura 1. Diagrama de Gantt (Autores)

11. PRESUPUESTO

11.1 COSTOS POR SERVICIOS PERSONALES

Los costos por servicios personales incluyen el trabajo hecho por los dos autores del proyecto. Basándose en un salario mensual de los ingenieros y considerando que le dedicaran aproximadamente medio tiempo a su proyecto y que la duración del proyecto será desde junio 2017 a Noviembre 2017, los costos son:

Salario base: \$2.500.000

Ingeniero	Salario m/t	Cantidad	Total Ingeniero
Xenia Vivian Cadena Díaz	\$1.250.000	6 Meses	\$7.500.000
Nelson Arley Carantón Galeano	\$1.250.000	6 Meses	\$7.500.000
TOTAL COSTOS SERVICIOS PERSONALES			\$15.000.000

Tabla 2. Costos por servicios personales (Autores)

11.2 GASTOS GENERALES

Los costos directos generados por el proyecto se calculan así:

Gasto	Costo Unitario	Costo x 2	Costo X 6 M
Transporte	\$100.000	\$200.000	\$1.200.000
Papelería	\$40.000	\$80.000	\$80.000
Impresión	\$10.000	\$10.000	\$10.000
Dominio	\$4.000	\$8.000	\$48.000
Host	\$6.000	\$12.000	\$72.000
Internet Hogar	\$65.000	\$130.000	\$780.000
Datos Celular	\$40.000	\$80.000	\$480.000
TOTAL GASTOS GENERALES			\$2.670.000

Tabla 3. Gastos generales (Autores)

11.3 COSTO TOTAL DEL PROYECTO

Los costos totales del proyecto son la suma de los costos por servicios personales y los gastos generales. Quedan así:

Descripción costo	Valor Costo
Costos por servicios personales	\$15.000.000
Gastos generales	\$2.670.000
TOTAL COSTOS DEL PROYECTO	\$17.670.000

Tabla 4. Costo total del proyecto (Autores)

BIBLIOGRAFÍA

- [1] E. Herrera-Viedma, L. Olvera, E. Peis, C. Porcel. "Revisión de los sistemas de recomendaciones para la recuperación de información. Tendencias de investigación en organización del conocimiento (Trends in knowledge organization research)" José Antonio Frías, Ed. Crispulo Travieso, Universidad de Salamanca, (2003), 507-513.
- [2] R.R. Yager. "Fuzzy Logic Methods in Recommender Systems". Fuzzy Sets and Systems, Volume 136, Issue 2 (2003), 133-149.
- [3] G. Bafoutsou, G. Mentzas. "Review and Functional Classification of Collaborative Systems". International Journal of Information Management, 22 (2002), 281-305.
- [4] P. Resnick, H.R. Varian, Guest Editors. Recommender Systems. Communications of the ACM, 40 (3) (1997), 56-89.
- [5] BunchBall "An Introduction to the Use of Game Dynamics to Influence Behavior" Gamification 101 (2010)
- [6] Ramírez J.L. "Gamificación. Mecánicas de juegos en tu vida personal y profesional" Alfaomega (2014)
- [7] J.B. Schafer, J.A. Konstan, J. Riedl. "MetaRecommendation Systems: User-controlled Integration of Diverse Recommendations". ACM Conference on Information and Knowledge Management (CIKM-02), November 5-7, 2002, McClean, VA.
- [8] Vélez Langs, O. Santos, C. Aproximando a los Sistemas Recomendadores desde los Algoritmos Genéticos. REVISTA COLOMBIANA DE COMPUTACIÓN. VII- 2. 9. 2006.
- [9] L. Bejarano, J. Meza. Clase Seminario de Investigación I – Especialización en Ingeniería de Software – Universidad Distrital Francisco José de Caldas (2017)

- [10] ¿Qué es una encuesta? Características Generales. - Estadistica.mat.uson (2010) - <http://www.estadistica.mat.uson.mx/Material/queesunaencuesta.pdf>
- [11] M. Arriaza Balmón. “Guía práctica de análisis de datos” Consejería de Innovación, Ciencia y Empresa. Ideagonal Diseño Gráfico.[21-32] (2008)
- [12] E. Herrera, C. Porcel, L. Hidalgo. “Sistemas de recomendaciones: herramientas para el filtrado de información en Internet” – Universidad Pompeu Fabra, 2 Edición (2004)
- [13] Ramírez J.L. “Gamificación. Mecánicas de juegos en tu vida personal y profesional” Alfaomega (2014)
- [14] Extractado de La Guía de Arquitectura Versión 2.0a del grupo de Patterns and Practices de Microsoft.
- [15] Tomado del sitio oficial de Spring <https://spring.io/docs> Última revisión 2017
- [16] Schwaber, Sutherland. (2013). La Guía Definitiva de Scrum: Las Reglas del Juego. Recuperado de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- [17] A. Menzinsky, G. López, J. Palacio. “Scrum Manager BoK” Cap.1 (V 2.6 - 2016) tomado de https://www.scrummanager.net/bok/index.php?title=Modelo_original_de_Scrum_para_desarrollo_de_software