TFG

1. Introducción
   1. ¿Qué es lo que pretendo hacer?
   2. ¿Por qué lo que quiero hacer es importante? ¿Qué necesidad hay en el mercado de mi producto, investigación, o similar?

1.- Introducción

En la actualidad, las empresas están sufriendo una transformación en todos sus ámbitos. Las nuevas tecnologías suponen la entrada a un paradigma digital donde todos los servicios deben estar interconectados y funcionales. Las empresas deben ser capaces de operar y relacionarse con sus clientes de manera que puedan adaptarse a los nuevos cambios y retos acordes con los estándares tecnológicos actuales. La digitalización y la implementación de soluciones de IT avanzadas no solo mejoran la eficiencia operativa, sino que también ofrecen nuevas oportunidades para interactuar con los consumidores, construir una imagen de marca sólida y desarrollar estrategias de marketing efectivas que hagan que una empresa destaque frente al resto, o, por el contrario, que desaparezca del mercado. En este contexto, una empresa debe enfocar grandes esfuerzos en la creación de una página web bien diseñada y funcional. Una buena página web se ha convertido en un elemento esencial para cualquier empresa que aspire a destacar en un mercado cada vez más competitivo y tecnológicamente más desarrollado.

Las páginas web modernas son mucho más que simples escaparates estáticos; son plataformas dinámicas que permiten una interacción cliente-empresa mucho más efectiva. Las empresas ofrecen servicios de todo tipo a través de sus propias páginas web, desde la simple y tradicional presentación de información hasta la posibilidad de realizar transacciones, gestionar reservas, ofrecer soporte en tiempo real y mucho más.

Un servicio de gran utilidad es la posibilidad de integrar en una página web empresarial un sistema de fidelización por puntos. Este tipo de sistemas es extremadamente eficaz para fomentar la lealtad del cliente, aumentar la retención y, en última instancia, incrementar los ingresos. En el sector de la restauración, un sistema de puntos como él que hemos creado para este proyecto puede ser especialmente llamativo, ya que permite a los restaurantes crear una relación más cercana y personalizada con sus clientes.

* 1. ¿Qué es lo que pretendo hacer?

Nuestro objetivo en este proyecto es la creación de un sistema de puntos de fidelización enfocado al sector de la restauración. Nuestra intención es ofrecer el esqueleto de un servicio sencillo de implementar y que permita una gran adaptabilidad para las necesidades de cada restaurante.

A través de este sistema, los clientes pueden acumular puntos con cada compra, los cuales pueden canjear por descuentos, promociones especiales u otros beneficios. En nuestra idea, cualquier restaurante podrá implementar un sistema basado en lo que proponemos y que sea completamente operativo y modular.

* 1. ¿Por qué lo que quiero hacer es importante? ¿Qué necesidad hay en el mercado de mi producto, investigación, o similar?

La importancia de este proyecto radica en darle la facilidad y las herramientas a pequeños establecimientos para que tengan sus propios sistemas por puntos en sus locales sin una gran inversión económica ni de tiempo.

Hay una clara tendencia en el mercado enfocada a que los establecimientos tengan una relación más estrecha con sus consumidores a través de las tecnologías. Los consumidores valoran cada vez más las experiencias personalizadas, y un sistema de puntos que se integra con una cuenta de usuario en línea permite al restaurante ofrecer promociones y recompensas adaptadas a los hábitos y preferencias de cada cliente. Esta personalización no solo mejora la satisfacción del cliente, sino que también puede traducirse en un aumento de las ventas y la lealtad del cliente a largo plazo.

Al crear un proyecto sencillo como este permitimos a los locales más pequeños tener un sistema de puntos a través de una web como lo pueden tener grandes empresas de restauración que lideran el mercado actual. Además, mediante el empleo de QR su uso para los usuarios es sencillo de entender al ser una tecnología con lo que están familiarizados.

* 1. Objetivos

1. **Desarrollar y Configurar la Base de Datos:**

* Crear una base de datos robusta que almacene información de usuarios, pedidos, facturas y transacciones.

1. **Diseñar y Desarrollar una Página Web Interactiva con Django:**

* Crear una página web funcional y de fácil uso utilizando el framework Django.
* Incluir funcionalidades de autenticación de usuarios, permitiendo a los clientes registrarse, iniciar sesión y gestionar sus cuentas.

1. **Integrar la Página Web con la Base de Datos:**

* Implementar una API en la página web que permita la interacción con el sistema de puntos leyendo el QR.
* La página web debe poder enviar la información del QR a la base de datos

1. **Desarrollar un Programa para la Terminal del Restaurante:**

* Creación y gestión de las cuentas de clientes. El programa debe ser capaz de generar ticket con las facturas.
* Crear un programa para el terminal del restaurante que permita a los camareros tomar pedidos, generar cuentas y crear códigos QR.
* El programa debe conectarse con la base de datos para leer el menú disponible.

1. **Implementar el Sistema de Puntos de Fidelización:**

* Desarrollar un sistema de puntos que incentive a los clientes a regresar al restaurante, acumulando puntos con cada compra.
* Integrar el sistema de puntos con la API de la página web para que los clientes puedan ver y gestionar sus puntos acumulados.

1. **Implementar el Sistema en Docker:**

* Desarrollar un contenedor que tenga nuestra base de datos y el framework Django para poder hacer pruebas en local

1. **Facilitar la Interacción Digital con los Clientes:**

* Proveer una plataforma intuitiva donde los clientes puedan interactuar fácilmente con el restaurante a través de sus dispositivos.
* Ofrecer una experiencia digital completa, desde la autenticación y gestión de cuentas hasta la realización de transacciones y la visualización de puntos acumulados.

1. Planificación

La planificación de nuestro proyecto llevó a cabo en colaboración mutua y minuciosidad. Las dos primeras semanas las dedicamos exclusivamente a la conceptualización de la idea y desarrollar cuales son los puntos clave que necesitaríamos, así como pensar por encima que tecnologías deberíamos implementar. En este proceso discutimos cuáles serían los objetivos del proyecto y trazamos un plan general de funcionamiento y desarrollo para una mayor fluidez ala hora de trabajar en equipo.

A partir de la segunda semana comenzamos a dividirnos las tareas siguiendo unos primeros objetivos que resumimos y dejamos por escrito. La división de trabajo se realizó de manera estratégica que cada uno podía elegir por qué parte del proyecto empezar, de esta forma cada uno escogió tareas que aprovechaban las fortalezas y habilidades de cada uno.

Tras una semana realizando las tareas que teníamos programadas y estando en comunicación para informar de cualquier necesidad o problema que nos encontramos, nos reuníamos al finalizar la misma para una reunión de control y ver avances. Cada uno mostraba en qué punto se encontraba con respecto a sus tareas asignadas y discutíamos cualquier obstáculo que pudiera haber surgido durante estas reuniones.

Estas reuniones semanales también nos permitían discernir cuáles serían las tareas de la siguiente semana incluyendo nuevas tareas que podían surgir durante el desarrollo del proyecto, asegurando que ambos estuviéramos alineados con los objetivos del proyecto y que el trabajo se realizara de manera coherente. Esta metodología de planificación y revisión semanal nos permitió resolver problemas rápidamente, así como mantener un ritmo de trabajo aceptable.

Las últimas dos semanas las dedicamos a juntar nuestras partes del proyecto e integrarlas completamente para asegurarnos que todo funcionaba correctamente. Estas dos semanas finales también las utilizamos para la escritura y revisión del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| Semana | Guillermo | Jose Daniel |  |
| 1 | Creación idea proyecto | Creación idea proyecto |  |
| 2 | Planificación del desarrollo | Planificación del desarrollo |  |
| 3 | Diseño y creación de la base de datos | Diseñar y Desarrollar una Página Web Interactiva con Django |  |
| 4 | Investigación programa para generar QR | Investigación de API para lector de QR |  |
| 5 | Desarrollar un Programa para la Terminal del Restaurante con generador de QR | Implementación API en Django |  |
| 6 | Dockerizar programas | Desarrollo de interación web – base de datos creando autenticación de usuarios |  |
| 7 | Escritura de trabajo | Puesta a punto del trabajo |  |
| 8 | Escritura de trabajo | Escritura de trabajo |  |
| 9 |  |  |  |

3.- Descripción Informática, o equivalente

En este apartado vamos a explicar de forma detallada en que consiste, como es el diseño del proyecto y que requerimientos hacen falta para poder llevarlo a cabo.

El proyecto podemos dividirlo en dos partes: restaurante y servidor.

1.- Restaurante:

Requisitos hardware:

* Una terminal con acceso a internet. Puede ser un ordenador personal convencional, algún dispositivo tipo Tablet o cualquier dispositivo capaz de correr un programa.
* Una impresora o un dispositivo para imprimir las cuentas en el restaurante. Esto será necesario para poder enseñar de forma física al cliente el QR necesario para que el cliente pueda escanearlo. También es posible mostrar el QR desde un dispositivo móvil o una Tablet para no tener que imprimir en papel.

En nuestro caso estamos utilizando nuestro ordenador personal para simular la terminal del restaurante.

Requisitos Software:

* Es necesario descargar las dependencias adecuadas para poder utilizar Python en la terminal o terminales.
* Instalación de conector Python con la base de datos.

En nuestro caso estamos utilizando una maquina virtual con Ubuntu 22.04.3 LTS. El conector que hemos utilizado es el de Python con MYSQL

2.- Servidor

Requisitos Hardware:

* Este apartado depende de las necesidades del negocio, en caso de un local grande que necesite una gestión importante seria necesario tener un servidor donde alojar los programas que dictaremos más adelante. En caso de no tener un volumen de negocio muy grande es posible crear un pequeño servidor por tu cuenta o incluso utilizar un ordenador. Existe la opción de no necesitar un servidor físico y utilizar los servicios de Cloud para hospedar nuestros programas.

En nuestro caso al ser un entorno de pruebas estamos utilizando nuestros ordenadores personales

Requisitos Software:

* Instalación de las dependencias de Docker y Docker-Compose
* Sistema operativo capaz de aguantar el Docker. (Para una mejor optimización podemos utilizar un sistema operativo de tipo Server).

En nuestro caso estamos utilizando un sistema operativo Ubuntu Desktop al estar en un entorno de pruebas.

Ahora vamos a describir de forma más detallada como estás configurados cada uno de los servicios que necesitamos.

2.2.1 Base de datos.

Vamos a emplear una base de datos MYSQL por nuestros conocimientos previos y manejo que tenemos con este servicio. Se podría sustituir por otra base de datos diferente a elección del empresario.

Diseño:

2.2.2 Programa de Cuentas.

El programa está diseñado para ser utilizado en un restaurante para gestionar los pedidos de las mesas, generar cuentas y crear códigos QR con los detalles de la cuenta. Utiliza una base de datos MySQL para almacenar la información del menú y maneja los pedidos y pagos mediante interacciones en la terminal. Los códigos QR generados permiten a los clientes escanear y obtener puntos que pueden ser canjeados en futuras visitas.

Este programa se trata de un código escrito en Python. Vamos a ir apartado por apartado del código explicando su función:

A screenshot of a computer

Description automatically generatedIMPORTACIONES:

Estas son las importaciones necesarias para el programa:

* mysql.connector: Para conectarse y operar con la base de datos MySQL.
* math: Proporciona funciones matemáticas, en este caso se usa para redondear números.
* json: Para trabajar con datos en formato JSON.
* qrcode: Para generar códigos QR.
* datetime: Para trabajar con fechas y horas.

A screen shot of a computer program

Description automatically generated**Funcion conectar\_base\_datos()**

Esta función intenta conectarse a una base de datos MySQL con las credenciales y parámetros especificados. Si la conexión es exitosa, devuelve la conexión; de lo contrario, imprime un mensaje de error y devuelve None.

Función obtener\_platos\_y\_precios(conexion)

A computer screen with text on it

Description automatically generated

Esta función obtiene una lista de platos y sus precios desde la base de datos. Ejecuta una consulta SQL para seleccionar id\_plato, nombre y precio de la tabla menu, y almacena los resultados en una lista de tuplas.

Función crear\_cuenta\_pedido(platos\_precios, num\_mesa)

A computer screen with text

Description automatically generated

Esta función permite crear una cuenta para un pedido. Los usuarios pueden ingresar el número del plato que desean agregar al pedido. La función suma los precios de los platos seleccionados y devuelve una lista que contiene el número de mesa y el total de la cuenta.

Función seleccionar\_mesa(mesas\_ocupadas)

A screenshot of a computer program

Description automatically generated

Esta función permite seleccionar una mesa para el pedido, asegurándose de que el número de mesa sea válido y que la mesa no esté ya ocupada. Devuelve el número de mesa seleccionado.

A screenshot of a computer code

Description automatically generatedFunción menu()

Simplemente saca por pantalla el despliegue de opciones que se pueden realizar y devuelve en su salida el numero que el usuario ha seleccionado.

Función generar\_json\_pedido(pedido)

A screen shot of a computer code

Description automatically generated

Esta función genera un archivo JSON y un código QR para un pedido. Calcula los puntos del pedido, crea un archivo JSON con los detalles del pedido y genera un código QR que contiene los mismos datos. El archivo JSON y la imagen del código QR se guardan con nombres que incluyen la fecha y hora actuales.

Función main()

A screen shot of a computer program

Description automatically generated

Esta es la función principal del programa. Se encarga de coordinar todas las funciones:

1. Conecta a la base de datos.
2. Obtiene la lista de platos y precios.
3. Mantiene un bucle de operación que permite al usuario seleccionar opciones del menú:
   * Ver la lista de platos y precios.
   * Crear una cuenta para un pedido.
   * Pagar la cuenta de una mesa.
4. Maneja las mesas ocupadas y los pedidos.
5. Genera y guarda el archivo JSON y el código QR cuando se paga una cuenta.

2.2.3 Dockerización de MYSQL y Django

Esta es una parte fundamental del proyecto ya que nos permite crear un contenedor con todo incluido listo para lanzarse y utilizarse. Por supuesto, este proyecto solo es un esqueleto, en un entorno real habría que configurar el contenedor según las necesidades del empresario.

En nuestro caso hemos incluido en Docker tres imágenes: MYSQL, Django y Phpmyadmin. En un entorno real lo ideal sería conectar el Django a un servidor web para lanzar las paginas web de forma adecuada. En nuestro caso no será necesario porque estamos en un entorno de pruebas, por tanto, lanzaremos las paginas web desde el propio framwork de Django que permite esta opción en un entorno local. Phpmyadmin no es necesario para utilizar django o

Pasos a seguir:

1. Necesitamos Instalar Docker y Docker-compose en nuestra maquina, nosotros lo haremos en una maquina virtual de Ubuntu.
   1. Instalar dependencias necesarias:

sudo apt install apt-transport-https ca-certificates curl software-properties-common

* 1. Agregar la clave GPG oficial de Docker:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

* 1. Agregar el repositorio de Docker a tus fuentes APT :

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

* 1. Actualizar el índice de paquetes de nuevo:

sudo apt update

* 1. Instalar Docker:

sudo apt install docker-ce

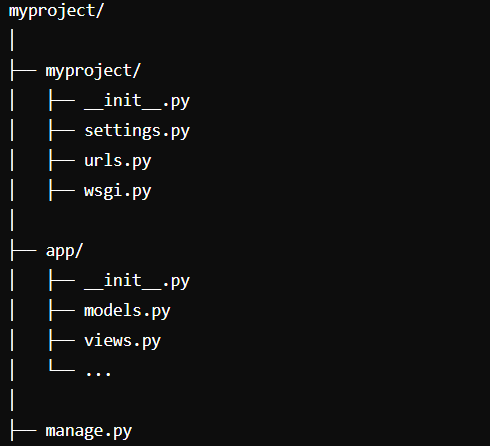
* 1. Instalar Docker-compose:

sudo curl -L "https://github.com/docker/compose/releases/download/<VERSION>/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

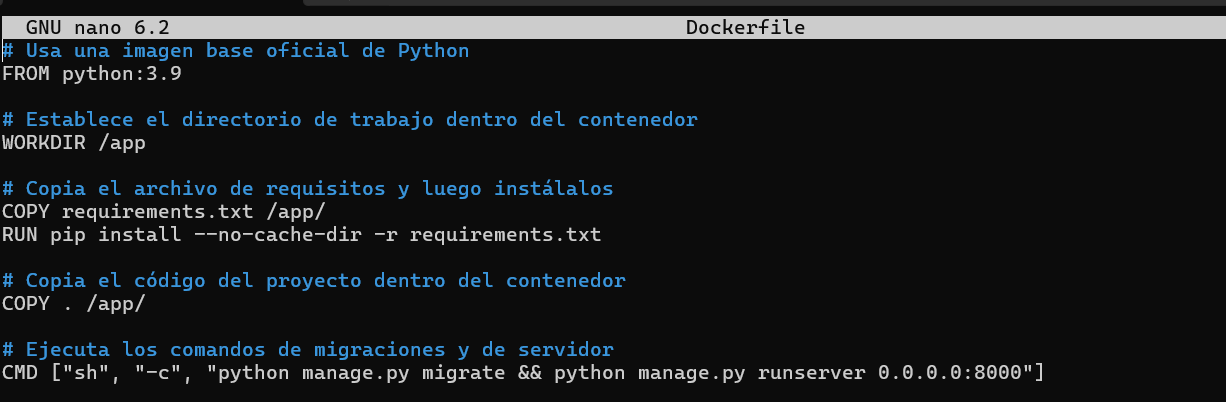
* 1. Dar permisos de ejecución al binario:

sudo chmod +x /usr/local/bin/docker-compose

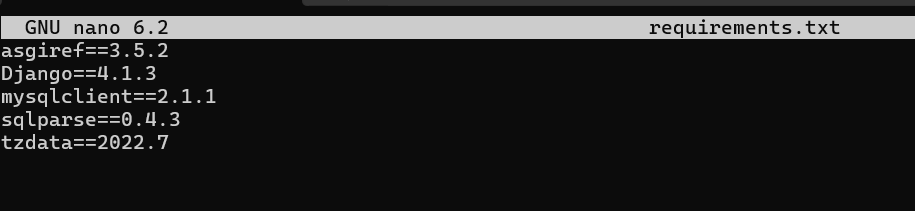
2.- Creamos un nuevo directorio donde queremos que se encuentre el contenedor. Copiamos dentro todos los recursos de nuestro proyecto de Django. Dejo un esquema básico y a modo de ejmplo de como es la estructura:



3.- Creamos un archivo Dockerfile en la raíz del proyecto (junto a manage.py):

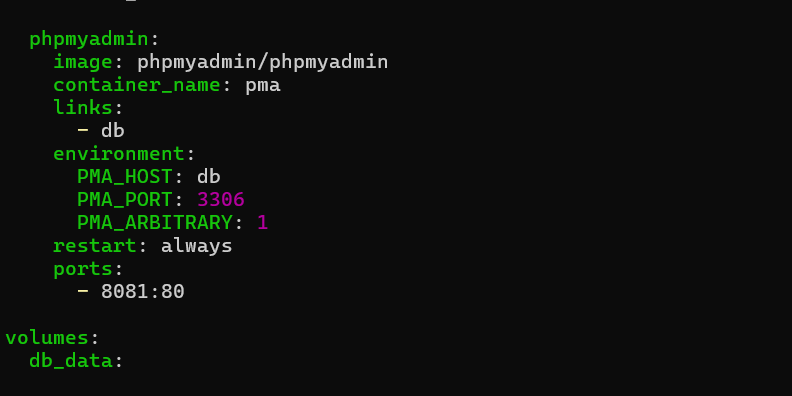
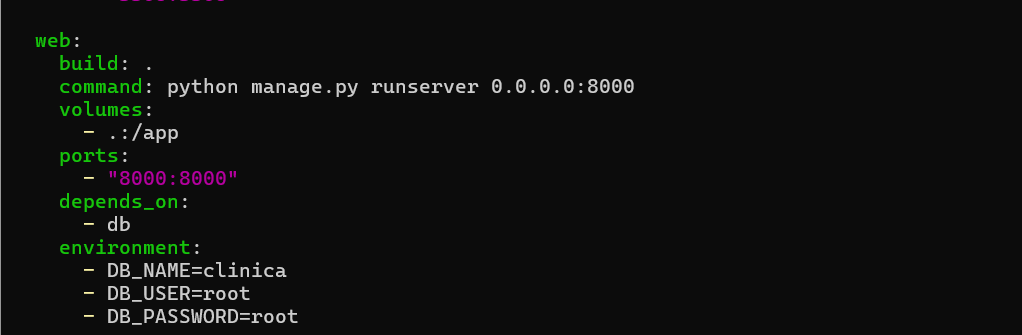
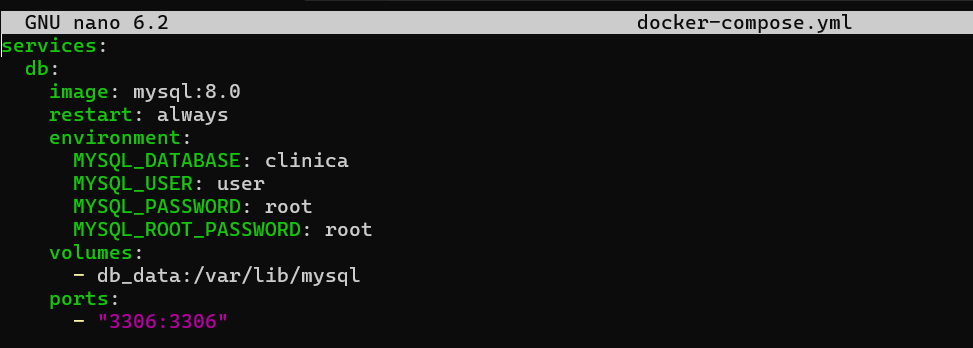


4.- Creamos el archivo requirements.txt con los siguientes elementos:

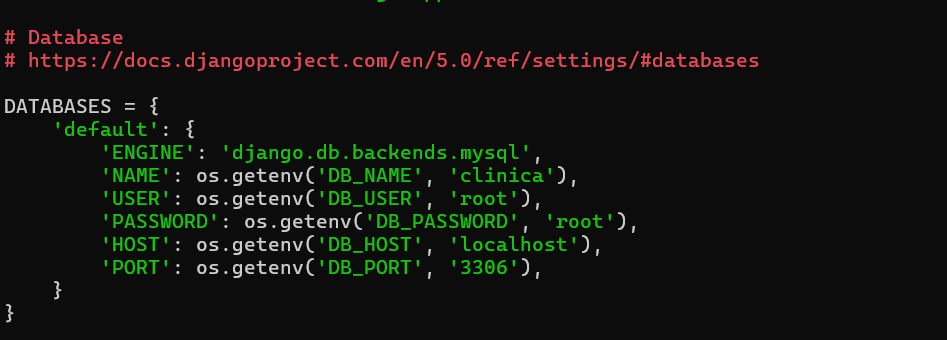


Este archivo dependera de tu contenedor y de tu proyecto de Django.

5.- Crear el docker-compose.yml en la raiz del directorio:



6.- Modificar settings.py en la carpeta del proyecto de Django para usar las variables de entorno para la configuración de la base de datos:



7.- Construir y ejecutar los contenedores:

docker-compose up –build

docker-compose up -d

1. Manual de usuario
   1. Manual de Instalación
   2. Manual de Ejecución

4- Manual de usuarios

En este apartado nos centraremos en como instalar y ejecutar los recursos de nuestro proyecto para su funcionamiento. En el anexo se encuentra una carpeta correspondiente a nuestros contenedores, un