



Institut Puig Castellar
Projecte / Crèdit de síntesi
Document funcional

DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Hector Aguilera Lopez Nacho Martin Martin

1. Introducción y contexto

Objetivo: explicar de qué trata el proyecto y por qué es necesario.

Describe:

- El problema o necesidad que se quiere resolver.
- Quién será el usuario o cliente final.
- Qué solución se propone y con qué propósito.

Ejemplo:

Se desarrollará una aplicación web para gestionar las reservas del gimnasio del centro. Permitirá a los usuarios consultar y reservar espacios desde el móvil.

Nuestro proyecto busca que personas de todas las edades puedan tener un sitio web para jugar a los juegos simples clásicos con amigos, todo en un mismo lugar. La plataforma ofrece tres juegos: Flappy Bird (esquivar tuberías), Space Invaders (disparar aliens con una nave) y Jetpack Astronaut (astronauta con jetpack que esquivaba aliens).

2. Análisis de requisitos

2.1. Requisitos funcionales (RF)

Qué debe hacer el sistema.

Enumera las funciones principales, numeradas como RF1, RF2, etc.

Código	Descripción del requisito funcional
RF1	Se mostrará la puntuación en tiempo real durante cada partida.
RF2	Los usuarios podrán acceder a los tres juegos y jugar.
RF3	Cada juego tendrá sus propias mecánicas (tuberías, disparo de aliens, esquiva con jetpack).
RF4	El sistema guardará la puntuación máxima (high score) de cada sesión.
RF5	La web mostrará un menú principal con acceso a los tres juegos.

2.2. Requisitos no funcionales (RNF)

Cómo debe comportarse el sistema.

Incluye aspectos como rendimiento, seguridad, compatibilidad o facilidad de uso.

Código	Descripción del requisito no funcional
RNF1	Las páginas deberán cargarse en menos de tres segundos.
RNF2	Los elementos generados aleatoriamente (tuberías, aliens, obstáculos) deben variar en cada partida.
RNF3	Los juegos deben ser compatibles con los navegadores modernos más habituales.
RNF4	La interfaz debe ser clara e intuitiva para usuarios de todas las edades.

2.3. Restricciones

Condiciones o limitaciones del proyecto.

- **Lenguajes o tecnologías obligatorias:**

HTML5 – Estructura y Canvas para el renderizado de los juegos.

CSS3 – Diseño visual y estilos de la plataforma web.

JavaScript – Lógica de los tres juegos (movimiento, colisiones, generación de elementos, puntuación).

Recursos gráficos y sonidos – Pueden ser importados en formatos como PNG, WAV o OGG para sprites, animaciones y efectos de sonido.

- **Recursos disponibles:**

Tiempo de desarrollo: Limitado, ajustado al calendario del curso.

Equipo de trabajo: Grupos de máximo 2 integrantes.

VS Code instalado en los equipos del centro.

Computadoras con soporte para ejecutar el navegador y los juegos en HTML5/Canvas.

Materiales del curso: tutoriales, documentación de MDN y W3Schools, ejemplos de minijuegos proporcionados o buscados por los alumnos.

Recursos gráficos y sonoros:

Sprites de los personajes y fondos (PNG o similar).

Fondos y animaciones simples.

Efectos de sonido básicos para salto, colisión y puntuación.

Dependencias o limitaciones técnicas:

3. Análisis de usuarios y roles

Objetivo: identificar quién usará el sistema y qué podrá hacer.

Describe los distintos tipos de usuario, sus necesidades y sus permisos.

Rol	Descripción	Permisos principales
Administrador	Gestiona usuarios y puntuaciones.	Alta, baja y modificación de datos.
Usuario	Utiliza el sistema para jugar a los tres juegos.	Jugar y consultar sus propios datos.

4. Casos de uso / Escenarios de uso

Objetivo: mostrar cómo interactúan los usuarios con el sistema.

Selecciona de tres a cinco casos principales y descríbelos brevemente.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Utilizar los juegos	Usuario	Accede a los tres juegos correctamente y puede jugar a cualquiera de ellos.	Funcionamiento correcto.
CU2	Jugar a Flappy Bird	Usuario	El usuario controla el pájaro esquivando tuberías.	El juego muestra puntuación y detecta colisiones.
CU3	Jugar a Space Invaders	Usuario	El usuario maneja la nave y dispara a los aliens.	Los aliens se eliminan y la puntuación aumenta.
CU4	Jugar a Jetpack Astronaut	Usuario	El astronauta avanza esquivando aliens.	El juego detecta colisiones correctamente.
—	—	Usuario	—	—

5. Modelo de datos o estructura de la información

Objetivo: representar la información que gestionará el sistema.

Incluye las entidades principales (tablas u objetos) y las relaciones entre ellas.

Entidades principales

1. Usuario
2. Juego
3. Partida

Relaciones

- Un usuario puede jugar a cualquiera de los tres juegos.
- Un juego puede ser jugado por múltiples usuarios.
- Una partida pertenece a un usuario y a un juego, y tiene una puntuación diferente.

Tabla resumen de campos

Entidad	Campos
Usuario	id_usuario (PK)
Juego	id_juego (PK), nombre, descripción, url_archivo (ruta al juego)
Partida	id_partida (PK), id_usuario (FK), id_juego (FK), puntuacion, fecha

6. Diseño de la interfaz

Objetivo: visualizar la estructura y navegación del sistema antes de desarrollarlo.

Incluye bocetos o capturas de las pantallas principales y una breve descripción de su función.
Para cada pantalla, indica:

Pantalla 1: Inicio

Funcionalidad:

Presenta la página principal con el título de la web y acceso al menú de los tres juegos.

Casos de uso:

- CU1 Acceder a la web y ver los juegos disponibles.

Pantalla 2: Menú de Juegos

Funcionalidad:

Muestra los tres juegos disponibles (Flappy Bird, Space Invaders y Jetpack Astronaut) con botones de acceso a cada uno.

Casos de uso:

- CU2, CU3, CU4 — Seleccionar e iniciar un juego.

Pantalla 3: Pantalla de juego

Funcionalidad:

Muestra el juego seleccionado en funcionamiento con marcador de puntuación y botón de vuelta al menú.

Casos de uso:

- CU2, CU3, CU4 — Jugar y consultar puntuación.

7. Planificación técnica

Objetivo: planificar el desarrollo del proyecto.

Indica las tecnologías y herramientas que se utilizarán, y cómo se organizará el trabajo.

Lenguajes y frameworks

- **Frontend:** HTML5, CSS3 y JavaScript (Canvas API para los juegos).
- **Backend:** PHP o Node.js (dependiendo de disponibilidad).
- **Framework opcional:** Express (Node) o ningún framework si se trabaja en PHP sencillo.

Herramientas de diseño o edición

- Figma / Canva (bocetos de interfaz)
- VS Code (editor de código)
- GitHub (control de versiones y bajada de assets gráficos)

Reparto de tareas

- Héctor: estructura backend, lógica de inicio de sesión y puntuaciones.
- Nacho: interfaz visual, integración de los tres juegos, diseño general.
- Ambos pueden colaborar en ajustes finales.

Cronograma estimado

- Semana 1: diseño, estructura del proyecto.
- Semana 2: desarrollo del sistema de usuarios (registro, puntuación).
- Semana 3: integración de los tres juegos.
- Semana 4: pruebas, mejoras y entrega final.

8. Análisis de riesgos

Objetivo: identificar posibles problemas y cómo se afrontarán.

8.1. Identificación de riesgos

Ejemplos:

- Falta de tiempo: puede ocurrir si las tareas no se reparten bien o si se subestima el trabajo necesario al desarrollar tres juegos.
- Problemas técnicos: pueden aparecer al integrar los minijuegos o al conectar el frontend con el backend.
- Desorganización del grupo: si no hay comunicación clara, puede haber retrasos o malentendidos.
- Cambios en los requisitos: si se decide añadir nuevos juegos o funciones a mitad del desarrollo, puede aumentar el trabajo.

8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará.

- Falta de tiempo: es un riesgo con alta probabilidad e impacto. Para reducirlo, se repartirán las tareas desde el inicio y se marcarán objetivos semanales.
- Problemas técnicos: tienen una probabilidad media. Para prevenirlos, se probarán los minijuegos por separado y se revisará la compatibilidad antes de integrarlos.
- Pérdida de datos: su probabilidad es baja, pero el impacto sería alto. Se harán copias de seguridad del código en GitHub.
- Desorganización del grupo: probabilidad media. Se evitará con comunicación frecuente y una revisión del progreso cada pocos días.
- Cambios en los requisitos: probabilidad baja. Para controlarlo, se definirá desde el principio qué juegos y funciones se incluirán y solo se añadirán extras si sobra tiempo.

9. Validación y criterios de éxito

Objetivo: definir cómo sabremos que el proyecto funciona correctamente.

- Los tres juegos funcionan bien y sin errores.
- La interfaz es clara y se ve bien.

Se realizarán pruebas funcionales (acceder a los juegos, sistema de puntuación), pruebas con usuarios (compañeros que prueben la web) y pruebas de rendimiento para comprobar que todo carga en pocos segundos.

10. Conclusión

Objetivo: cerrar el análisis y preparar la siguiente fase.

- El análisis ha permitido definir de forma clara el funcionamiento del sistema: una plataforma sencilla que ofrece tres juegos clásicos (Flappy Bird, Space Invaders y Jetpack Astronaut), accesibles desde cualquier dispositivo con navegador. También se han elegido las tecnologías necesarias (HTML5, CSS3, JavaScript) y se ha planificado el trabajo de forma ordenada entre Héctor y Nacho.

Finaliza con una breve reflexión o frase de cierre, por ejemplo:

"El análisis funcional proporciona una visión clara y realista del proyecto. A partir de aquí, comienza la fase de desarrollo."