**Asignatura:** Taller de Desarrollo de Aplicaciones

**Código y Sección:** TIHI31/D-B50-N3-P12-C1/D

**Nombre del profesor:** Javier Eduardo Ahumada Cortes

**Nombre del equipo**: Los Improvisados

**Nombre de los integrantes del grupo:**

-Patrick Enrique Santibañez Meza

-Iris Muñoz Hinestroza

Prototipo de Juego de Rol

Contenido

[I Introducción. 3](#_Toc196942569)

[II Instrumentos para el Levantamiento de Requerimientos 3](#_Toc196942570)

[2.1 Analisis de Documentacion. 3](#_Toc196942571)

[2.2 Prototipado. 4](#_Toc196942572)

[III Historia de Usuario. 4](#_Toc196942573)

[3.1 Epicas. 4](#_Toc196942574)

[3.2 Levantamientos de los Requerimientos. 4](#_Toc196942575)

[IV Objetivo de Servicio 6](#_Toc196942576)

[Post-Implementación 9](#_Toc196942577)

[Seguridad y Control de Acceso 10](#_Toc196942578)

[V Casos de Uso 11](#_Toc196942579)

[5.1 Diagrama Gestion de Sistema de Usuarios. 11](#_Toc196942580)

[5.2 Diagrama Gestion de Personajes Jugador. 12](#_Toc196942581)

[5.3 Diagrama Gestion Para GM. 13](#_Toc196942582)

[5.4 Diagrama Creación de Personajes. 14](#_Toc196942583)

[VI Referencias bibliográficas. 14](#_Toc196942584)

[4.1 Instrumentos para el Levantamiento de Requerimientos. 14](#_Toc196942585)

# I Introducción.

Esta iniciativa responde a la necesidad de modernizar sus productos y ofrecer una plataforma centralizada para la creación y gestión de personajes, habilidades, poderes y estados de juego.

El desarrollo de esta aplicación se ha planificado utilizando la metodología ágil Scrum, lo que permite una entrega incremental del producto mediante iteraciones denominadas *sprints*. Cada funcionalidad requerida se ha descompuesto en épicas e historias de usuario, priorizadas y organizadas en un product backlog alojado en Azure DevOps, herramienta que facilitará la gestión colaborativa del proyecto.

El objetivo principal de este prototipo es establecer una base funcional sólida que permita a los jugadores crear y personalizar personajes con características, mientras que el GM podrá gestionar el equilibrio del juego respetando reglas definidas, tales como la imposibilidad de modificar elementos iniciales de los personajes creados por los jugadores.

El presente informe documenta el prototipo inicial, las historias de usuario consideradas, las decisiones de diseño, las restricciones del sistema y las funcionalidades implementadas, sirviendo como base para futuras ampliaciones y validaciones con los usuarios finales.

Hardware minimo: Para trabajar con Python, Pygame, MongoDB, Visual Studio y GitHub de forma fluida durante el desarrollo de tu juego de rol

-Procesador: Intel Core i3

-Ram: 8GB ddr4

-Almacenamiento: 256 gb ssd o hdd

-Tarjeta Grafica: Integral Intel HD 4000 o superior

-Sistema Operativo: Windows 10/11

-Pantalla: 1366x768 de resolución

Software:

-Python: Para lógica y funciones del Juego.

-Pygame: Biblioteca para el desarrollo 2D

-MongoDB: Base de datos no Relacionales

-Visual Studio: Es un editor de código liviano, multiplataforma y altamente personalizable

-GitHub: Permite gestionar el código fuente de forma segura, trabajar en equipo mediante ramas

Metodología SCRUM:

Se utilizará la metodología ágil SCRUM para la gestión del desarrollo del sistema. El trabajo se organizará en sprints, donde se planificarán, desarrollarán y evaluarán entregables funcionales del prototipo.

Cada funcionalidad estará descrita mediante historias de usuario, las cuales representarán los requerimientos tanto del Jugador como del Game Master (GM).

# II Instrumentos para el Levantamiento de Requerimientos

## 2.1 Analisis de Documentacion.



Analizamos Los Requerimientos pedidos en Prototipo de Juego de Rol pagina 4

Investigación sobre que son los Juegos RPG(Rol)(Bibliografía)

## 2.2 Prototipado.

Crear Prototipo de Juego de Rol con los requerimientos pedidos

# III Historia de Usuario.

## 3.1 Epicas.

1. **Epica**: Sistema de Autentificacion
   * 1. Historia de usuario: Como Jugador o GM necesito poder iniciar sesión con un usuario y contraseña
     2. Historia de usuario: El sistema debe verificar entre Jugador o GM
2. **Epica**: Creacion de Personajes
   * 1. Historia de Usuario: Como jugador quiero crear un personaje con nombre, raza, nivel y estados.
     2. Historia de Usuario: Como jugador quiero agregar habilidades iniciales a mi personaje
     3. Historia de Usuario: Como jugador quiero agregar equipamiento inicial a mi personaje
     4. Historia de Usuario: Como jugador quiero agregar poderes iniciales a mi personaje
3. **Epica**: Menu de personaje de Jugador
   * 1. Historia de Usuario: Como Jugador quiero ver el estado de mi personaje
     2. Historia de Usuario: Como jugador quiero modificar el equipamento de mi personaje siempre que este Vivo.
4. **Epica**: Funciones del GM(Game Master)
   * 1. Historia de Usuario: Como GM quiero modificar las fichas de los personajes
     2. Historia de Usuario: Como GM quiero ver una lista de todos los personajes
     3. Historia de Usuario: Como GM quiero agregar o editar habilidades y poderes a las razas
     4. Historia de Usuario: Como GM quiero agregar nuevas razas, equipamientos y estados
5. **Epica**: Gestion de Listas
   * 1. Historia de Usuario: Como GM quiero ver listas de habilidades separadas por razas
     2. Historia de Usuario: Como GM quiero ver listas de poderes separadas por razas
     3. Historia de Usuario: Como GM quiero ver listas de habilidades separadas por razas

## 3.2 Levantamientos de los Requerimientos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RECOPILACIÓN DE HISTORIAS DE USUARIO** | | | | |
| **ID** | **Rol** | **Característica / Funcionalidad** | **Razón / Resultado** | **Condiciones** |
| 01 | Como Jugador | Necesito iniciar sesión con un usuario y contraseña | Con la finalidad de acceder al sistema y administrar mis personajes | Debe permitir autenticación con credenciales únicas |
| 02 | Como Jugador o GM diferentes | Necesito que el sistema identifique mi rol (Jugador o GM) al iniciar sesión | Para acceder a las funciones correspondientes a mi rol | Un jugador no puede ser GM al mismo tiempo |
| 03 | Como Jugador | Necesito crear un personaje con nombre, raza, nivel y estado | Para poder jugar dentro del sistema | El personaje debe iniciar en nivel 1 y estado vivo |
| 04 | Como Jugador | Necesito asignar habilidades iniciales a mi personaje | Para que tenga características especiales desde el inicio | Debe permitir ingresar solo 2 habilidades al comienzo |
| 05 | Como Jugador | Necesito agregar equipamiento inicial a mi personaje | Para que pueda usarlo en el juego | Debe permitir solo 1 equipamiento al inicio |
| 06 | Como Jugador | Necesito agregar un poder inicial a mi personaje | Para que pueda utilizar habilidades especiales en el juego | Solo se permite 1 poder al inicio |
| 07 | Como Jugador | Necesito ver el estado de mi personaje | Para conocer su nivel, equipo y estado actual | Debe mostrar nombre, raza, nivel y estado |
| 08 | Como Jugador | Necesito modificar el equipamiento de mi personaje | Para mejorar sus capacidades durante el juego | Solo es posible si el personaje está vivo |
| 09 | Como Jugador | Necesito ver un resumen de mis personajes creados | Para hacer seguimiento de su progreso | Debe mostrar nombre, raza, nivel y estado |
| 10 | Como GM | Necesito modificar las fichas de los personajes | Para administrar y equilibrar el juego | No puedo eliminar las habilidades, equipamiento o poder inicial del jugador |
| 11 | Como GM | Necesito ver una lista de todos los personajes de la partida | Para gestionar y controlar el estado de los jugadores | Debe mostrar nombre, raza, nivel y estado de cada personaje |
| 12 | Como GM | Necesito agregar o editar habilidades y poderes a las razas | Para equilibrar el juego y permitir crecimiento de personajes | Solo puedo editar los que yo haya agregado |
| 13 | Como GM | Necesito agregar nuevas razas, equipamientos y estados | Para ampliar las opciones en el juego | No puedo modificar las razas base ni los estados base (Vivo y Muerto) |
| 14 | Como GM | Necesito ver listas de habilidades y poderes separadas por razas | Para asegurar que cada raza tenga habilidades y poderes únicos | Solo puedo asignar poderes y habilidades a las razas a las que pertenecen |
| 15 | Como GM | Necesito generar informes de personajes | Para analizar el estado del juego y sus jugadores | Debe permitir exportar la información en formato visual y/o descargable |
| 16 | Como GM | Necesito iniciar sesión con un usuario y contraseña | Con la finalidad de gestionar el sistema y los personajes | Debe permitir autenticación con credenciales únicas |

# IV Objetivo de Servicio

**Datos del Cliente**

|  |  |
| --- | --- |
| **Nombre** | Infinity Creations |
| **Rut** | 76.098.111-1 |
| **Teléfono** | +569 63191046 |
| **Email** | infinitycreations@corp.com |

**Datos de los Consultores técnicos** (Coloque aquí la identificación de los integrantes del grupo)

|  |  |
| --- | --- |
| **Nombre** | Patrick Enrique Santibañez Meza |
| **Rut** | 20.015.532-7 |
| **Nombre** | Iris Muñoz Hinestroza |
| **Rut** | 25.839.212-4 |
| **Link Azure** | https://dev.azure.com/patricksantibanez1/Juego%20de%20Rol |

**Observaciones generales**

|  |
| --- |
| El principal objetivo del proyecto es desarrollar un prototipo funcional de una aplicación que permita la creación y gestión centralizada de personajes, habilidades, poderes y estados dentro de un juego de rol. Esta herramienta busca facilitar tanto la personalización por parte de los jugadores como el control y equilibrio del juego por parte del Game Master (GM), respetando reglas predefinidas.  **Alcance del Proyecto**  El proyecto incluye:   * La implementación de una plataforma basada en Python y Pygame para el entorno gráfico. * La gestión de datos mediante MongoDB para almacenar información de personajes y reglas del juego. * El uso de Visual Studio y GitHub como herramientas de desarrollo colaborativo. * La aplicación de la metodología ágil Scrum para planificar y entregar funcionalidades de manera incremental mediante sprints. * La documentación de historias de usuario, decisiones de diseño, restricciones técnicas y funcionalidades implementadas, con miras a futuras mejoras y validaciones con usuarios finales. |

**Descripción del Servicio:**

|  |
| --- |
| **Para efectuar el desarrollo de la aplicación se requiere del siguiente entorno de trabajo:**  1.- Describa la infraestructura necesaria para la implementación de la aplicación: Oficinas y Espacios Físicos  * **Oficina central o coworking con al menos 2 espacios diferenciados**:   + Área de desarrollo (trabajo técnico)   + Sala de reuniones para planificación de sprints, revisiones y retrospectivas SCRUM * Acceso a internet de alta velocidad (mínimo 300 Mbps) * Seguridad física (llaves, cámaras o acceso con tarjetas/RFID) * Aire acondicionado y ventilación adecuados  **Computadoras de Desarrollo**  * **Cantidad recomendada**: Al menos 4 a 6 equipos para un equipo de desarrollo pequeño (Scrum Master, Product Owner, 2-3 desarrolladores, 1 tester si aplica). * **Especificaciones mínimas por equipo (para desarrollo fluido en Python/Pygame/MongoDB):**   + CPU: Intel i3 (10ma gen en adelante) o AMD Ryzen 3   + RAM: 8GB DDR4   + Almacenamiento: SSD 256 GB mínimo   + GPU: Integrada o dedicada básica (Integral Intel HD 4000 o superior)   + Resolución: 1366x768 (para buena visualización de editores y herramientas de depuración)  **Servidores y Servicios en la Nube**  * **Base de datos y backend**:   + MongoDB Atlas (MongoDB en la nube gestionada, escalable) * **Repositorio de código y colaboración**:   + GitHub (uso de repositorios privados, integración con Azure DevOps) * **Gestión del proyecto**:   + Azure DevOps (para product backlog, boards SCRUM, sprints y tareas) * **Despliegue de prototipos web (si se requiere)**:   + Render, Heroku o Vercel para entornos de prueba/demostración * **Almacenamiento de archivos compartidos**:   + Google Drive o OneDrive corporativo * **Servidor local (opcional)**:   + Un mini servidor (Intel NUC o similar) con Linux o Windows Server para pruebas locales si no se usa 100% cloud  **Otros Servicios Complementarios**  * **Correo corporativo (por ejemplo, Google Workspace o Microsoft 365)** * **Firewall y antivirus para protección básica** * **VPN (si se trabaja remotamente para acceso seguro al entorno de desarrollo)**  Escalabilidad futura Si el proyecto crece:   * Considera servicios como AWS, Azure o Google Cloud para escalado automático. * Añadir una herramienta de CI/CD (como GitHub Actions o Jenkins). * Repositorio de contenedores (Docker Hub) y uso de Docker en servidores si se desea portabilidad y despliegue rápido.   2.- Describa la configuración necesaria del hardware para que funcione la aplicación:  Hardware minimo: Para trabajar con Python, Pygame, MongoDB, Visual Studio y GitHub de forma fluida durante el desarrollo de tu juego de rol  -Procesador: Intel Core i3  -Ram: 8GB ddr4  -Almacenamiento: 256 gb ssd o hdd  -Tarjeta Grafica: Integral Intel HD 4000 o superior  -Sistema Operativo: Windows 10/11  3.- Describa el software necesario para el desarrollo, y posterior implementación de la aplicación:  Software:  -Python: Para lógica y funciones del Juego.  -Pygame: Biblioteca para el desarrollo 2D  -MongoDB: Base de datos no Relacionales  -Visual Studio: Es un editor de código liviano, multiplataforma y altamente personalizable  -GitHub: Permite gestionar el código fuente de forma segura, trabajar en equipo mediante ramas **Post-Implementación****Mantenimiento Correctivo**  * Monitorear errores en tiempo real (usando herramientas como **Sentry** o **LogRocket**). * Revisar logs del sistema (idealmente almacenados en un servicio como AWS CloudWatch o en archivos rotativos). * Atender reportes de bugs de los usuarios y aplicar parches.  **Mantenimiento Preventivo**  * Actualizar dependencias de Python, Pygame, MongoDB y otros paquetes regularmente con herramientas como pip-tools o Dependabot. * Respaldos automáticos de la base de datos (frecuencia diaria o semanal). * Pruebas automatizadas periódicas (unitarias y de integración) para prevenir errores tras cambios.  **Mantenimiento Evolutivo**  * Añadir nuevas funcionalidades a partir de feedback de los jugadores o el Game Master (GM). * Ajustar balance de juego (por ejemplo, habilidades o parámetros de personajes). * Rediseñar o refactorizar componentes para mejorar rendimiento o escalabilidad.  **Seguridad y Control de Acceso**  * **Backups automáticos** (diarios en MongoDB Atlas o S3) * **Autenticación de usuarios** usando JWT o Firebase Auth * **HTTPS** con Let's Encrypt o certificados en AWS * **Revisión de código continua** (pull requests en GitHub, con CI) |

# V Casos de Uso

## 5.1 Diagrama Gestion de Sistema de Usuarios.

Diagrama

El contenido generado por IA puede ser incorrecto.

## 5.2 Diagrama Gestion de Personajes Jugador.

Diagrama

El contenido generado por IA puede ser incorrecto.

## 5.3 Diagrama Gestion Para GM.

Diagrama

El contenido generado por IA puede ser incorrecto.

## 5.4 Diagrama Creación de Personajes.

Diagrama

El contenido generado por IA puede ser incorrecto.

# VI Referencias bibliográficas.

## 4.1 Instrumentos para el Levantamiento de Requerimientos.

<https://www.pccomponentes.com/que-es-un-rpg?srsltid=AfmBOoqOvF_tbyNnSjbZN-Utstig8xgccoIC5M495yRkAw7u8hD8m5sj>

<https://www.geekno.com/glosario/rpg>