

2023-2

DESARROLLO WEB BACK-END



ARQUITECTURA DEL SISTEMA



Janet Illescas Coria

318219309



Ulises Rodríguez García

318042202



Zurisadai Uribe García

318223197



Víctor Emiliano Cruz Hernández

318081166



Facultad de Ciencias, UNAM

Índice

Introducción	3
1.1 Objetivo del documento	3
1.2 Framework y base de datos	3
Descripción general del sistema	3
2.1 Proceso general para encargado de tableta	3
1.1.1 Asignación de mesa	3
2.1.2 Selección de platillos	4
2.1.3 Mandar orden	4
2.1.4 Votación de helado	4
2.1.5 Cierre de sesión	4
2.2 Proceso general para administrador	4
2.2.1 Login	5
2.2.2 Edición de alimentos y helados	5
2.2.3 Edición de administradores	5
2.2.4 Cierre de sesión	5
Definición de la arquitectura	5
3.1 Capas de la arquitectura	6
3.1.1 Presentación	6
Vistas	6
Controladores	6
Modelos	6
3.1.2 Lógica de Negocios	6
3.1.2 Datos	6
Arquitectura de la infraestructura	6
4.1 Cliente	6
4.1 Servidor de aplicación	7
4.1 Servidor de base de datos	7
Diagrama de despliegue	7
Diagrama de componentes	8

Introducción

1.1 Objetivo del documento

El objetivo del documento es mantener organizada la Arquitectura Técnica Organizacional y proveer una fuente de referencia para los analistas y diseñadores de la aplicación.

Los analistas y desarrolladores de la aplicación utilizarán este documento para propósitos informativos de entendimiento de la actual arquitectura, y para los propósitos del análisis para discernir si la arquitectura, como es, apoyará sus requisitos de la aplicación.

1.2 Framework y base de datos

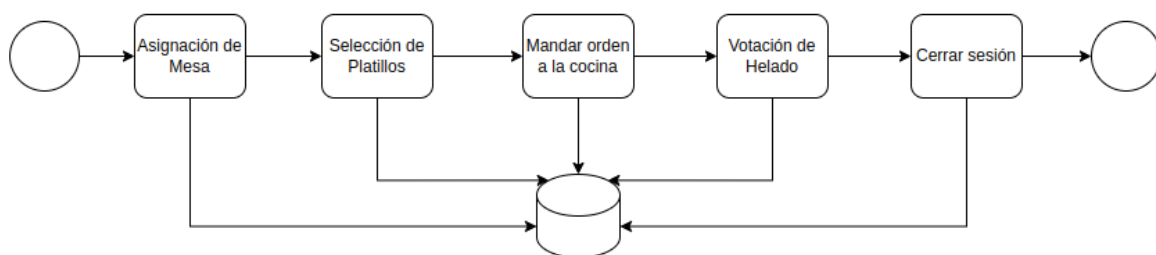
El sistema está desarrollado en [Django](#) para una flexibilidad en el desarrollo del sistema. Los datos son almacenados en una base de datos relacional implementada en [SQLite](#).

Django es un framework basado en python que incentiva el desarrollo rápido y diseño limpio de aplicaciones web. Oficialmente tiene soporte para SQLite, adicionalmente se puede formar un ecosistema de desarrollo con Apache, Nginx, Guinicorn y aplicaciones con Java EE.

Para la ejecución local del sistema se utilizarán contenedores y entornos virtuales para que pueda ser ejecutado en distintas plataformas sin necesidad de ajustarse a ellas. Utilizaremos [Docker](#) y [Conda](#) como herramientas para lograr esto.

Descripción general del sistema

2.1 Proceso general para encargado de tableta



El sistema se estará constituido por los siguientes módulos de encargado de tableta:

- Asignación de mesa
- Selección de platillos
- Mandar orden
- Votación de helado
- Cierre de sesión

1.1.1 Asignación de mesa

En este módulo el encargado de mesa asigna una tableta a una mesa con su respectiva ubicación. Se verifica que no exista otra tableta con el mismo número de mesa. El

encargado de mesa selecciona el número de mesa de una lista predefinida así como la ubicación en la que se encuentra en el restaurante.

2.1.2 Selección de platillos

Una vez iniciada la sesión este módulo se encarga de seleccionar los alimentos que el comensal escoja así como las cantidades en su respectivo carrito. En este módulo se tienen las siguientes características:

- Se cuenta con un carrito de compra
- Incorpora comunicación con la base de datos para la consulta de platillos disponibles y realizar la transferencia de selección de alimentos.
- Realiza el despliegue de los alimentos que se cuentan en la base de datos.

2.1.3 Mandar orden

En el momento que se desee, al tener el carrito con los productos deseados, se puede enviar la orden a cocina.

Antes de realizar dicha acción, se genera una lista con todos los productos del carrito que deberá ser validada por el comensal. Una vez validado se envían los datos de la mesa con la lista a cocina y vacía el carrito de compra.

Finalmente se notifica al comensal que su orden ha sido enviada exitosamente e invita a que se lleve a cabo la votación de helado (opcional).

2.1.4 Votación de helado

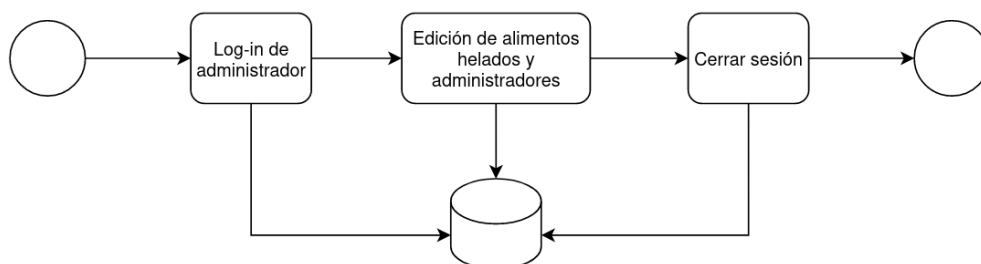
La votación de helado es la parte final del sistema para un comensal. Una vez hecha la votación del helado se asume que el comensal ya no va a pedir más alimentos. Para esto se genera una vista donde se añaden votos indefinidamente hasta que se seleccione la opción de terminar la votación. Una vez concluida, los datos del comensal vigente se eliminan de la base de datos y se entiende que la tableta queda libre para un nuevo comensal con la mesa dada.

2.1.5 Cierre de sesión

Este módulo es el encargado de eliminar todos los datos de la sesión, es decir, los productos en el carrito, votación realizada y la mesa a la cuál fue asignada la tableta.

Una vez terminada la jornada se debe cerrar la sesión de todas las tabletas.

2.2 Proceso general para administrador



El sistema se estará constituido por los siguientes módulos de administrador:

- Login

- Edición de alimentos y helados
- Edición de administradores
- Cierre de sesión

2.2.1 Login

El usuario administrador inicia sesión ingresando su nombre de usuario y contraseña, se validan que los datos sean correctos en base de datos y si es así permite el acceso a la vista de administrador.

2.2.2 Edición de alimentos y helados

En este módulo el administrador puede agregar, editar y eliminar alimentos y helados. Además de especificar la categoría a la que pertenece cada alimento (entradas, platillos, bebidas y postres).

2.2.3 Edición de administradores

Permite agregar y eliminar nuevos usuarios administradores.

2.2.4 Cierre de sesión

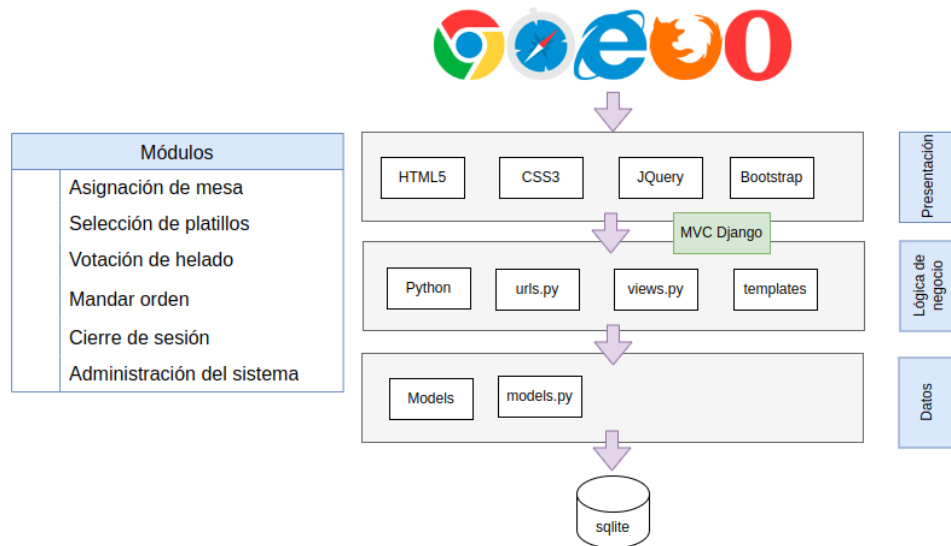
Se cierra la sesión del administrador, saliendo de la ventana de administración, impidiendo cualquier edición de alimentos y helados por otro usuario que no sea administrador .

Definición de la arquitectura

La arquitectura seleccionada para el desarrollo del sistema está desarrollada para un cliente WEB. Hacemos la distinción entre usuarios de la siguiente manera:

- Identificamos como *encargados de tabletas* a los empleados que se encargan de asignar a la tableta su número de mesa.
- Identificamos a los *comensales* como a los clientes que, una vez asignado el número de mesa a la tableta, son los que piden los alimentos que se encuentran disponibles y son los que pueden votar en la sección de votación de helado.
- Identificamos a los *administradores* como aquellos usuarios que acceden para realizar mantenimiento de la aplicación, es decir, los que agregan, eliminan o modifican los alimentos, helados y crean super usuarios.

Los usuarios dispondrán de una aplicación cliente web, que se comunicará con una aplicación de servidor la cual implementará las reglas del negocio y a su vez será la de intermediaria entre la aplicación cliente web y la base de datos.



3.1 Capas de la arquitectura

3.1.1 Presentación

Nuestra presentación se basa en el modelo MVC:

Vistas

Django utiliza *templates* para el manejo de las vistas donde este manipula los controladores necesarios para las vistas. Cada *template* es una vista al usuario de cómo se muestra la información de los datos donde Django como parte de los controladores maneja las urls por su cuenta.

Controladores

Cada vista se implementa como una función callback implementada en el archivo "views.py". Esta vista después es renderizada como un template.

Modelos

Todos los modelos se implementan en un archivo "models.py" donde esta se comunica con la base de datos manipulada totalmente por Django. Es posible visualizar y manipular estos modelos en la interfaz de superusuario que el mismo framework ofrece o en otro sistema gestor de bases de datos, en particular, [DB Browser](#).

3.1.2 Lógica de Negocios

La capa de servicios y la capa de modelo de dominio se maneja en la carpeta "cincuentaAmigos/views/" ésta se encarga de clasificar los servicios y funcionalidades de la lógica de aplicación, a saber, el despliegue del menú, los helados, así como los resultados en las votaciones y en la manipulación del carrito de compra.

3.1.2 Datos

Los datos se almacenan en una base de datos relacional en sqlite. Django encapsula la comunicación entre el acceso a los datos de forma que su administración y regulación queda determinada por el framework.

Arquitectura de la infraestructura

4.1 Cliente

Sistema Operativo	Multiplataforma
Navegador	Chrome, Firefox, Internet Explorer, Brave, entre otros
Tecnologías utilizadas	HTML5, CSS3, JQuery, Bootstrap
Lenguajes utilizados	Javascript

4.1 Servidor de aplicación

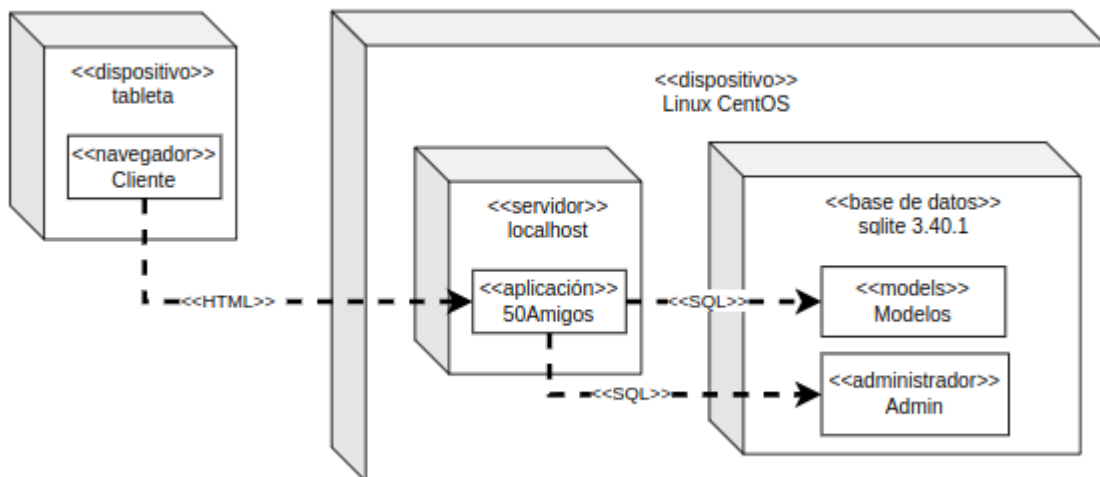
Sistema Operativo	Multiplataforma
Application Server	No definido
Lenguajes utilizadas	python3
Frameworks utilizados	Django

4.1 Servidor de base de datos

Sistema Operativo	Multiplataforma
Base de Datos	sqlite 3.40.1

Diagrama de despliegue

A continuación se muestra el diagrama de Despliegue de la aplicación:



Los dispositivos físicos se presentan del lado del Cliente y del Servidor. Cada nodo es un elemento donde se ejecutan los componentes del sistema, entre ellos las aplicaciones, bibliotecas y el motor de la base de datos.

Diagrama de componentes

El siguiente diagrama muestra las componentes en que está dividida la funcionalidad del sistema, los conectores identifican la relación entre los componentes.

Los módulos de la aplicación se encuentran contenidos en el bloque de “50Amigos”.

