

**Documento de Estado del Proyecto**

**Proyecto de Rector Simulator**

**Versión 1.1**

## 1. Introducción

Este documento tiene como objetivo analizar y especificar los requisitos y funcionalidades del proyecto *Rector Simulator*. Se incluirán las historias de usuario, la estructura del proyecto, los elementos técnicos y las herramientas utilizadas para el correcto desarrollo y gestión del proyecto.

## 2. Resumen del Proyecto

**“Rector Simulator”** es un juego de simulación y toma de decisiones en el que los jugadores asumen el rol de un rector universitario. El objetivo es gestionar el presupuesto y recursos de la universidad, mantener la aprobación del alumnado y tomar decisiones críticas que influyen en el éxito de la institución.

## 3. Actividades Realizadas

### 3.1. Crear Repositorio del Proyecto

- **Item:** Repositorio en Github.
- **Estado:** Completado

### 3.2. Desarrollar el Plan de Proyecto

- **Item:** Project Charter.
- **Estado:** Completado

### 3.3. Elaborar Cronograma del Proyecto

- **Item:** Cronograma del proyecto.
- **Estado:** Completado.

### 3.4. Historias de Usuario

- **HU 01:** Documento de historia de usuario 01.
- **HU 02:** Documento de historia de usuario 02.
- **HU 03:** Documento de historia de usuario 03.
- **HU 04:** Documento de historia de usuario 04.
- **HU 05:** Documento de historia de usuario 05.
- **HU 06:** Documento de historia de usuario 06.
- **Estado:** Completado

### 3.5. Elaboración de la Lista de Épicas

- **Item:** Documento de lista de épicas.
- **Estado:** Completado.

### 3.6. Elaboración del Documento de Análisis

- **Item:** Documento de análisis.
- **Estado:** Completado.

### 3.7. Creación de Entorno de Gestión en Jira

- **Ítem:** Configuración del entorno de gestión en Jira.
- **Estado:** Completado.

### 3.8. Reporte del Estado Actual del Proyecto

- **Item:** Documento de estado del proyecto.
- **Estado:** En progreso.

## 4. Requisitos del proyecto:

### 4.1. Requisitos funcionales:

- **Sistema de Toma de Decisiones:** El jugador debe tomar decisiones que afecten la gestión de la universidad.
- **Gestión de Recursos:** Controlar el presupuesto y los fondos de la universidad.
- **Aprobación del Alumnado:** Monitorizar la satisfacción del alumnado a través de un sistema de puntuación.
- **Persistencia de Puntajes:** Guardar los 5 mejores puntajes de cada jugador.
- **Eventos Aleatorios:** Generar eventos que obligan al jugador a tomar decisiones críticas.

### 4.2. Requisitos no funcionales:

- **Compatibilidad:** El juego debe funcionar en navegadores como Firefox, Edge y Google Chrome.
- **Rendimiento:** El sistema debe manejar múltiples eventos simultáneos sin retrasos.
- **Usabilidad:** El diseño debe ser intuitivo y fácil de usar.

## 5. Próximos Pasos

### 5.1. Análisis:

- Especificar Arquitectura y Diseño del Software
- Especificar diseño UI
- Especificar diseño de la Base de Datos
- Reporte de Estado del Proyecto #2

## **5.2. Diseño:**

- Codificación de Historia de usuario 1
- Codificación de Historia de usuario 2
- Codificación de Historia de usuario 3
- Codificación de Historia de usuario 4
- Codificación de Historia de usuario 5
- Codificación de Historia de usuario 6
- Ejecutar Pruebas de Calidad
- Desplegar Aplicación
- Reporte de Estado del Proyecto #3

## **6. Conclusiones**

El desarrollo de la primera etapa del proyecto Rector Simulator ha sido completado satisfactoriamente. Se logró el cumplimiento de todos los hitos establecidos en el cronograma. Durante esta fase, se logró definir claramente los requisitos funcionales y no funcionales del sistema, establecer la estructura técnica mediante la creación del repositorio en GitHub y la configuración del entorno de gestión en Jira, así como documentar las historias de usuario y las épicas necesarias para el desarrollo.