

WRENCH STUDIO



LOS INTEGRANTES DEL GRUPO:

- José Canto Peral (061093573x@uma.es)
- Ignacio Cobos Navarro (0610929636@uma.es)
- Mario Cortés Herrera (0610921089@uma.es)
- Nicolás Melley (nicolomelley@uma.es)
- Javier Molina Colmenero (javimc1610@uma.es)
- Jaime Nevado Farfán (jaimenevadof@uma.es)
- Barbara Parzonka (0611167465@uma.es)
- Andrés Ruiz Sánchez (andresruiz@uma.es)

REPOSITORIO DEL GRUPO:

<https://github.com/JaimeNevado/Wrench-Studio>

TABLA DE CONTENIDOS

Sección 1: Introducción	3
Sección 2: Roles.....	4
Sección 3: Gestión del riesgo	5
Sección 4: Planificación	6
Sección 5: Requisitos.....	7
Sección 6: Herramientas	10

Sección 1: Introducción

Nuestro objetivo es diseñar y desarrollar una aplicación que posibilite a los usuarios pasar tiempo con entretenimiento no adictivo. Para este objetivo proponemos un videojuego en 2D que simulará la vida de un estudiante universitario, para la primera versión del proyecto tendremos un estudiante del grado de Ingeniería del Software. Dentro del juego tendremos la posibilidad de decidir que hace nuestro personaje en cada momento, si estudia, entrena, come, etc... Dependiendo las decisiones que se tomen durante la partida nuestro personaje sacará mejores o peores notas a lo largo de su curso.

El problema de que el juego no sea adictivo lo resolvemos dejando unos tiempos de espera en cada actividad, por ejemplo asistir a una clase podría dejar 30 minutos de espera, además, cuando volvamos de esta espera nos encontraremos con unos datos que representarán lo aprendido durante esa clase. De esta forma tendremos un juego no adictivo, educativo y orientativo ya que le podrá ser de ayuda a muchas personas para hacerse una idea de que trata la carrera y sus asignaturas.

[*Ir a siguiente sección*](#)

Sección 2: Roles

Integrante/Papel	Analista	Diseñador gráfico	Scrum Master	Product Owner	Programador	Tester	Modelador
José Canto Peral							
Ignacio Cobos Navarro							
Mario Cortés Herrera							
Nicoló Melley							
Javier Molina Colmenero							
Jaime Nevado Farfán							
Barbara Parzonka							
Andrés Ruiz Sánchez							

Tabla 1 División de papeles

Rol	Descripción
<i>Analista</i>	Describe requisitos
<i>Diseñador gráfico</i>	Se encarga de gráficos necesarios para software. Diseña la parte visual de la aplicación
<i>Scrum Master</i>	Juega el rol de administrador del proyecto. Cuida el orden de la planificación y los deadline del proyecto. Organiza y dirige reuniones.
<i>Product Owner</i>	Tiene una visión clara del proyecto. Es el decisivo en caso de conflictos. Indica qué debe estar en el backlog.
<i>Programador</i>	Se encarga de la codificación del software
<i>Tester</i>	Hace pruebas del software diseñado, encuentra errores y propone formas de mejorar el sistema
<i>Modelador</i>	Se encarga de la documentación del proyecto. Realiza los diagramas necesarios.

Tabla 2 Descripción de roles

[Ir a siguiente sección](#)

Sección 3: Gestión del riesgo

Tipo del riesgo	Descripción del riesgo	Probabilidad	Efectos del riesgo	Estrategia para mitigar el riesgo
Producto y proyecto	Las herramientas utilizadas tardan en renderizar debido a grandes programas Optimización del juego	Moderada	Tolerable	Encontrar otros programas, trabajar en paralelo, asignar más integrantes a trabajos que exigen mucho tiempo Usar buenas prácticas de programación
Proyecto	Pérdida de información durante el proceso	Moderada	Serio	Hacer back-ups de los archivos trabajados del proyecto
Negocio y producto	Posible incompatibilidad entre macOS y Windows	Alta	Bajo	Dividir las labores incompatibles entre los integrantes del grupo acorde a su sistema operativo
Proyecto	Baja por enfermedad	Baja	Serio	Buena división de tareas
Proyecto	Uso indebido de herramientas, falta de conocimiento y/o experiencia en trabajo con herramientas utilizadas	Moderada	Serio	Documentarse y estudiar el uso de distintos programas
Proyecto	Subestimación del tiempo	Alta	Serio	Planificación eficiente, buena comunicación entre integrantes colaboradores de misma tarea

[Ir a siguiente sección](#)

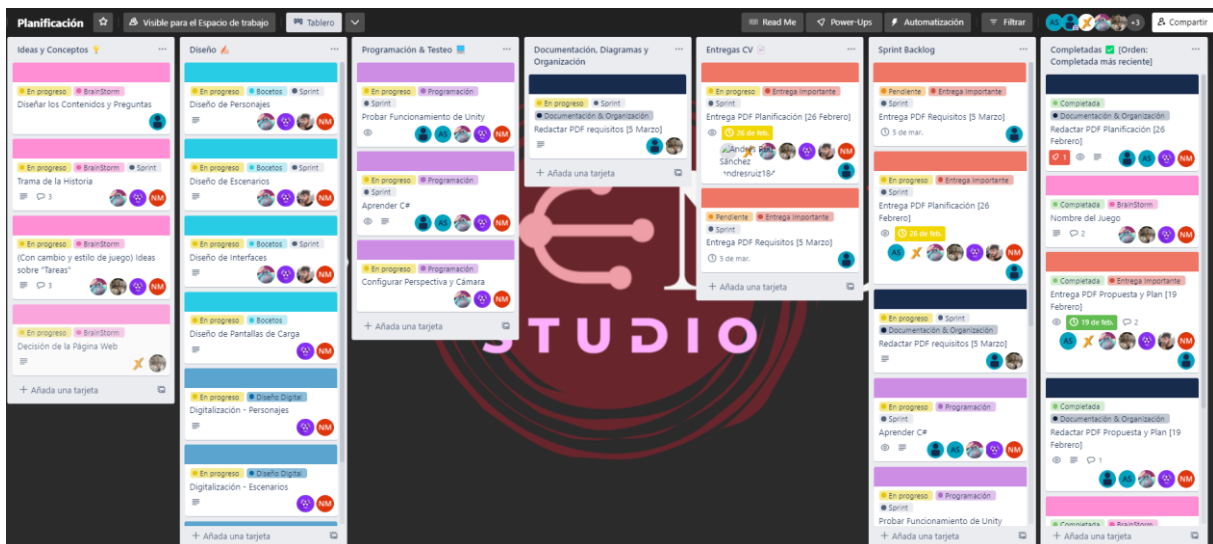
Sección 4: Planificación

Hemos elegido el proceso software ágil ya que puede resultar más útil si se producen cambios y surgen nuevas ideas durante el proyecto. Es más flexible con respecto al plan original. Se centra más en el funcionamiento del software que en detalles de la documentación. Permite colaboración y mayor flexibilidad entre los miembros del equipo.

Enlace para nuestra página trello: <https://trello.com/b/pwwPYUcS/planificaci%C3%B3n>

Nuestro sistema de planificación divide tareas según temas y roles que se deben encargar de esta tarea, haciendo así que el product backlog esté diferenciado en distintos apartados. Además, tenemos una lista Sprint Backlog en la que ponemos todas las tareas correspondientes al sprint actual de forma que podemos ver claramente lo que tenemos que hacer hasta que termine el sprint. Por último, tenemos también una lista con las tareas completadas, para tener constancia de todo lo que hemos terminado en nuestro proyecto.

Aquí se puede ver una captura de nuestro tablero con las distintas listas y tarjetas:



Nota: Está pendiente a cambios y seguramente en el próximo informe cambiemos su estructura.

[Ir a la siguiente sección](#)

Sección 5: Requisitos

NFR01: Lenguaje de programación

Como: programador,
quiero que lenguaje programación usado para desarrollo del sistema sea C#.
Para poder cumplir los objetivos del proyecto.

NFR02: Herramienta programación

Como: programador,
quiero usar herramienta Unity.
Para agilizar trabajo y manejo de interfaz gráfica pensada para el sistema (Aplicación conocida y con mucha documentación).

NFR03: Tiempo limitado de juego

Como: product owner,
quiero que el sistema ponga limitación sobre el tiempo que un usuario pueda usarlo.
Para cumplir el objetivo de desarrollar software no adictivo.

NFR04: Velocidad de ejecución

Como: product owner,
quiero que el sistema no deba tardar demasiado en realizar su ejecución de tareas básicas, cómo movimiento de personaje, aparición del menú, etc.
Para proveer buen rendimiento del sistema y confortabilidad en manejo del sistema.

NFR05: Monousuario

Como: programador,
Quiero que el juego (sistema) solo proporcione acceso a un sólo usuario del mismo equipo en un momento concreto
Para facilitar el funcionamiento del sistema (y que el juego es para un jugador).

NFR06: Diseño de gráficas

Como: diseñador gráfico,
quiero usar herramientas enumeradas a continuación: Piskel, Illustrator, Photoshop y Canva.
Para diseñar gráficos necesarios para el sistema.

NFR07: Derechos de autor de gráficas

Como: product owner,
quiero que las gráficas usadas en sistema sean diseñadas por los miembros de equipo (diseñadores gráficos)
Para facilitar el manejo de ley de derecho de autor y asegurar la propiedad de los assets.

NFR08: Conjunto de caracteres usados

Como: product owner,
quiero que se use en todas las comunicaciones el conjunto de caracteres ASCII estándar
Para facilitar comunicación y hacer más legible todas notificaciones.

FR01: Escenario

Como: product owner,
quiero que el sistema muestre el escenario, por cuál se mueve el usuario y proporcione su posición.
Para facilitar el manejo del movimiento de jugador.

Pruebas de aceptación

- Al encender, se nos muestra el escenario de la Universidad, el piso del jugador, la cafetería, entre otros; dónde pasa la trama del juego.
- Cuando el usuario se mueve, la visión del jugador se adapta al movimiento.

FR02: Movimiento por el mapa

Como: usuario,
quiero que el sistema permita moverse por el mapa usando el teclado (A,W,S y D o las flechas de dirección).
Para facilitar el manejo del movimiento del jugador.

Pruebas de aceptación

- Personaje (y por tanto la pantalla de mapa) se muestran, proporcionando al usuario ciertas funcionalidades de movimiento y acciones dentro del juego. (Al pulsar “arriba o W”, se mueve hacia arriba, al pulsar “derecha o D”, hacia la derecha, pulsar “Esc” sale la pausa del juego, etc.)
- En los menús se maneja con las mismas teclas de dirección de movimiento, con teclas como click izquierdo se podría marcar como aceptación de la opción que se selecciona.
- **Nota:** hay muchos controles que todavía tenemos que pensar que hacer pero que por el poco tiempo que llevamos de desarrollo no hemos profundizado tanto.

FR03: Menú de opciones

Como: product owner,
quiero que el sistema proporcione al usuario el poder de elegir que acción (de las disponibles) quiere ejecutar en ciertos momentos.
Para proveer al usuario entretenimiento mediante actividades del software desarrollado.

Pruebas de aceptación

- En pantalla del juego hay un botón que al pulsar muestra menú de opciones básicas (pausa) cómo salir o parar.
- En momentos elegidos por diseñadores, aparece menú de opciones con posibles acciones, que el usuario puede elegir.
- El usuario debe poder obtener breve información sobre cada acción antes de elegir.
- Al elegir distintas acciones, se espera diferentes resultados.

FR04: Tiempo de espera

Como: product owner,
quiero que al sistema se pueda interrumpir debido a ciertas condiciones
Para evitar la posibilidad de adicción al juego.

Pruebas de aceptación

- Ciertas acciones en el juego son tener que esperar X tiempo a que el personaje las complete, como por ejemplo ir a clase o estudiar. Estas esperas se recompensarán en el juego.
- Después de x tiempo, el juego debe volver al momento parado para poder continuar el juego.
- Al iniciar una actividad se mostrará un contador de tiempo y el jugador no podrá hacer nada hasta que termine si acepta realizarla. No hacer suficientes acciones así puede repercutir en las recompensas del juego.

FR05: Oportunidad de aprendizaje

Como: product owner,

quiero que el sistema proporcione al usuario la posibilidad de aprendizaje y prueba de su conocimiento del grado de Ingeniería de Software.

Para fines educativos.

FR06: Monotarea

Como: programador,

quiero que el usuario pueda ocuparse sólo de una tarea al mismo tiempo

Para facilitar el manejo del juego.

Pruebas de aceptación

- Una acción elegida tiene que completarse o ser parada antes de poder seguir con otra tarea.

OFR01: Modos del juego

Como: product owner,

quiero que se pueda elegir entre diferentes modos de juego, por ejemplo, elegir qué acciones hacer para el personaje, como estudiar, ir a comprar, investigar los escenarios, cambiar el atuendo, y muchas más ideas que se nos ocurran.

Para crear diversidad en el juego.

Pruebas de aceptación

- Según se juega, el jugador decide las acciones del personaje para avanzar en la trama de este.

Sección 6: Herramientas

Durante la realización del proyecto el grupo usó varias herramientas.

- ⇒ **Comunicación**
 - WhatsApp
 - Discord
- ⇒ **Elaboración de documentos**
 - Docs Google
 - Microsoft Word
- ⇒ **Trabajo colaborativo**
 - Github
- ⇒ **Diseño de gráficas**
 - Piskel
 - Illustrator
 - Photoshop
 - Canva
- ⇒ **Programación**
 - Visual Studio Code
 - Unity
- ⇒ **Planificación**
 - Trello

[Volver a comienzo](#)