

Documentación del Proyecto

Proyecto Desarrollo de Aplicaciones Web

Índice

Propuesta, definición y análisis del proyecto

1. Definición del proyecto
 - 1.1 Obtención de la información necesaria para la definición del proyecto
 - 1.2 Descripción detallada del proyecto
 - 1.3 Ámbito del proyecto
 - 1.4 ¿Adaptación o creación
 - 1.5 Conceptos básicos del proyecto software
 - 1.6 Definición de tareas
 - 1.7 Estimación de tiempos

Análisis del Proyecto Web

- 1.8 Requisitos técnicos web
- 1.9 Elementos del contenido
- 1.10 Diseño gráfico
- 1.11 Herramientas para web

Ingeniería del proyecto web

- 1.12 Diseño del software
- 1.13 Diseño de datos
- 1.14 Diseño del sistema
- 1.15 Diseño de la interfaz de usuario
- 1.16 Diseño del contenido

Identificación y cuantificación de contingencias

2. Aseguramiento de la calidad

Propuesta, definición y análisis del proyecto

1. Definición del proyecto

- **Obtención de la información necesaria para la definición del proyecto**

Me he guiado por una aplicación llamada Numier tpv, una aplicación de gestión de restaurantes de pago, en un restaurante de un familiar mio

- **Descripción detallada del proyecto**

Se trata de una aplicación web en la que tanto dueños de restaurantes/bares como sus empleados pueden gestionar digitalmente los pedidos de su local de una manera más rápida y eficaz.

El usuario con rol de Dueño puede gestionar su local añadiendo productos a la carta y/o añadiendo empleados a su local.

Mientras un empleado con rol de Camarero podrá crear/editar/finalizar pedidos en el local y el cocinero podrá ver estos pedidos y realizarlos haciéndolo saber a los empleados del restaurante.

- **Ámbito del proyecto**

Para la parte del desarrollo del Back-End he decidido escoger Spring Tools Suite para Eclipse, es un IDE de Eclipse el cual te viene integrado el plugin de Spring de manera nativa el cual te ahorra configuraciones y con Eclipse se hace mucho más fácil la instalación de las dependencias. Con el cual lo he desarrollado en una API Rest.

Para la parte del desarrollo del Front-End he elegido el Framework de Google, Angular, en Visual Studio Code. Ya que es un Framework muy conocido y muy fácil de aprender.

Por último para la parte de la base de datos he elegido MariaDB.

- **¿Adaptación o creación?**

Este proyecto ha sido una adaptación ya que ya existen actualmente otras aplicaciones que se dedica a lo mismo

- **Conceptos básicos del proyecto software**

Para la correcta comprensión del documento conviene conocer algunos conceptos básicos.

- **Framework:** Es una estructura conceptual y tecnológica de soporte definido, normalmente con módulos de software específicos, que pueden servir de base para la organización y desarrollo de software.
- **TypeScript:** Lenguaje de programación de código abierto desarrollado por Microsoft.
- **Angular:** Es un framework para aplicaciones web desarrollado en TypeScript que se usa para crear y mantener aplicaciones de una sola página.
- **Base de datos relacional:** Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- **ORM:** Técnica de programación que convierte los datos del sistema de tipos en un lenguaje orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.
- **Spring:** Framework para el desarrollo de aplicaciones y contenedor de inversión de control para Java.

Planificación del proyecto web.

- Definición de tareas

- Realización de la maqueta: con el software de Axure realizamos un MockUp con el aspecto que queremos que la aplicación tenga.
- Realización de la estructura del Back-End, modelo de datos ORM, repositorios y controladores necesarios para la API Rest.
- Implementación de seguridad en la parte del Back-End mediante JWT y Spring Security
- Realización de las pruebas pertinentes de la API mediante Postman
- Realización de la estructura del Front-End, creación de componentes, servicios e interfaces necesarios para el desarrollo
- Implementación de seguridad en la parte del Front-End mediante JWT y Guardians
- Realización de las vistas de los componentes y desarrollo de los mismos componentes enlazándolos con los servicios creados, que estos sirven de puente con la API.
- Testeo de la aplicación, para detectar posibles fallos y corregir estos.
- Despliegue de la aplicación, usando plataforma como Heroku para el despliegue del Back-End, la plataforma de GitHub Pages para el Front-End y Clever Cloud para el alojamiento de la base de datos.
- Creación de la documentación del proyecto.

- Estimación de tiempos

Usaremos % del tiempo total empleado en las tareas realizadas en el punto anterior.

1. 10%
2. 25%
3. 5%
4. 5%
5. 10%
6. 5%
7. 25%
8. 5%
9. 5%
10. 5%

Análisis del Proyecto Web

- Requisitos técnicos web

En el desarrollo del Back-End se ha usado el IDE Spring Tools Suite Eclipse, es Eclipse pero con el plugin de Spring integrado. Y el lenguaje de programación ha sido Java con el framework de Spring para el desarrollo de aplicaciones y contenedor de inversión de control.

En el desarrollo del Front-End ha sido desarrollado con el entorno de desarrollo Visual Studio Code. La parte del diseño ha sido desarrollada con HTML y CSS/SCSS, haciéndolo responsive con Bootstrap 5 y el lenguaje de programación se ha realizado con TypeScript en el Framework de Angular.

Se ha usado una base de datos relacional SQL, usando como gestor de base de datos MariaDB y DBeaver como herramienta visual para la gestión de la base de datos.

Para el control de versiones se ha usado Git y se ha hecho pruebas del código a través de Postman en la parte Back-End y test manuales en la parte visual del Front-End, Chrome.

Por último se ha desplegado la aplicación en plataformas como Heroku (parte Back-End), Clever Cloud (parte Base de Datos) y GitHub Pages (parte Front-End).

- Elementos del contenido

Los principales elementos que se puede encontrar en nuestra aplicación son: Texto, imágenes, enlaces, formulario reactivos, barra de navegación lateral y superior, botones interactivos, etc.

- Diseño Gráfico

Se ha usado como colores primarios el azul y blanco para el estilo de la página. El logotipo de la aplicación es un logo simple y moderno creado en photoshop. El tipo de letra que ha sido usado 'Meriweather' y el tamaño de la letra es distinto según la función necesaria.

- Herramientas para Web

Se ha usado herramientas como Google Fonts para la fuente de la letra, también Angular Material Icons y PrimeNG Icons para los iconos y por último la librería de PrimeNG para la creación de tablas dinámicas

Ingeniería del proyecto web

- Diseño del software

Para el Back-End disponemos de un Tomcat integrado que dispone el framework de Spring, y las dependencias elegidas son:

- Lombok: Ofrece una librería de anotaciones la cual ayudara a reducir la cantidad de código escrito.
- Spring JPA y MariaDB Driver: Para controlar la persistencia de datos en el almacenamiento SQL.
- Spring Security: Ofrece una seguridad a la API.
- Spring Boot: Reinicia automáticamente el servidor después de guardar algún fichero
- Auth0: Ofrece autenticación por JWT a la API.
- SpringFox: Documenta la API a través de Swagger.
- Spring Mail: Permite el envío de mensajes a través de un correo electrónico.

Para la parte del Front-End disponemos del framework de Angular, y tenemos varias dependencias/módulos que son:

- PrimeNG: Es una colección de componentes de interfaz de usuario para Angular de código abierto.
- Material Angular: Es un módulo construido para angular el cual permite implementar componentes Angular con un diseño basado en Material Design
- Reactive Forms: Es un módulo que permite la creación de formularios reactivos
- auth0: Librería que permite la autenticación mediante JWT.

- Diseño de datos

El sistema de gestión para la base de datos he usado MariaDB, ya que existe una gran cantidad de información para poder realizar cualquier tipo de desarrollo, además de ser de código abierto.

- Diseño del sistema

Capa de presentación:

- Controller: Este gestiona los accesos a los servicios de la api desde la capa de presentación.
- Model: Recoge los modelos de las clases que tiene el proyecto
- Repository: Interfaces que se encargan de la comunicación a la base de datos y que nos provee Spring .
- Security: Recoge las clases que son necesarias para la seguridad de la API
- Service: Intermediario entre los repositorios y los controladores.
- Mail: Recoge las clases necesarias para enviar correo.
- DTO: Para crear un objeto plano con información concreta.
- Exception: Recoge clases con excepciones definidas.
- Resources: Almacena datos de configuración de la aplicación.

- **Diseño de la interfaz de usuario**

Para el diseño de la interfaz se ha tenido en cuenta aspectos como los colores usados, el contraste, la disminución de ruido visual, minimalista y eliminación de elementos innecesarios.

- **Diseño del contenido**

Se ha optado por un estilo minimalista, fácil de visualizar y entender, con elementos y contenido sencillos para el uso.

Utilizando colores claros y con contraste entre ellos, además de una navegación sencilla a través de la barra lateral.

Identificación y cuantificación de contingencias

La reestructuración de las tablas de la base de datos y modelos a causa de la agregación de nuevos campos cuando ha sido necesario.

Además al haber sido desplegado en la nube se han tenido que arreglar ciertos fallos a causa de esto y/o añadir nuevas dependencias al proyecto para garantizar un funcionamiento y diseño correcto

2. Aseguramiento de la calidad

Se han realizado pruebas tanto en el servidor Back-End como en el Front-End para garantizar la calidad de la aplicación web.

A través de Postman hemos podido comprobar el funcionamiento correcto de los controladores de la API que serán necesarios usar en la parte del Front-End.

Mientras tanto en la parte del Front-End se ha comprobado de manera manual probando las funcionalidades en el navegador, comprobando los botones, imágenes, tablas funcionen correctamente.

Además la aplicación ha sido probada por usuarios tester para así comprobar que todo funciona correctamente y tener un feedback ajeno.