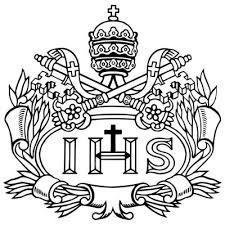
Pontificia Universidad Javeriana



Ingeniería Sistemas

Introducción a la ingeniería de software

Anabel Montero

**Proyecto: Primera Entrega  
Grupo: Teletubbies**

***Autores:***

Esteban Villa Berrio: estebana-villa@javeriana.edu.co

Sergio Triana: triana\_se@javeriana.edu.co

Nicolas Barragan:[nbarragans@javeriana.edu.co](mailto:nbarragans@javeriana.edu.co)

Santiago Torres:[sa-torres@javeriana.edu.co](mailto:nbarragans@javeriana.edu.co)

Julián Rizo:[jr.rizo@javeriana.edu.co](mailto:nbarragans@javeriana.edu.co)

Cristian David Gonzalez:[cd-gonzalez@javeriana.edu.co](mailto:nbarragans@javeriana.edu.co)

Julián Carrillo:[jacarrilloc@javeriana.edu.co](mailto:nbarragans@javeriana.edu.co)

**Historial de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| Número de cambio | Fecha del cambio | Descripción | Integrante que hizo el cambio |
| 1 | 26/02/2021 | Se añadió la descripción y la propuesta del proyecto | David González |
| 2 | 27/02/2021 | Se añadió tabla de prioridad y riesgo. Se añadieron 12 historias de usuario. | David González Julián Rizo |
| 3 | 27/02/2021 | Se añadieron características del usuario | Julián Rizo David González |
| 4 | 27/02/2021 | Se terminaron las tablas y se agregaron los mockups | Esteban Villa, Nicolas Barragan, Sergio Triana, Julián Carillo, Santiago Torres |
| 5 | 6/03/2021 | Introducción a funciones del producto | Julián Rizo |
| 6 | 6/03/2021 | Mockups historias de usuario 1-6 | Julián Rizo |
| 7 | 7/03/2021 | Adelanto de propósito de requisitos no funcionales | Santiago Torres |
| 8 | 07/03/2021 | Adelanto de modelo de Ciclo de vida | Nicolás Barragán |
| 9 | 07/03/2021 | Se añadió el plan de trabajo del proyecto | David González |
| 10 | 06/03/2021 | Se terminó el demo para la presentación del proyecto | Esteban Villa |
| 11 | 07/03/2021 | Agregar conclusión | Esteban Villa |
| 12 | 07/03/2021 | Se termina modelo de ciclo de vida y se agrega conclusión | Nicolás Barragán |
| 13 | 07/03/2021 | finaliza restricciones generales | Santiago Torres |
| 14 | 07/03/2021 | realización del apartado lenguajes y herramientas | Sergio Triana |
| 15. | 07/03/2021 | Corrección a historias de usuario, userStoryMap y trello. | Nicolás Barragán y Sergio Triana |
| 16 | 07/03/2021 | Se agregan lista de figuras y de tablas. Se junta todo el contenido de la entrega en un ZIP | Sergio Triana y Nicolás Barragán. |

**Tabla de contenido**

[1 Introducción 6](#_Toc66110838)

[2 Propuesta del proyecto 7](#_Toc66110839)

[3 Modelo de ciclo de vida 8](#_Toc66110840)

[4 Equipo de trabajo y distribución de roles 9](#_Toc66110841)

[5 Lenguajes y herramientas 11](#_Toc66110842)

[6 Funciones del producto 12](#_Toc66110843)

[6.1 Características del usuario 14](#_Toc66110844)

[Consumidor 14](#_Toc66110845)

[Desarrollador 15](#_Toc66110846)

[Administrador Rocket 15](#_Toc66110847)

[7 Requisitos No Funcionales 16](#_Toc66110848)

[8 Plan de trabajo del proyecto 20](#_Toc66110849)

[9 Métodos y herramientas de Estimación 22](#_Toc66110850)

[10 Conclusiones 23](#_Toc66110851)

[11 Anexos 23](#_Toc66110852)

[12 Referencias 24](#_Toc66110853)

**Lista de figuras**

[Ilustración 1: Organigrama 10](#_Toc66104834)

[Ilustración 2: Ejemplo Mockup historia de usuario 14](#_Toc66104835)

[Ilustración 3: Ejemplo historias de usuario miro 21](#_Toc66104836)

[Ilustración 4: Ejemplo historias de usuario product backlog 21](#_Toc66104837)

[Ilustración 5: Ejemplo columnas Trello 22](#_Toc66104838)

**Lista de tablas**

[Tabla 1: Equipo de trabajo y roles 10](#_Toc66104768)

[Tabla 2: Prioridad 12](#_Toc66104769)

[Tabla 3: Riesgo 13](#_Toc66104770)

[Tabla 4: Ejemplo de historia de usuario 13](#_Toc66104771)

[Tabla 5: Consumidor 14](#_Toc66104772)

[Tabla 6: Desarrollador 15](#_Toc66104773)

[Tabla 7: Administrador Rocket 15](#_Toc66104774)

[Tabla 8: Estética 16](#_Toc66104775)

[Tabla 9: Restricción consistencia 17](#_Toc66104776)

[Tabla 10: Restricción documentación 17](#_Toc66104777)

[Tabla 11: Restricción precisión 17](#_Toc66104778)

[Tabla 12: Restricción velocidad 18](#_Toc66104779)

[Tabla 13: Restricción eficiencia 18](#_Toc66104780)

[Tabla 14: Restricción tiempo de respuesta 18](#_Toc66104781)

[Tabla 15: Restricción adaptabilidad 19](#_Toc66104782)

[Tabla 16: Restricción disponibilidad 19](#_Toc66104783)

[Tabla 17: Restricción compatibilidad 19](#_Toc66104784)

[Tabla 18: Restricción extensibilidad 19](#_Toc66104785)

[Tabla 19: Restricción lenguaje de programación 20](#_Toc66104786)

[Tabla 20: Restricción dispositivos y conexión a internet 20](#_Toc66104787)

# Introducción

El mercado de los videojuegos es complejo y tiene muchas variables a considerar. Los usuarios, el diseño, los desarrolladores, todo el marketing que está implícito en cada título que sale al mercado opaca constantemente a los pequeños y medianos desarrollos que muchas veces por falta de financiación mueren ante los grandes de la industria. Esto lleva a la desmotivación de dichos creadores que tuvieron una idea que no pudo surgir y que posteriormente podrían pensar en dejar de hacer lo que le gusta al no tener oportunidad en el mercado.

Por esto surge la iniciativa “Rocket”. Esta permite avivar la cultura de la donación y que esos desarrolladores pequeños y medianos generen una comunidad a través de la cual puedan proyectar sus juegos y sus futuros desarrollos. Teniendo así, una oportunidad en un nuevo mundo donde no serán opacados por enormes compañías y tendrán una comunidad que los respalde.

# 2 Propuesta del proyecto

Nosotros como desarrolladores creemos que los buenos títulos y la mejor jugabilidad no debe estar permeada de solamente dinero, sino también de grandes ideas que no necesariamente por estar aún en proceso de desarrollo, sean malas. Así nació “Rocket”.

Nuestro proyecto quiere darle una oportunidad a los pequeños y medianos desarrolladores de entrar al mercado de los videojuegos sin tener que ser aplastados por las grandes industrias de desarrollo del mundo. Con la posibilidad de trabajo conjunto, ganar un equipo y promocionar sus juegos con interfaces amigables con alta difusión a nivel global. El marketing exagerado dejará de ser la razón por la que los juegos subirán al top list de los mejores juegos de la temporada, ya que “Rocket” impulsará esos juegos que antes quizás pudieron ser invisibles para poder competir con los grandes y las grandes casas desarrolladoras.

Rocket se basa de una plataforma donde los desarrolladores individuales, pequeños y/o medianos pueden registrarse. Posteriormente podrán solicitar la publicación de su(s) videojuego(s) para que la comunidad los adquiera. Para esto el equipo Rocket validará la originalidad, jugabilidad y calidad del videojuego para proponer un precio de venta que el desarrollador esté de acuerdo, y así, ofrecerlo a la comunidad. Además, Rocket cuenta con un sistema de donaciones para aquellos desarrolladores que requieren dinero puedan postularse para continuar con sus proyectos para que no queden estancados a la mitad.

# 3 Modelo de ciclo de vida

Los proyectos de desarrollo de software tienen un ciclo de vida. Es decir, actividades y tareas que se involucran en el desarrollo y mantenimiento de un producto de software, abarcando desde la definición del proyecto hasta la finalización.

Para llevar el control y progreso de este ciclo de vida existen diferentes modelos que llevan la gestión de acciones, procesos, resultados de procesos, relaciones y secuencias. Existen distintos modelos como, por ejemplo: en cascada, incremental, evolutivo, ágiles.

Al iniciar el desarrollo de un proyecto de software se debe definir qué modelos serán utilizados para el control de este. Para esto partimos de definir ciertos criterios respecto a nuestro proyecto de los cuales el modelo debe cumplir en su mayoría.

**Criterios para la selección de un modelo de ciclo de vida**

* Tipo de proyecto: No todos los proyectos son iguales. Por tanto, se define nuestro tipo de proyecto como único, ya que es un proyecto que no busca actualizarse a futuro y que al cabo de 3 meses ya será descontinuado.
* Estándares en el tipo de proyecto: Cuáles son los modelos de ciclos de vida más usados de acuerdo con el tipo de proyecto que se está desarrollando.
* Adaptabilidad con otros modelos: Qué tan fácil es tomar características de otros modelos de ciclo de vida e incluirlas en la que se quiere utilizar. Se quiere ver la flexibilidad del modelo para juntarse con otros.
* Qué tan cómodo está el equipo con el modelo: Qué tanta experiencia tiene el equipo con el modelo. Si ya lo han manejado, o apenas lo conocen teóricamente. Probablemente quien no se sienta cómodo con el modelo puede llegar a no entender el proceso y perderse, causando así un retraso en el desarrollo del proyecto de software.
* Rendimiento y comodidad para con el proyecto: Qué tan rápido nos permitirá avanzar con el proyecto. Qué tan fácil se adapta al tipo de proyecto y qué tanto beneficiará al proyecto y su desarrollo.

**Modelos analizados**

A continuación, se presentan los modelos analizados enseguida de una descripción breve de lo que se habló de los mismo al proponerlos como modelos de ciclo de vida para este proyecto.

* **Ágiles: SCRUM:** Método ágil muy conocido. Perfecto que requiere rapidez y flexibilidad. Para equipos de 2 a 7 personas. Se adapta rápido al cambio según las necesidades.
* **Incremental:** Se pensaba bueno dado que cada parte que se va terminando se puede utilizar. Frente a cualquier error se desecha la última iteración de trabajo. En nuestro caso nuestro proyecto tiene únicamente 2 iteraciones de las cuales en la primera se entrega casi todo el proyecto. [3]
* **Cascada:** Requiere más documentación de la que teníamos pensada para este proyecto. Al ser un proyecto que puede ir cambiando en cuanto a requisitos resulta inviable. [2]

**Modelo seleccionado**

**SCRUM**

Elegimos scrum para el desarrollo del proyecto ya que se adapta rápido al tipo de proyecto. Es perfecto para la cantidad de personas que pertenecemos al equipo. Nuestro proyecto tiene 2 iteraciones y esta metodología es perfecta para llevar un buen control y repartición de tareas para la culminación de esta. Aquí hacemos una repartición de roles, y planificamos Sprints.

SCRUM nos permite trabajar de acuerdo con incrementos, planteando un producto backlog, sprint backlog. A esta metodología adicionamos el trabajo a partir de TRELLO, herramientas que nos ayuda a llevar el control y el progreso de las historias de usuario.

Los roles se distribuyen así:

* SCRUM máster: Nicolás Barragán.
* Product Owner: Sergio Triana. (Sergio servirá de producto owner, es decir, se pondrá en posición de cliente dueño para que nuestro producto final sea óptimo)
* SCRUM team: Julián Rizo, Julián Carrillo, Cristian González, Santiago Torres, Esteban Villa.

**Apartado 7 con referencias de [1]**

# 4 Equipo de trabajo y distribución de roles

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Rol | Descripción | Responsabilidades |
| David González | * Encargado de planeación del proyecto * Desarrollador | Revisa y valida la calidad de las actividades asignadas a cada uno de los integrantes del proyecto. | Se encarga de modelar la planeación del proyecto y según lo que el líder del proyecto dice, asignar responsabilidades en los sprints. |
| Nicolás Barragán | * Líder * Desarrollador | guía al equipo y lleva control del avance. Motiva al equipo al trabajo oportuno y de calidad. | Revisar partes terminadas. Aconsejar. Desarrollar funcionalidades. |
| Sergio Esteban Triana | * Gestor de herramientas lenguajes y ambiente de desarrollo * Desarrollador | Selección de lenguajes de programación que mejor se adaptan a nuestro proyecto, desarrollador de la página web | Desarrollador de la página web, designación de los lenguajes de programación |
| Santiago Torres | * Analista de requerimientos y restricciones * Desarrollador |  |  |
| Esteban Villa | * Diseño front ende * Desarrollador | Desarrollo del diseño general del proyecto | Gestor inicial del Diseño en la implementación del demo para así más adelante desarrollarlo en código. |
| Julián Rizo | * Encargado estimación * Desarrollador | Coordina el equipo para las reuniones de estimación de actividades, y desarrollador de la página web | Gestión de reuniones de estimación de actividades, gestión de herramientas de estimación. |
| Julián Carrillo | * Analista de requerimientos y restricciones * Desarrollador |  |  |

Tabla 1: Equipo de trabajo y roles

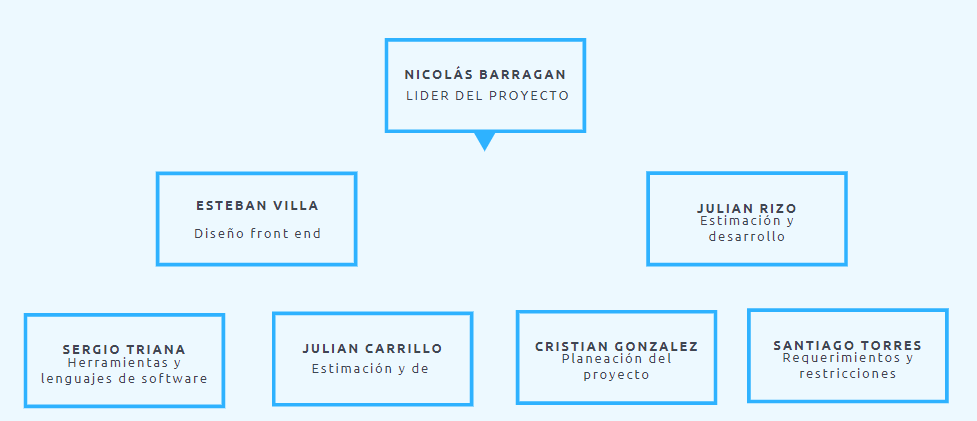


Ilustración 1: Organigrama

# 5 Lenguajes y herramientas

Para la correcta realización de nuestro proyecto se identificaron claramente un abanico de herramientas y lenguajes de programación los cuales facilitarán y generarán un flujo de trabajo estable y eficiente.

en este apartado se describirán aquellas herramientas utilizadas para lograr el objetivo principal del proyecto ROCKET.

**HERRAMIENTAS:**

1. **Star UML**

Esta Herramienta nos proporciona un ambiente de modelado en donde generamos un diagrama de casos de uso donde se pusieron todos aquellos casos identificados para darle una experiencia al usuario completa y muy buena.[4]

1. **MIRO**

Es un software online colaborativo en el cual se realizó el desarrollo de un User Story Map (Mapa de Historias de Usuario) en el que se dejó registro de todas aquellas funcionalidades que se desarrollarán en 2 sprints: product backlog 1 y product backlog 2. Manteniendo una guía de flujo de trabajo entre los involucrados estando a distancia.[5]

Cada uno de los sprints nombrados anteriormente dejarán un avance del desarrollo del proyecto cumpliendo con las historias de usuario estipuladas para cada uno de estos.

# 

1. **Balsamiq**

Es un software de modelado el cual nos facilita el desarrollo de los mockups requeridos para hacer las vistas principales de cada historia de usuario y hacernos a la idea de un posible resultado final, los mockups nos sirven para guiarnos en el desarrollo visual de nuestro aplicativo y poder tener una referencia de a dónde queremos llegar en el producto final.[6]

1. **Visual Studio Code**

Es un entorno para editar código fuente, esta herramienta la usamos para poder editar nuestro código fuente del proyecto, además que tiene la capacidad de poder reconocer varios lenguajes de programación lo que nos facilita el desarrollo.[7]

**LENGUAJES:**

1. **HTML**

HTML es un lenguaje de programación destinado al desarrollo de páginas web, fue uno de los lenguajes seleccionados para realizar nuestro proyecto, este lenguaje nos ayuda a codificar toda la estructura básica de nuestro proyecto.[8]

1. **CSS (CASCADE STYLE SHEETS)**

Este lenguaje de programación está enfocado al diseño gráfico y el papel que juega en nuestro proyecto va acompañado a darle un apartado visual a nuestra estructura dada por el HTML pensando en llegar a lo presentado en los mockups desarrollados. Con este lenguaje podemos modificar las vistas de nuestra página de manera fácil y eficaz.[9]

1. **JAVASCRIPT**

Lenguaje enfocado a darles mejoras a las interfaces de usuario y a las páginas web dándole capacidad de hacer desarrollos más eficaces y mejor para el usuario final. con este lenguaje podremos hacer el desarrollo del comportamiento de los objetos de la página web.[10]

# 6 Funciones del producto

Para describir las funcionalidades del producto, se desarrollaron casos hipotéticos con el usuario definido y la funcionalidad que se usa en dicho caso, esto es llamado historia de usuario; en esta también se encontrará la prioridad de dicha funcionalidad en el producto, y el riesgo de esta, teniendo en cuenta la complejidad, importancia y tiempo de demanda. Los atributos que se pueden dar en estos dos últimos campos explicados se describen en una tabla para su comprensión.

A continuación, se describirán todas las historias de usuario que podrían ocurrir para describir el funcionamiento del producto.

|  |  |
| --- | --- |
| **Prioridad** | **Descripción** |
| **Alta** | Fundamental para el funcionamiento del negocio o plataforma |
| **Alta/Media** | Necesaria para el correcto funcionamiento del negocio o plataforma |
| **Media** | Útil para el funcionamiento del negocio y sin esta alguna funcionalidad fallaría |
| **Media/Baja** | Necesaria pero no fundamental, asociada a alguna otra funcionalidad de prioridad alta, media/alta o media. |
| **Baja** | Intrascendente para el funcionamiento de la plataforma o modelo de negocio, y útil en pocas ocasiones |

Tabla 2: Prioridad

|  |  |
| --- | --- |
| **Riesgo** | **Descripción** |
| **Alto** | Presenta un alto nivel de complejidad y de tiempo para demandar |
| **Medio** | Presenta un nivel de complejidad medio |
| **Bajo** | Presenta un bajo nivel de complejidad y desarrollarse no representa un riesgo para el proyecto |

Tabla 3: Riesgo

Tabla 4: Ejemplo de historia de usuario

|  |  |
| --- | --- |
| **Historia de usuario** | |
| **Número: 1** | **Usuario: Consumidor y Desarrollador** |

|  |  |
| --- | --- |
| **Nombre de la historia:** “Como usuario y desarrollador quiero buscar videojuegos” | |
| **Prioridad en negocio: Alta/Media** | **Riesgo en desarrollo: Bajo** |
| **Puntos/Horas estimados: 5h** | **Iteración asignada: 1° Product backlog** |
| **Programador responsable: Julián Rizo** | |
| **Descripción: En la página web se debe tener siempre la funcionalidad de buscar por nombre los videojuegos que coincidan con este.** | |
| **Criterios de aceptación: Los resultados de la búsqueda son correctos el 100% de las veces.** | |
| **Validación: Se prueba diversas búsquedas con diversos nombres, el resultado debe ser congruente con estos.** | |

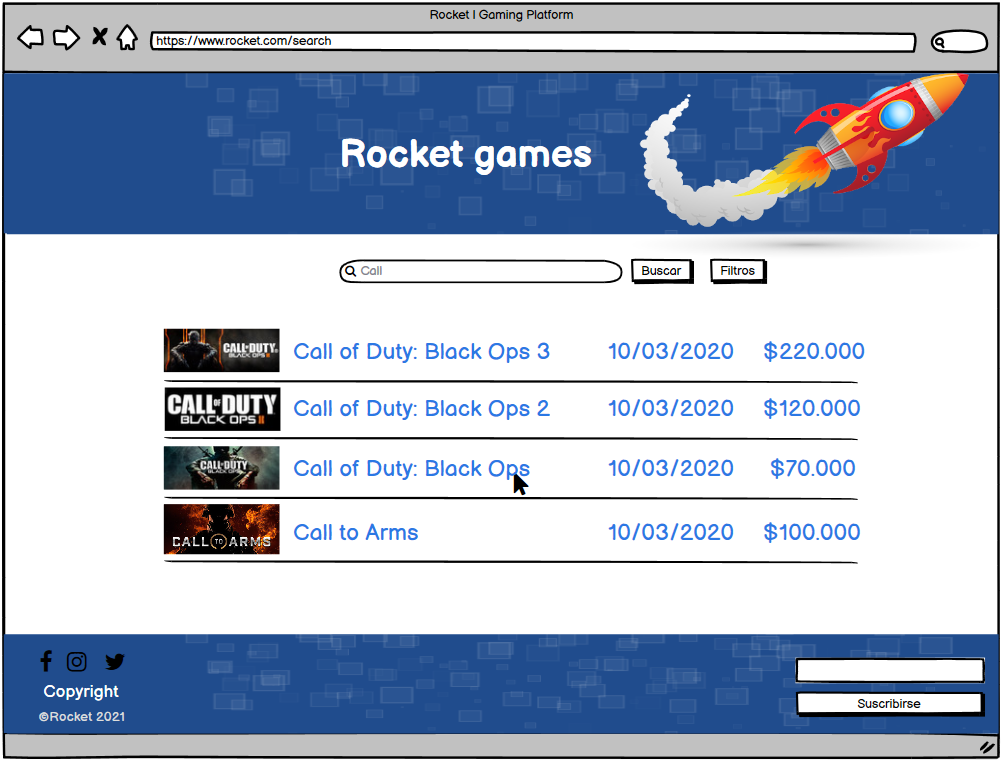
****

Ilustración 2: Ejemplo Mockup historia de usuario

**Se mostró un ejemplo de historia de usuario con mockup y documentación. El resto se encuentra en anexos.**

## 6.1 Características del usuario

### Consumidor

Tabla 5: Consumidor

|  |  |
| --- | --- |
| **Características del usuario:** | **Descripción** |
| **Nivel de seguridad o de privilegios** | 1. Privacidad con datos personales y financieros. 2. Realizar donaciones. 3. Realizar compra de videojuegos. 4. Postear en los foros. |
| **Roles** | Es el principal usuario a la hora de navegar en la página, realizar búsquedas filtradas de su preferencia, realizar pagos, donaciones, y también hacer uso de los foros que tendrá la página. |

|  |  |
| --- | --- |
|  |  |
| **Nivel de estudios o experiencia técnica** | Nivel bajo en navegación dentro de páginas web.  Nivel medio en búsquedas filtradas básicas dentro de páginas web. |
| **Frecuencia de uso** | Alta frecuencia de uso de la plataforma, en su mayoría diaria. |

*Tabla 1. Descripción de las características del usuario*

### Desarrollador

Tabla 6: Desarrollador

|  |  |
| --- | --- |
| **Características del usuario** | **Descripción** |
| **Nivel de seguridad o de privilegios** | 1. Edición de textos de sus propios proyectos 2. Mandar solicitud para retirar donaciones 3. Generar pagaré para recibir donaciones 4. Soporte rápido para necesidades dentro de la plataforma. |
| **Roles** | Es el administrador de los desarrollos a los que está referenciado y tiene la posibilidad de editar la página de sus juegos. Se encarga de desarrollar los aspectos básicos y avanzados del juego y junto a su equipo de Desarrolladores darle visibilidad al proyecto. |

|  |  |
| --- | --- |
|  |  |
| **Nivel de estudios o experiencia técnica** | Nivel medio/alto en desarrollo de Software y videojuegos.  Nivel medio/alto en edición y diseño |
| **Frecuencia de uso** | Alta periodicidad, con entradas en su mayoría diarias a la plataforma. |

### Administrador Rocket

Tabla 7: Administrador Rocket

|  |  |
| --- | --- |
| **Características del usuario** | **Descripción** |
| **Nivel de seguridad o de privilegios** | Todos los privilegios sobre la configuración de la página web.  Ver información únicamente personal de usuarios y desarrolladores. |
| **Roles** | Es el administrador del sitio web y moderador en los foros. |

|  |  |
| --- | --- |
|  |  |
| **Nivel de estudios o experiencia técnica** | Nivel medio/alto en desarrollo de Software.  Nivel medio/alto en administración de plataformas. |
| **Frecuencia de uso** | Diario y en algunos casos semanal, se debe estar pendiente para revisar los tickets de soporte que se hayan mandado por los otros desarrolladores o consumidores. |

# 7 Requisitos No Funcionales

Este sistema está restringido bajo los factores de calidad de usabilidad, confiabilidad, prestación y soporte (FURPS+) que serán desarrollados a continuación; este sistema al ser un software desarrollado en java script y siendo un proyecto de desarrollo web contiene unas características propias de dichos proyectos y el propósito de este apartado es aclarar y definir de manera clara y concisa las restricciones generales del sistema además de las restricciones de software y hardware del mismo, cabe aclarar también que los requisitos presentados en esta sección son los mínimos que debe cumplir este proyecto mas no los deseables para este.

Contenido

Restricciones generales: El proyecto rocket está regido por los siguientes componentes: estética, consistencia, documentación, precisión, velocidad, eficiencia, tiempo de respuesta, adaptabilidad, disponibilidad, compatibilidad y extensibilidad.

Tabla 8: Estética

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Estética | Es necesario tener una buena estética en la página web para aumentar su usabilidad y tener una apariencia más amigable, sencilla y vistosa a los ojos del usuario. | Usabilidad |

Tabla 9: Restricción consistencia

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Consistencia | Es necesario tener unos convenios establecidos con respecto a la interfaz gráfica del proyecto, tener en cuenta el uso de iconos estandarizados(conocidos), del mismo tamaño y con el mismo diseño, a su vez también es útil conocer convenios como siempre dejar la barra de búsqueda en un lugar fijo sin importar el cambio de pestaña, el color de los botones de aceptar debe ser llamativo (verde o amarillo) y el de cancelar debe ser rojo, etc. | Usabilidad |

Tabla 10: Restricción documentación

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Documentación | Para tener una correcta usabilidad es necesaria una documentación de diseño que sea entregada al cliente o sirva como estándar, forma de comunicación y flujo de información para los miembros del equipo. | Usabilidad |

Tabla 11: Restricción precisión

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Precisión | Para cumplir a cabalidad los objetivos de un proyecto es necesario tener información confiable de forma precisa y rápida, en este proyecto no es la excepción y se busca continuamente la información más útil para el desarrollo de este con la fuente más confiable posible. | Confiabilidad |

Tabla 12: Restricción velocidad

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Velocidad | Las transiciones, cambios de página, apertura de menús, menús de fotos, presentación de títulos, opciones de compra, sistema de transacciones y búsqueda de títulos/donaciones deben de ser fluidas y rápidas. | Presentación |

Tabla 13: Restricción eficiencia

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Eficiencia | La página web debe ser eficiente y funcional para el público de usuarios estimados, no se puede caer, no debe entorpecer transacciones ni búsquedas. | Presentación |

Tabla 14: Restricción tiempo de respuesta

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Tiempo de Respuesta | La página web debe tener un tiempo de respuesta aceptable para el público de usuarios estimados, no se puede tardar en búsquedas, no debe de tardar en cargar interfaces ni multimedia, los cambios de página deben tener un tiempo de respuesta aceptable y que muestre una presentación adecuada sobre el producto al usuario. | Presentación |

Tabla 15: Restricción adaptabilidad

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Adaptabilidad | El equipo, el desarrollo del proyecto y el proyecto en sí deben estar preparados para el cambio o la adición de nuevos requisitos por parte del cliente o por necesidad misma del proyecto. | Soporte |

Tabla 16: Restricción disponibilidad

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Disponibilidad | El servicio de Rocket debe estar siempre disponible, al ser una página web debe de tener disponibilidad para el usuario 24/7. | Soporte |

Tabla 17: Restricción compatibilidad

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Compatibilidad | El servicio de Rocket debe ser compatible con los estándares y los dominios establecidos en la web para ser un servicio web como tal, además de ser compatible con ciertas herramientas de desarrollo para garantizar unas mayores y más fáciles capacidades de adaptabilidad y extensibilidad al proyecto. | Soporte |

Tabla 18: Restricción extensibilidad

|  |  |  |
| --- | --- | --- |
| Restricción: | Descripción: | Clasificación FURPS+ |
| Extensibilidad | El servicio de Rocket debe estar preparado para adaptar o añadir más contenido en un futuro sin dañar el producto base aquí presentado con el fin de aumentar la capacidad funcional del servicio web. | Soporte |

* ·Restricciones de software: Como todo programa ejecutable es necesario tener presente diferentes características que van a actuar como limitante al momento de ejecución y desarrollo de las actividades de dicho programa, y el proyecto Rocket, no es la excepción al caso anteriormente mencionado. Para Rocket las limitaciones que existen a nivel de software son las mismas que presenta el lenguaje utilizado para el desarrollo de este.

Tabla 19: Restricción lenguaje de programación

|  |  |
| --- | --- |
| Restricción | Descripción |
| Lenguaje de programación | Restricciones generadas por el mismo lenguaje usado, que para este caso es JavaScript con sus determinadas limitaciones. |
| Navegador | Es necesario tener instalado en el dispositivo un navegador con la posibilidad de acceder a la red, para así, poder ejecutar el sistema rocket, además de la limitación generada por los dominios web |

* Restricciones de hardware: Para la ejecución de Rocket es necesario cumplir con ciertos requisitos mínimos a nivel de hardware para poder ejecutar el aplicativo

Tabla 20: Restricción dispositivos y conexión a internet

|  |  |
| --- | --- |
| Restricción | Descripción |
| Dispositivo | Es necesario un dispositivo donde sea posible contener un navegador con la capacidad de acceder a Rocket, y así acceder a Rocket |
| Conexión a internet | Es necesario que el dispositivo anteriormente señalado, cuente con una conexión a internet estable, así sea cableada o vía inalámbrica (wifi) |

# 8 Plan de trabajo del proyecto

Rocket es un proyecto con demasiadas funcionalidades, desde la donación hasta la búsqueda de juegos con filtros específicos que permitan organizar la información de la mejor forma para sus usuarios. A partir de estas, hemos visto la necesidad de implementar un mapa de historias de usuario en la plataforma de “miro”, que nos permita tener un control del desarrollo desde las funcionalidades más avanzadas hasta las más básicas del proyecto.

La herramienta está dividida en “historias épicas” e “historias generales” (imagen x). Las historias generales en color verde se encargan de dividir las más grandes funcionalidades que queremos que tenga el proyecto “Publicar videojuegos”, “Comprar videojuegos” y “Donaciones”. Las historias épicas en color amarillo son formas de dividir “las historias generales” en funcionalidades más específicas, que a la larga nos ayudan a dividir la aplicación correctamente. Y, por último, las historias de usuario en color azul donde se tienen las características en su forma más específica para poder dividir su desarrollo en las diferentes etapas del “Product Backlog 1” y el “Product Backlog 2” (Imagen x).

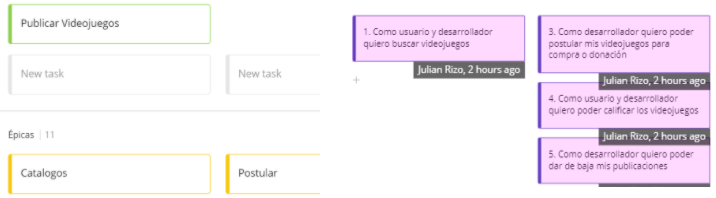


Ilustración 3: Ejemplo historias de usuario miro

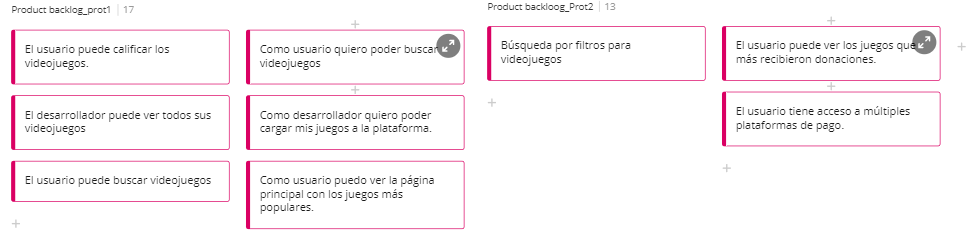


Ilustración 4: Ejemplo historias de usuario product backlog

El tablero completo de historias de usuario en los anexos o en el siguiente enlace: https://miro.com/app/board/o9J\_lUGEOEc=/.

Con el fin de mantener control sobre el proyecto, el equipo se ha encargado de añadir un tablero en Trello que además de permitir tener todas las historias de usuario, ayuda a cada uno de los integrantes a conocer el trabajo que tienen disponible para realizar en las etapas del sprint.

Dividido en secciones, el tablero de Trello tiene “Questions for the next meeting”, “to do”, “1 product backlog”, “doing” y “done”. Questions for the next meeting es la primera sección donde están las próximas reuniones y las preguntas para estas próximas reuniones que tenga algún integrante para el proyecto. “To do” son actividades próximas que nos llevaran a cumplir las historias de usuario. “1 product backlog” son las historias de usuario a realizar para la primera revisión del proyecto, que están divididas por personas y fechas máximas para realizar ciertas funcionalidades. “Doing” es la pestaña para revisar lo que se está haciendo en ese momento para cada integrante y las funcionalidades en desarrollo. “Done” es la pestaña del tablero que se encarga de llevar registro de lo que se termina y posteriormente valida por el líder del proyecto para posterior entrega.

El tablero completo de Trello en los anexos o en el siguiente enlace: https://trello.com/b/n9Ro7oda/entrega.



Ilustración 5: Ejemplo columnas Trello

# 9 Métodos y herramientas de Estimación

Para el presente proyecto, en las diversas estimaciones del mismo se hizo uso de la planeación ágil, concretamente se llevó a cabo la primera etapa, el release planning, el cual se centra en planear el esfuerzo y duración de las características desde el principio hasta el lanzamiento del sistema, es decir se tiene una perspectiva a largo plazo; para esto se hizo uso del método planning poker concretamente la variante fibonacci, el cual es una práctica ágil para llevar a cabo la estimación en conjunto con todo el equipo, logrando conclusiones en la planificación de manera concreta.

Este método se lleva a cabo como una mesa de discusión, tratando todas las historias de usuario una por una; para cada una de estas se estima el tiempo y esfuerzo de las tareas representadas en valores numéricos por parte de cada uno de los integrantes; cómo se usó la variante fibonacci, estos valores numéricos solo pueden ser parte de la serie fibonacci, esto permite tener en cuenta la incertidumbre y margen de error para tareas grandes; el procedimiento de este método, se llevó a cabo con una herramienta similar a la mostrada en clase, la cual permite tener reuniones de estimación con el método planning poker, con la variante fibonacci <https://www.planningpoker.com/>.[11]

Al llevar a cabo la planificación con dicha herramienta en una reunión con todos los miembros del equipo, se obtuvo los resultados de cada una de las estimaciones de tiempo y esfuerzo para cada funcionalidad del aplicativo expresadas en historias de usuario, proceso que se llevó de manera bastante ágil, llegando a un acuerdo con los integrantes del grupo.

La herramienta utilizada mencionada, nos arrojó los resultados de las estimaciones realizadas para cada historia en un archivo .csv lo que posteriormente se llevó a un archivo de Excel que se adjuntará como el archivo adjunto “TablaResultadosPlanningPoker.xlsx”.

Adicionalmente se tomó como evidencia del proceso llevado a cabo, en un historial de resultados que se adjuntará como el archivo adjunto “ResultadosPlanningPoker.docx”.

# Conclusiones

* Cuando se quiere llevar a cabo un proyecto de desarrollo de software un pilar fundamental para llegar a la culminación de este es la correcta selección del modelo de ciclo de vida a utilizar. Dado que a la larga va a organizar cómo, en qué orden y en qué momento se deben terminar o hacer ciertos hitos que marcarán el correcto avance del proyecto, que posteriormente nos guiarán hacia el éxito de este.
* Una aplicación adecuada y oportuna de las técnicas de gestión de proyecto así mismo como el uso y soporte a través de herramientas permite una correcta gestión en el desarrollo de este. Todos estos conceptos nos permitieron focalizar e implementar oportunamente en las distintas etapas del trabajo para la solución de una plataforma impulsora de videojuegos independientes llegando al final con un demo que nos permite una visualización más clara en la implementación real del código.
* En la planeación de un proyecto de software es importante un buen proceso de estimación del tiempo de las actividades propuestas, bien sean desarrollo de funcionalidades o de otra índole; esto para tener los tiempos bien definidos y en últimas tener un lanzamiento del software acorde con lo definido. Para estos resultados existen diversas técnicas y herramientas de estimación, dependiendo del enfoque del equipo.
* La planeación y gestión inicial de un proyecto agiliza, facilita y hace que el desarrollo del proyecto sea mucho más eficaz en tiempo y calidad, por eso una clara identificación de casos de uso y repartición de trabajo en sprints es clave para el proyecto.
* Un proyecto de software debe ser un proyecto integral que tenga en cuenta diversos estándares tanto de desarrollo como de diseño, de igual manera las estrategias de comunicación entre el equipo son vitales para cumplir a cabalidad los objetivos planteados en dicho proyecto y así se pueden desarrollar las diversas partes indispensables para la finalización de este proyecto que demuestran que la integralidad es clave para obtener los resultados esperados a la hora de incursionar en un proyecto de desarrollo de software.
* La implementación de las metodologías ágiles en el proyecto nos permitió encontrar la forma de organizar el proyecto, y organizarnos como equipo mucho más fácilmente. Las responsabilidades se convirtieron en tareas que podíamos revisar y marcar como realizadas para en cada sprint poder revisar avances y corregir a tiempo cualquier situación que se pudiera presentar.

# Anexos

Todos los anexos irán dentro del ZIP en el que encontró este documento (Principal).

* Los anexos de documentación de historias de usuario y sus respectivos MOCKUPS estarán adjuntos en el ZIP en un documento llamado “Anexos”.
* El demo de la aplicación a base de MOCKUPS en Balsamiq se encuentra dentro del ZIP del proyecto llamado “Entrega 1 Teletubbies”. Ahí encontrará el proyecto de Balsamiq.
* Los anexos de estimaciones por Planning Poker van en el documento “Anexos”.
* El User Story Map (en miro) va adjunto como imagen, con el nombre “UserStoryMap”
* El modelo canvas se adjunta como .jpg con el nombre “canvas”.
* Los resultados del planning Poker en Excel se encuentra como documento .xlsx con el nombre TablaResultadosPlanningPoker.
* Para ver el plan de trabajo puede ver el trello en el siguiente enlace: <https://trello.com/invite/b/n9Ro7oda/6bbea04528da7164f1e7668408f4727d/entrega> o en un anexo donde hay una imagen de este.
* El reporte gerencial se muestra como documento adjunto con el nombre: “Reporte Gerencial”.

# Referencias

[1] Montero, A., 2021. *Modelos de Proceso de Software. Modelos de Ciclos de Vida del Software*.

[2] Montero, A., 2020. *Modelos de ciclo de vida: Desarrollo en cascada*. [video] Available at: <https://www.youtube.com/watch?v=umbW4BZb7SM&ab\_channel=AnabelMontero> [Accessed 7 March 2021].

[3] Montero, A., 2020. *Modelos de ciclo de vida: Desarrollo incremental*. [video] Available at: <https://www.youtube.com/watch?v=0KH0EVehQJ0&ab\_channel=AnabelMontero> [Accessed 7 March 2021].

[4] Zamenfeld, S., Zamenfeld, S. and Zamenfeld, S., 2021. *StarUML una herramienta para modelado*. [online] Brainlabs.com.ar. Available at: <https://www.brainlabs.com.ar/novedad/staruml-una-herramienta-para-modelado/> [Accessed 7 March 2021].

[5] González, G., 2021. *Miro: una plataforma colaborativa para dibujar en pizarras en tiempo real y con videoconferencias*. [online] Genbeta.com. Available at: <https://www.genbeta.com/herramientas/miro-plataforma-colaborativa-para-dibujar-pizarras-tiempo-real-videoconferencias> [Accessed 7 March 2021].

[6] ISDI. 2021. *La herramienta que te permite realizar prototipos de tus proyectos: Balsamiq*. [online] Available at: <https://www.isdi.education/es/isdigital-now/herramienta-te-permite-realizar-prototipos-de-tus-proyectos-balsamiq> [Accessed 7 March 2021].

[7] Es.wikipedia.org. 2021. *Visual Studio Code*. [online] Available at: <https://es.wikipedia.org/wiki/Visual\_Studio\_Code> [Accessed 7 March 2021].

[8] Developer.mozilla.org. 2021. *Conceptos básicos de HTML - Aprende sobre desarrollo web | MDN*. [online] Available at: <https://developer.mozilla.org/es/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basics> [Accessed 7 March 2021].

[9] Es.wikipedia.org. 2021. *Hoja de estilos en cascada*. [online] Available at: <https://es.wikipedia.org/wiki/Hoja\_de\_estilos\_en\_cascada> [Accessed 7 March 2021].

[10] Es.wikipedia.org. 2021. *JavaScript*. [online] Available at: <https://es.wikipedia.org/wiki/JavaScript> [Accessed 7 March 2021].

[11] planningpoker.com. 2021. *Planning poker*. [online] Available at: <https://www.planningpoker.com/about/> [Accessed 8 March 2021].