****

MÓDULO PROYECTO

|  |
| --- |
| CFGS Desarrollo de Aplicaciones Multiplataforma  Informática y Comunicaciones |

|  |
| --- |
| **EL OASIS** |
| ***Tutor individual:*** *Juan Antonio Alonso Velasco*  ***Tutor colectivo:*** *Cristina Silván Pardo*  ***Año:*** *2023-2024*  ***Fecha de presentación:*** *17/06/2024* |

|  |  |
| --- | --- |
| **Nombre y Apellidos: Sergio Benito Aranzana**  **Email: sergy2210@gmail.com** |  |

índice

[1 Identificación proyecto 4](#_Toc127986556)

[2 Organización de la memoria 4](#_Toc127986557)

[3 Descripción general del proyecto 4](#_Toc127986558)

[3.1 Objetivos 4](#_Toc127986559)

[3.2 Cuestiones metodológicas 4](#_Toc127986560)

[3.3 Entorno de trabajo (tecnologías de desarrollo y herramientas) 4](#_Toc127986561)

[4 Descripción general del producto 4](#_Toc127986562)

[4.1 Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar. 4](#_Toc127986563)

[4.2 Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas. 4](#_Toc127986564)

[4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha 4](#_Toc127986565)

[5 Planificación y presupuesto 4](#_Toc127986566)

[6 Documentación Técnica: análisis, diseño, implementación y pruebas. 4](#_Toc127986567)

[6.1 Especificación de requisitos 4](#_Toc127986568)

[6.2 Análisis del sistema 4](#_Toc127986569)

[6.3 Diseño del sistema: 4](#_Toc127986570)

[6.3.1 Diseño de la Base de Datos 4](#_Toc127986571)

[6.3.2 Diseño de la Interfaz de usuario. 4](#_Toc127986572)

[6.3.3 Diseño de la Aplicación. 4](#_Toc127986573)

[6.4 Implementación: 5](#_Toc127986574)

[6.4.1 Entorno de desarrollo. 5](#_Toc127986575)

[6.4.2 Estructura del código. 5](#_Toc127986576)

[6.4.3 Cuestiones de diseño e implementación reseñables. 5](#_Toc127986577)

[6.5 Pruebas. 5](#_Toc127986578)

[7 Manuales de usuario 5](#_Toc127986579)

[7.1 Manual de usuario 5](#_Toc127986580)

[7.2 Manual de instalación 5](#_Toc127986581)

[8 Conclusiones y posibles ampliaciones 5](#_Toc127986582)

[9 Bibliografía 5](#_Toc127986583)

[10 Anexos 5](#_Toc127986584)

# 

# Descripción general del proyecto

## Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación integral para la gestión y promoción de un centro de actividades y eventos denominado "EL OASIS". El proyecto incluye el desarrollo de una aplicación cliente, una API robusta y una base de datos que respalde todas las operaciones necesarias. Los objetivos específicos del proyecto son los siguientes:

1. **Desarrollo de la Aplicación Cliente**:
   * Crear una interfaz de usuario intuitiva y accesible que permita a los usuarios navegar fácilmente por las distintas secciones del centro.
   * Implementar funcionalidades que permitan a los usuarios registrarse, iniciar sesión, y gestionar sus perfiles.
   * Proveer a los usuarios con herramientas para buscar y reservar actividades y eventos ofrecidos por el centro.
2. **Implementación de la API**:
   * Desarrollar una API RESTful que sirva como intermediario entre la aplicación cliente y la base de datos.
   * Asegurar que la API gestione eficientemente las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para todas las entidades relevantes como usuarios, eventos, reservas, etc.
   * Implementar medidas de seguridad para proteger los datos de los usuarios y asegurar la integridad de las operaciones realizadas a través de la API.
3. **Diseño y Gestión de la Base de Datos**:
   * Diseñar una base de datos relacional que almacene de manera eficiente toda la información necesaria para el funcionamiento del centro.
   * Asegurar la integridad y consistencia de los datos mediante el uso de restricciones y claves foráneas.
   * Optimizar las consultas para mejorar el rendimiento de la aplicación, asegurando tiempos de respuesta rápidos y eficientes.
4. **Integración y Pruebas**:
   * Integrar todos los componentes del proyecto (aplicación cliente, API y base de datos) de manera cohesiva.
   * Realizar pruebas exhaustivas para identificar y corregir errores y asegurar que todas las funcionalidades trabajen según lo esperado.
   * Recoger feedback de usuarios potenciales para realizar mejoras y ajustes necesarios antes del lanzamiento oficial.
5. **Documentación**:
   * Elaborar una documentación detallada que cubra todos los aspectos del desarrollo del proyecto, incluyendo la arquitectura de la aplicación, el diseño de la base de datos, y el uso de la API.
   * Proveer guías de usuario y manuales técnicos para facilitar el mantenimiento futuro y posibles expansiones del sistema.
6. **Escalabilidad y Mantenimiento**:
   * Diseñar la aplicación con una arquitectura escalable que permita futuras ampliaciones de funcionalidades sin comprometer el rendimiento.
   * Asegurar que el código sea limpio, modular y bien documentado para facilitar el mantenimiento y las futuras actualizaciones del sistema.

## Cuestiones metodológicas

Para asegurar el éxito del proyecto "EL OASIS", se han seguido una serie de métodos y prácticas estándar en el desarrollo de software. A continuación, se describen las cuestiones metodológicas clave que han guiado el desarrollo del proyecto:

* **1. Análisis de Requisitos**

El primer paso del proyecto fue llevar a cabo un análisis exhaustivo de requisitos para entender las necesidades y expectativas de los usuarios finales y las partes interesadas. Este análisis incluyó:

* **Reuniones con los Stakeholders**: Reuniones regulares con los administradores del centro y potenciales usuarios para definir y priorizar funcionalidades.
* **Estudio de Mercado**: Investigación de aplicaciones similares para identificar características importantes y oportunidades de mejora.
* **Especificaciones de Requisitos**: Documentación de los requisitos funcionales y no funcionales del sistema.
* **2. Diseño del Sistema**

Con los requisitos claramente definidos, se procedió a la fase de diseño del sistema:

* **Arquitectura del Sistema**: Se diseñó una arquitectura de tres capas (cliente, servidor/API y base de datos) para separar las responsabilidades y mejorar la mantenibilidad.
* **Modelado UML**: Se utilizaron diagramas UML (Diagramas de Clases, de Secuencia, y de Casos de Uso) para representar gráficamente la estructura y comportamiento del sistema.
* **Prototipado**: Desarrollo de prototipos de baja fidelidad para validar la interfaz de usuario y las principales interacciones antes de la implementación.
* **3. Desarrollo Ágil**

Se adoptó una metodología ágil para el desarrollo del software, permitiendo iteraciones rápidas y ajustes basados en el feedback continuo:

* **Sprints**: El trabajo se organizó en sprints de dos semanas, con objetivos claros y entregables definidos para cada sprint.
* **Scrum**: Se realizaron reuniones diarias de Scrum para revisar el progreso, identificar obstáculos y ajustar las prioridades según fuera necesario.
* **Integración Continua**: Se utilizó un sistema de integración continua para integrar y probar el código frecuentemente, asegurando la detección temprana de errores.
* **4. Desarrollo y Pruebas**

El desarrollo se llevó a cabo en paralelo con pruebas continuas para garantizar la calidad del software:

* **Desarrollo Modular**: El código se desarrolló de manera modular, facilitando la reutilización y el mantenimiento.
* **Pruebas Unitarias**: Cada módulo fue sometido a pruebas unitarias para verificar su correcto funcionamiento de manera aislada.
* **Pruebas de Integración**: Se realizaron pruebas de integración para asegurar que los diferentes módulos del sistema funcionaran juntos sin problemas.
* **Pruebas de Usuario**: Se llevaron a cabo pruebas de usuario con versiones beta para recopilar feedback y realizar ajustes necesarios.
* **5. Implementación de la API y Base de Datos**

La API y la base de datos se desarrollaron siguiendo buenas prácticas para asegurar eficiencia y seguridad:

* **Diseño de la API**: Se diseñó una API RESTful con endpoints claros y documentados utilizando Swagger/OpenAPI.
* **Seguridad**: Implementación de autenticación y autorización para proteger los datos y asegurar que solo los usuarios autorizados puedan acceder a la información.
* **Optimización de la Base de Datos**: Se diseñó y normalizó la base de datos para evitar redundancias y asegurar la integridad de los datos.
* **6. Documentación y Mantenimiento**

La documentación y el plan de mantenimiento son cruciales para la sostenibilidad del proyecto:

* **Documentación Técnica**: Documentación detallada del código, arquitectura del sistema y API.
* **Guías de Usuario**: Manuales y tutoriales para ayudar a los usuarios finales a navegar y utilizar la aplicación.
* **Plan de Mantenimiento**: Estrategias para la actualización del sistema, manejo de errores y futuras expansiones.
* **Conclusión**

Las metodologías utilizadas en el desarrollo del proyecto "EL OASIS" han sido esenciales para garantizar la calidad, eficiencia y sostenibilidad del sistema. A través de un enfoque ágil y iterativo, combinado con un análisis detallado de requisitos y un diseño cuidadoso, se ha logrado crear una solución tecnológica robusta y alineada con las necesidades de los usuarios y las partes interesadas.

## Entorno de trabajo (tecnologías de desarrollo y herramientas)

Tecnologías de Desarrollo

Lenguajes de Programación y Marcas:

HTML: Utilizado para la estructura del contenido web.

CSS: Usado para el diseño y la presentación visual de la aplicación.

JavaScript: Utilizado para la interactividad y funcionalidad dinámica en la parte del cliente.

SQL: Utilizado para la gestión y consulta de la base de datos.

Frameworks y Librerías:

Node.js: Utilizado en el backend para construir la API y manejar el servidor.

Express.js: Framework para Node.js que facilita la creación de aplicaciones web y APIs.

jQuery: Librería de JavaScript para simplificar la manipulación del DOM y las solicitudes AJAX.

Base de Datos:

MySQL: Utilizado para almacenar y gestionar los datos de la aplicación. El archivo tfg.sql indica el uso de esta base de datos.

Herramientas de Desarrollo

Control de Versiones:

Git: Utilizado para el control de versiones del código fuente. Evidencia de esto es la presencia de un directorio .git.

Entorno de Desarrollo:

Visual Studio Code: Posiblemente usado como entorno de desarrollo integrado (IDE), evidenciado por la carpeta .vscode en el proyecto de la API.

Gestión de Dependencias:

npm (Node Package Manager): Utilizado para la gestión de dependencias de Node.js, como se puede ver en los archivos package.json y package-lock.json.

Archivos de Configuración:

.env: Archivo de configuración utilizado para manejar variables de entorno, probablemente conteniendo configuraciones sensibles como claves de API y credenciales de base de datos.

Estructura del Proyecto

Frontend (EL\_OASIS):

css: Contiene los archivos CSS para el estilo de la aplicación.

html: Contiene los archivos HTML que forman la estructura de las páginas web.

imagenes: Almacena imágenes utilizadas en la aplicación.

js: Contiene archivos JavaScript que manejan la lógica del cliente.

fuente y TFG: Posiblemente contienen fuentes de texto y otros documentos relacionados con el proyecto.

README.md: Documento que probablemente contiene información sobre cómo configurar y ejecutar el proyecto.

Backend/API (APIMAP):

bbdd: Puede contener scripts SQL o configuraciones relacionadas con la base de datos.

events, services, src: Directorios que probablemente contienen la lógica de negocio, servicios de la aplicación y el código fuente principal del backend.

node\_modules: Carpeta que contiene las dependencias instaladas por npm.

# Descripción general del producto

## Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.

## Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.

## Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha

# Planificación y presupuesto

# Documentación Técnica: análisis, diseño, implementación y pruebas.

## Especificación de requisitos

## Análisis del sistema

## Diseño del sistema:

### Diseño de la Base de Datos

### Diseño de la Interfaz de usuario.

### Diseño de la Aplicación.

## Implementación:

### Entorno de desarrollo.

### Estructura del código.

### Cuestiones de diseño e implementación reseñables.

## Pruebas.

# Manuales de usuario

## Manual de usuario

## Manual de instalación

# Conclusiones y posibles ampliaciones

# Bibliografía

# Anexos