**FASE 2**

**DISEÑO E IMPLEMENTACIÓN DEL PROYECTO**

**CommitGame**

Escuela de Informática y Telecomunicaciones

Octubre 2025

1. Identificación del Proyecto

| **Nombre de Proyecto** |
| --- |
| CommitGame |

1. Integrantes del Equipo de Trabajo

| **N°** | **Rut** | **Apellidos** | **Nombres** |
| --- | --- | --- | --- |
| 1 | 21.465.787-2 | Palma Vera | Nicolas Gabriel |
| 2 | 21.420.428-2 | Lazcano Ruminot | Francisco José |
| 3 | 21.539.459-K | Lorca Nelson | Antonio Emiliano |
| 4 | 20.627.379-8 | Tepano Roman | Daniel Osvaldo |

1. Registro de Control de Cambios

| **Revisión** | **Fecha** | **Páginas** | **Descripción del Cambio** | **Autor** |
| --- | --- | --- | --- | --- |
| 1 | 14-09-25 | Todas | Cambios según primera revisión docente (alcances y objetivos). | DDI |
| 2 | 15-09-25 | Todas | Avance de guia de estudiante (roles y metodología) | FLA |
| 3 | 29-05-25 | Todas | Cambio a tiempo pasado y tercera persona (No es para entrega ) | FLA |
| 4 | 01-10-25 | Todas | Cambio a tiempo pasado y tercera persona | FLA |

**ÍNDICE DE CONTENIDO**

[LISTA DE TABLAS 5](#_heading=h.lqyrq75arp8)

[LISTA DE FIGURAS 6](#_heading=h.73hpgxt3u4q8)

[LISTA DE DIAGRAMAS 7](#_heading=h.xnyb2oi6ab19)

[GLOSARIO 8](#_heading=h.y0r4eju8ub8l)

[1](#_heading=h.oy86xro82h48) Diseño e Implementación del Proyecto 9

[1.1](#_heading=h.42bydc2155xb) Resumen ejecutivo 9

[1.2](#_heading=h.ui7se3eljddm) Introducción 9

[1.3](#_heading=h.tv2yq0fzc3zz) Problemática a solucionar o necesidad a satisfacer 9

[1.4](#_heading=h.nq03v82u6kn9) Objetivos del Proyecto (general y específicos) 9

[1.5](#_heading=h.p7cr9jhisba4) Competencias del Perfil de Egreso 9

[1.6](#_heading=h.tphks0ub1zyv) Asignación de roles 9

[1.7](#_heading=h.cldxmo8kjgvq) Metodología utilizada en el Proyecto. 10

[1.8](#_heading=h.iz5nqnldungu) Creación de cronograma asociado al Proyecto (Carta Gantt) 10

[1.9](#_heading=h.74xao7lrm5dw) Riesgos Asociados al Proyecto 10

[1.10](#_heading=h.d3xrxte1b45v) Implementación del Proyecto 10

[Diseño y Arquitectura de la solución (Caso de uso de Software o plataforma de gestión) 10](#_heading=h.ut9ff010gtmk)

[Requerimientos técnicos 10](#_heading=h.cpcfy5tiqc8y)

[Desarrollo de la solución 10](#_heading=h.dukv0tjaktjr)

[Resultados de la solución 11](#_heading=h.yytrjzkuwj8l)

[1.11](#_heading=h.8lifwzy67y2r) Definición de Recursos y Costos asociados al Proyecto 11

[CONCLUSIÓN 12](#_heading=h.a2d3b1di7u15)

[BIBLIOGRAFÍA 13](#_heading=h.8r5vy3kheg39)

[ANEXOS 14](#_heading=h.hvumz3mos56n)

# LISTA DE TABLAS

# LISTA DE FIGURAS

# LISTA DE DIAGRAMAS

# GLOSARIO

| **Nombre** | **Descripción** |
| --- | --- |
| **Capstone** | Asignatura final de la carrera que representa la culminación del proceso de formación, en la cual el estudiante integra sus conocimientos y competencias en un proyecto aplicado. |
| **Daily meeting** | Es una reunión breve y diaria , generalmente de 15 minutos , en la que el equipo de trabajo se reunió para compartir avances . |
| **Gamificación** | Es la aplicación de mecánicas, dinámicas y elementos propios de los juegos en contextos o áreas no lúdicas, con el objetivo de motivar, facilitar el aprendizaje, mejorar la participación o fomentar determinados comportamientos, por ejemplo, la educación. |
| **Jira** | Herramienta de gestión de proyectos desarrollada por Atlassian, utilizada principalmente para planificar, rastrear y gestionar proyectos de software bajo metodologías ágiles como Scrum . |
| **Scrum** | Marco de trabajo ágil para la gestión de proyectos que se centra en la entrega incremental de valor. |
| **Scrum Master** | Facilitará al equipo Scrum que asegure la correcta aplicación de la metodología ágil, elimina obstáculos, promueve la colaboración y protege al equipo para que pueda trabajar de forma eficiente . |
| **Sprint Planning** | Es una reunión clave que se lleva a cabo en cada sprint en scrum . Su propósito es planificar las tareas y objetivos que se deben alcanzar en el sprint . |
| **Unity (Development platform )** | Será plataforma de creación de contenido en tiempo real que proporciona a los usuarios las herramientas para crear juegos . |
| **Product Owner** | Es un rol fundamental dentro de metodologías ágiles representa la voz del cliente o usuario final dentro del equipo de desarrollo . |
| **Product Backlog** | Es una lista priorizada de tareas y características que deben implementarse para completar el juego ,esto incluye elementos como mecánicas ,contenido , errores etc . |
| **Trello** | Aplicación de gestión visual de proyectos basada en tableros, listas y tarjetas. Permite organizar tareas de manera colaborativa, asignar responsables, establecer fechas límite y dar seguimiento al progreso de un proyecto. |

# Diseño e Implementación del Proyecto

## Resumen

Este proyecto consistió en el desarrollo de un videojuego 2D de puzzles y plataformas en Unity, diseñado como una herramienta educativa para facilitar el aprendizaje de programación básica de manera dinámica, interactiva y divertida. Actualmente, existe un gran interés por parte de jóvenes en aprender programación; sin embargo, muchas veces se enfrentan a conceptos complejos y abstractos que resultan poco intuitivos, lo que genera frustración y, en algunos casos, abandono de sus estudios. Este proyecto buscó abordar esta problemática ofreciendo un entorno accesible y motivador para adquirir habilidades de programación de forma gradual.

El videojuego está pensado principalmente para jóvenes de entre 14 y 25 años, aunque puede ser utilizado por cualquier persona interesada en aprender o reforzar sus conocimientos de programación. La plataforma permite escoger el nivel de habilidad del usuario, ya sea principiante, intermedio o avanzado, ofreciendo desafíos y ejercicios acordes a su experiencia. De esta manera, se promueve un aprendizaje personalizado que permite a cada jugador avanzar a su propio ritmo.

Entre las principales funcionalidades del proyecto se incluye la práctica de conceptos de programación y pseudocódigo a través de retos dentro del juego, así como un minijuego de mecanografía que ayuda a mejorar la escritura correcta y la velocidad de tipeo, habilidades complementarias útiles en el estudio de la programación. La combinación de puzzles, plataformas y ejercicios de ingenio busca mantener al usuario motivado y comprometido, haciendo que el aprendizaje sea un proceso más ameno y eficaz.

El desarrollo del videojuego se realizó en varias etapas. La primera es la planificación y diseño, donde se definieron objetivos, mecánicas, niveles, etc. La segunda etapa fué el desarrollo de mecánicas y programación de niveles, implementando movimientos, obstáculos y desafíos de forma progresiva.

La tercera etapa consistió en la integración del minijuego de mecanografía, que refuerza la escritura y velocidad de tipeo al que se accede a través del menú principal. La cuarta etapa fue testing, en la que se evaluó el funcionamiento, se corrigieron errores y se mejoró la experiencia del usuario.

Finalmente, se procedió al lanzamiento y evaluación final, recopilando comentarios de los usuarios y verificando que el proyecto cumpla sus objetivos educativos y de entretenimiento.

En síntesis, este proyecto representa una solución innovadora para introducir a los jóvenes al mundo de la programación de manera entretenida y accesible. Al integrar enseñanza, práctica y gamificación, se generó un impacto positivo en el aprendizaje y la retención de conocimientos, contribuyendo al desarrollo de habilidades técnicas y cognitivas que serán útiles tanto en su formación académica como en su vida profesional futura.

## Abstrac

Mismo resumen ejecutivo de una plana con el resumen del proyecto indicando los principales requerimientos y etapas del mismo (En inglés).

This project is the development of a 2D puzzle-platformer video game in **Unity**, designed as an educational tool to make learning to code dynamic, interactive, and engaging. There is currently significant interest among young people in learning to code; however, they often face complex and abstract concepts that are not intuitive, which leads to frustration and, in some cases, giving up on their studies. Our project seeks to address this problem by offering an accessible and motivating environment for gradually acquiring programming skills.

The video game is primarily intended for young people aged 14 to 25, although it can be used by anyone interested in learning or reinforcing their programming knowledge. The platform adapts to the user's skill level—beginner, intermediate, or advanced—offering challenges and exercises appropriate to their experience. In this way, it promotes personalized learning that allows each player to progress at their own pace.

The project's main features include practicing programming and pseudocode concepts through in-game challenges, as well as a typing minigame that helps improve correct writing and typing speed, useful complementary skills in the study of programming. The combination of puzzles, platforms, and problem-solving exercises seeks to keep the user motivated and engaged, making learning a more enjoyable and effective process.

The development of the video game will be carried out in several stages. The first stage is planning and design, where objectives, mechanics, levels, and other key elements will be defined. The second stage is development of mechanics and level programming, implementing movements, obstacles, and challenges progressively.

The third stage involves the integration of the typing minigame, which reinforces writing accuracy and typing speed and can be accessed through the main menu. The fourth stage is testing, during which the game's functionality will be evaluated, errors corrected, and the user experience improved.

Finally, the launch and final evaluation will take place, collecting user feedback and ensuring that the project meets its educational and entertainment objectives.

In summary, this project represents an innovative solution to introduce young people to the world of programming in an engaging and accessible manner. By integrating teaching, practice, and gamification, it is expected to generate a positive impact on learning and knowledge retention, contributing to the development of technical and cognitive skills that will be useful both in academic training and in future professional life.

## Introducción

En el proyecto de **Capstone** de la carrera de Ingeniería Informática, se aplicaron las competencias de egreso adquiridas durante la carrera para abordar una problemática común en el ámbito de la programación. La problemática identificada fue la dificultad que enfrentan muchos jóvenes al intentar aprender los conceptos básicos de programación, los cuales suelen ser abstractos y difíciles de comprender sin una experiencia práctica y directa. Para resolver esto, se diseñó una solución tecnológica que facilitara el aprendizaje mediante un enfoque más accesible y entretenido.

Se eligió una metodología ágil para el desarrollo del proyecto, específicamente **Scrum**, lo que permitió una organización flexible y colaborativa del trabajo. El uso de Scrum facilitó la asignación de actividades y funcionalidades a diferentes integrantes, así como la definición de roles específicos. Para apoyar su aplicación se emplearon herramientas externas como **Trello** y **Jira**, que ayudaron en la organización de tareas y el seguimiento de los avances. Esta metodología favoreció la adaptación continua a los cambios y facilitó la integración de nuevos enfoques o ajustes de acuerdo con las necesidades del proyecto. Asimismo, se prioriza la retroalimentación constante, tanto del equipo de desarrollo como de los usuarios, con el fin de asegurar que el producto final cumpliera con los objetivos educativos.

El proyecto, denominado **CommitGame**, consistió en el diseño y desarrollo de un videojuego educativo 2D, dirigido principalmente a jóvenes entre 14 y 25 años interesados en aprender los fundamentos de la programación. A través de esta plataforma, se buscó introducir conceptos de programación de una forma divertida y dinámica, superando la barrera de complejidad que muchos encuentran en los métodos tradicionales de enseñanza. El videojuego proporcionó una serie de retos y ejercicios interactivos que permitieron a los jugadores practicar y comprender mejor los conceptos clave de la programación, como variables, bucles y condicionales.

La estructura del juego incluyó diferentes niveles de dificultad, lo que permitió adaptar el contenido a las habilidades de cada usuario, desde principiantes hasta usuarios con algo de experiencia en programación. Además, se integraron mecánicas de gamificación, como recompensas y puntos, para mantener la motivación y el compromiso de los jugadores a lo largo del proceso de aprendizaje.  
  
En este contexto ,el proyecto **CommitGame** se planteó como una propuesta innovadora para apoyar la enseñanza de la programación , combinando lo lúdico con lo educativo . Su diseño buscó acercar a los jóvenes al mundo de la informática mediante una herramienta práctica que complementará los métodos tradicionales de enseñanza , ofreciendo una experiencia interactiva , motivadora y efectiva para el desarrollo de habilidades en programación

## 

## Problemática a solucionar o necesidad a satisfacer

En este punto se debe describir todas las características de la problemática, los antecedentes y todo tipo de evidencia que es necesaria para evidenciar el problema o la necesidad de cómo se origina el proyecto.

Para la elaboración y desarrollo del proyecto que se propuso a realizar en primer lugar, se enfocó en solucionar diversas problemáticas, como lo son que varios estudiantes se han atrasado con las asignaturas de su carrera por dificultades en materias de programación, también que los estudiantes nuevos que inician su proceso de estudio se les dificulta comprender los distintos conceptos y realizar las prácticas que se solicitan, y por último que las personas que quieren ingresar al mundo de la programación por mero gusto, también se les dificulta entender y seguir el hilo de lo que la programación conlleva.

A través de esta solución los estudiantes pueden aprender de manera más asequible, interactiva y dinámica, para que el aprendizaje de programación sea llevadero y sencillo, ya que se evidenció los problemas anteriormente mencionados por pares y docentes, así que se consideró que existía una necesidad alta por un medio que pudiera ayudar en el aprendizaje continuo y adaptado a las capacidades de cada uno, por lo que, el proyecto puede hacer que tanto los estudiantes nuevos como los que se están retrasando con asignaturas o incluso personas que quieran ingresar a la programación, les sea más fácil y más comprensiva la manera de aprender.

## Objetivos del Proyecto (general y específicos)

Se debe plantear el **objetivo general** del proyecto. Es importante recordar que el objetivo general responde a la pregunta sobre lo que el proyecto intentará alcanzar. En el fondo es el ¿Para qué?

Para los **objetivos específicos**, se deben plantear cada uno de los objetivos específicos asociados al proyecto. Es importante recordar que cada objetivo específico se encuentra asociado a lo concreto del proyecto en un determinado periodo de tiempo y es medible. En el fondo es ¿el cómo? Se deben definir por lo menos tres objetivos específicos asociados al proyecto (ya presentados en la Fase 1).

**Objetivo general:**

Diseñar y crear un videojuego para jóvenes entre 14 y 25 años con el fin de que aprendan las nociones básicas de la programación mediante la **gamificación.** Desarrollando una estructura didáctica, dinámica y comprensiva con distintos entornos de juego, permitiendo con esto reforzar los conocimientos previos adquiridos en las asignaturas como programación de algoritmos y desarrollo de software de escritorio.

**Objetivos específicos:**

* Promover e incentivar la programación a través de la **gamificación**.
* Reforzar conceptos básicos de la programación vistos en las asignaturas.
* Conectar el aprendizaje teórico con el práctico.
* Desarrollar el pensamiento lógico y crítico a través de la resolución de puzzles.

## Competencias del Perfil de Egreso

En este apartado debe insertar todas las competencias técnicas asociadas a su perfil de egreso y que se encuentran directamente relacionada al desarrollo del proyecto.

Debe insertar un mínimo de tres competencias

Las competencias técnicas asociadas al perfil de egreso son un conjunto de conocimientos, habilidades, actitudes y valores que un estudiante posee y es capaz de demostrar al finalizar su carrera, estas son las que que están relacionadas con el desarrollo del proyecto:

* Desarrollo de software: Construir programas y rutinas aplicando buenas prácticas de codificación.
* Gestión de proyectos: Administrar proyectos informáticos y apoyar la toma de decisiones.
* Análisis y soluciones: Proponer soluciones informáticas alineadas a los procesos de la organización.
* Seguridad informática: Identificar y resolver vulnerabilidades para cumplir normas de seguridad.
* Modelado de datos: Diseñar y construir modelos de datos escalables y acordes a los requerimientos.

## Asignación de roles

Cada integrante del grupo debe tener asignado un rol dentro del proyecto en donde se debe describir las funciones y tareas que ejecutarán.

La asignación de roles se realizó en la siguiente tabla con 3 distintos parámetros, empezando por el rol el cual cumplirá el integrante del grupo y se mantuvo así hasta el final del proyecto, por otro lado se encuentra el parámetro del responsable correspondiente a quien es la figura responsable de hacer valer el rol asignado y por último está la funcionalidad que en su totalidad explicará cuales son las tareas que llevó a cabo cada integrante con el rol asignado.

| **Rol** | **Responsable** | **Funcionalidad** |
| --- | --- | --- |
| Product Owner | Daniel Tepano | Priorizar funcionalidades  Decidir si se desarrolla primero: Diseño de niveles, efectos visuales/sonoros, integración de niveles y buscar equilibrio entre jugabilidad y aspectos técnicos .  Validación de cada sprints  ¿Los puzzles son jugables?  ¿La curva de dificultad está bien ajustada? |
| Scrum Master | Francisco Lazcano | Facilita las reuniones diarias para que los miembros del equipo enseñen sus avances, también ayuda la retroalimentación del equipo para identificar qué funcionó, qué no y que se puede mejorar |
| Equipo de desarrollo | Daniel Tepano, Francisco Lazcano, Antonio Lorca, Nicolás Palma | Serán responsables de escribir el código del videojuego, implementar las mecánicas de juego, optimizar rendimiento y corregir errores. |

#### 

## Metodología utilizada en el Proyecto.

En este apartado se debe mencionar la metodología de proyecto que se utilizará durante el desarrollo del mismo, mencionando todas las fases y detallar cuál es la función de cada una de las fases (según metodología ágil o tradicional).

Para el desarrollo del proyecto se utilizó una metodología ágil, específicamente **Scrum**, por ser una de las más adecuadas en proyectos de este tipo. Esta metodología permitió mantener un orden estructurado, un trabajo colaborativo y flexible, además de fomentar la interacción constante entre los integrantes del equipo. El uso de **Scrum** facilitó la asignación de actividades y funcionalidades a diferentes integrantes, así como la definición de roles específicos. Para apoyar su aplicación se emplearon herramientas externas como **Trello** y **Jira**, que ayudaron en la organización de tareas y el seguimiento de los avances.

En cuanto a la elección metodológica, no se aplicaron enfoques tradicionales como el modelo en cascada, ya que estos suelen ser rígidos y lineales, dificultando la adaptación a cambios en los requerimientos. Del mismo modo, tampoco se optó por modelos híbridos, debido a que podían generar una mayor carga de gestión administrativa en un proyecto de estas dimensiones.

En cambio, **Scrum** ofreció una mayor adaptabilidad, entregas parciales en menor tiempo y una comunicación continua entre los miembros del equipo, lo que resultó fundamental para responder a posibles imprevistos.

La metodología ágil aplicada estuvo compuesta por diferentes fases, cada una con un propósito particular:

* **Planteamiento:** en esta fase se definieron los objetivos principales del proyecto y se identificaron las necesidades iniciales.
* **Requerimientos priorizados:** se elaboró un listado de requisitos organizados en función de su importancia y valor para el proyecto, lo que permitió planificar de manera más eficiente.
* **Iteración:** se llevaron a cabo ciclos de trabajo denominados sprints, en los cuales se desarrollaron y entregaron funcionalidades específicas en periodos cortos de tiempo.
* **Puesta en marcha**: en esta etapa se realizaron las pruebas, ajustes y validaciones necesarias para garantizar el correcto funcionamiento del producto desarrollado.

Asimismo, todo el equipo participó activamente en el proceso, tanto en la planificación como en la mejora continua. Scrum contempló distintos niveles de planificación: el estratégico, representado por el **Product Backlog**; el táctico, correspondiente al **Sprint Planning**; y el operativo, implementado en las **Daily Meetings**. Esta estructura permitió mantener un control constante sobre el progreso del proyecto y asegurar la adaptación a los cambios que surgieron durante su desarrollo.

En conclusión, la metodología ágil con enfoque en **Scrum** resultó ser un recurso clave para la gestión del proyecto, ya que garantiza un desarrollo ordenado, participativo y adaptable a las necesidades que fueron apareciendo en cada etapa.

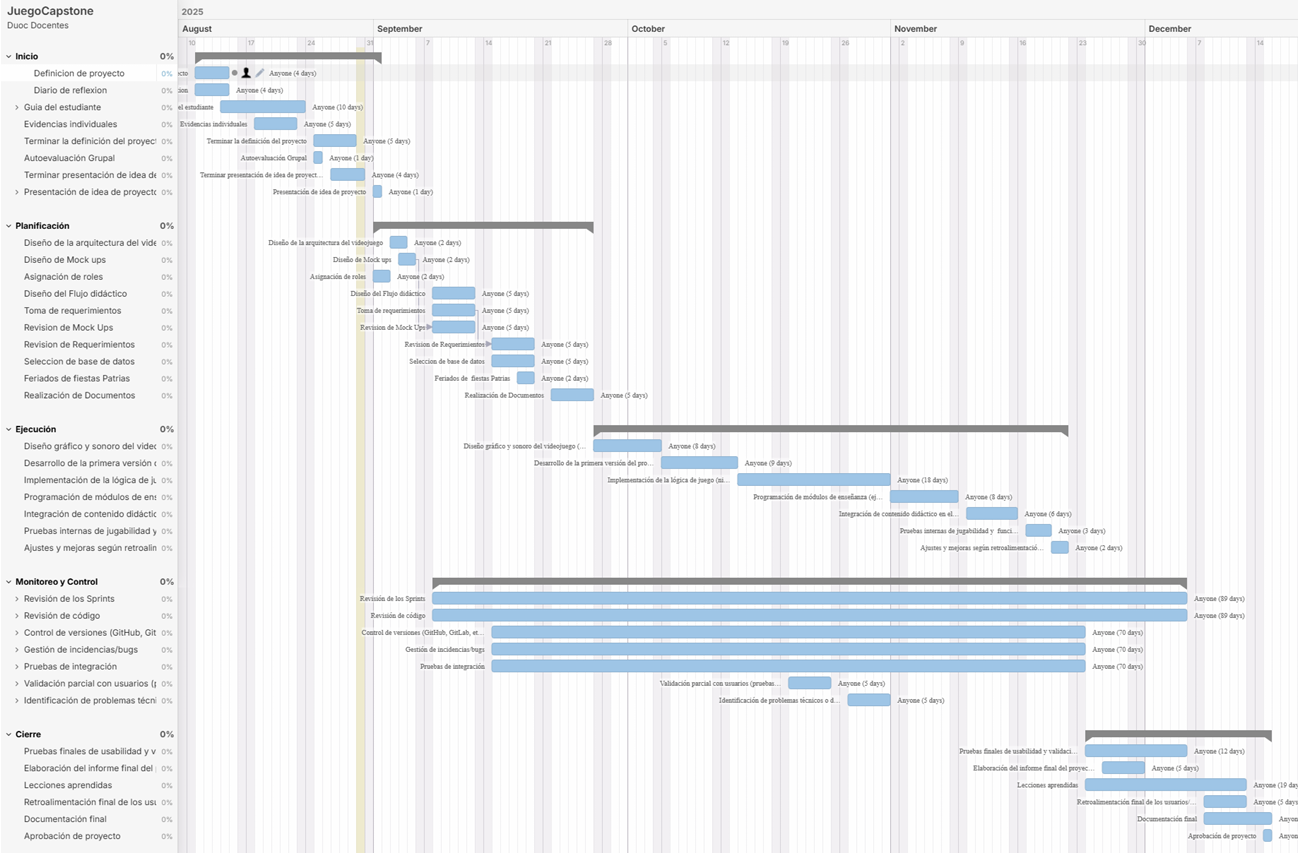
## Creación de cronograma asociado al Proyecto (Carta Gantt)

Debe insertar la carta Gantt del proyecto donde se debe evidenciar las principales etapas, recursos asociados a cada tarea y visualizar las fechas de cada una de ellas.

En la siguiente imagen se hace referencia a la carta Gantt del proyecto, se pudo evidenciar el desarrollo completo del proyecto a través de 5 fases que ayudó a llevar a cabo la solución, los cuales son, el inicio donde se evidencia la idea principal del proyecto junto con la propuesta de solución y la problemática la cual se resolvió, para la planificación dejamos en claro el diseño, las funciones y la asignación de los roles que cumplirá cada uno de los integrantes del equipo.

En ejecución es donde se a desarrollar el proyecto, para ello se desarrollarán los niveles, la banda sonora, el diseño gráfico del personaje en 2D, una implementación de la lógica de juego y pruebas internas de jugabilidad, por la parte de monitoreo y control nos encargaremos primordialmente de hacer revisiones periódicas que conllevarán el hacer pruebas de integración, validaciones por parte de los usuarios, revisión de código y revisión de los sprints.

Por último pero no menos importante el cierre en el cual realizaremos la pruebas finales de usabilidad para los usuarios, haremos una retroalimentación de todo lo aprendido a lo largo del desarrollo de nuestro proyecto y su futura evaluación por parte del comité.



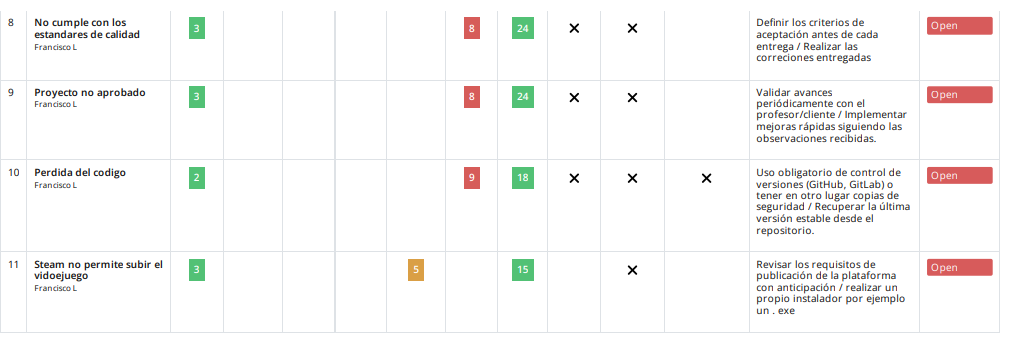
## 

## Riesgos Asociados al Proyecto

En este apartado se debe agregar la matriz de riesgos con la descripción de riesgos, probabilidad de ocurrencia, impacto, plan de mitigación y plan de contingencia, y por supuesto su métrica respecto a cómo se establece la criticidad e impacto.

En está imagen se demostró un registro de los riesgos encontrados a lo largo del desarrollo del proyecto, para realizar esté proceso se llegó a utilizar la herramienta de Teamwork, donde se desarrolló los riesgos encontrados dentro del proyecto.





## Implementación del Proyecto

### Diseño y Arquitectura de la solución (Caso de uso de Software o plataforma de gestión)

* Describir en profundidad el modelo de datos que permiten soportar los requerimientos de la organización de acuerdo al diseño definido que sea escalable en el tiempo.

### Requerimientos técnicos

* Detallar cada uno de los dispositivos, insumos y herramientas que se utilizaran para la implementación del proyecto.

### Desarrollo de la solución

* Se recomienda insertar capturas de pantalla con una breve explicación de cada uno de los pasos realizados en cada proceso de programación.
* Evidenciar la solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos del proyecto.
* Integrar los distintos componentes de la solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos del proyecto.

### Resultados de la solución

* En este apartado debe insertar todas las pruebas que fueron ejecutadas para chequear los resultados de la solución. Puede incorporar capturas de pantalla, videos o demostración en tiempo real de la solución.
* Los resultados deben ser mostrados en la exposición final frente a la comisión.

## Definición de Recursos y Costos asociados al Proyecto

# CONCLUSIÓN

Sintaxis final del documento que permite resumir y/o llegar a conclusiones desde la detección del problema o necesidad, pasando por el desarrollo del proyecto hasta las pruebas finales de funcionamiento.

# BIBLIOGRAFÍA

Para la bibliografía considerar Norma APA 7ma Edición.

# 

# ANEXOS

Debe incorporar como anexos en documento como archivo Word, todo tipo de información que complemente el proyecto.

En la entrega final del proyecto, deben enviar todos los archivos utilizados en la implementación y desarrollo del mismo. El docente mencionará unas clases antes de la entrega final dicho listado a entregar.