

PROYECTO FINAL

Entrega I



[Autor] | 04/04/2024



Contenido

Entrega I.....	1
Descripcion	3
Aplicaciones similares	3
Stack tecnologico.....	3
Objetivos y Requisitos.....	4
Modelo DB.....	6
Dishes	6
Restaurants.....	7
Roles	7
Users.....	7
Users_roles	7
.....	



Descripcion

Este proyecto consiste en una aplicación que se encarga un registro de platos de un restaurante en función de diario de modo que tengamos un sistemas de reseñas personal.

La aplicación sería un registro de locales y productos consumidos en el mismo con valoración, notas, precio, tiempo de espera intentando recabar el máximo de datos posibles sin excederse para evitar el consumo excesivo del tiempo del usuario, que pudiera ocasionar una de entrada de datos demasiado voluminosa, ya que al ser de carácter general en horario de ocio, el usuario solo querrá destinar un tiempo mínimo para registrar la experiencia ocurrida en ese momento, así mismo, también si es en horario laboral los tiempos se reducen aun mas por lo que es inevitable que la entrada de datos requerida sea mínima e indispensable para poder recabar datos para una futura consulta sin ser demasiados escuetos con la misma.

En el apartado grafica se ha decantado con una interfaz inspirada en aplicaciones de uso común como puede ser YouTube o Instagram amigable para el usuario, familiar, evitando fatiga visual e intentando ser lo mas intuitiva para el usuario final de modo que sin uso de ningún tipo de tutorial sea capaz de usarse debido a experiencias previas con aplicaciones mundialmente usuales.

Aplicaciones similares

En cuanto aplicaciones similares, hay un variado de aplicaciones con sistemas similares, como puede ser reseñas de Google, TripAdvisor, the fork, como otras , aunque el enfoque es diferente dado que el método de reseña es mas comunitario, pudiendo observar reseñas de restaurantes no de platos individuales ni pudiendo tener tus reseñas juntas y ordenadas desde el punto de vista de este proyecto. Por lo que resumidamente, hay aplicaciones similares pero sin el enfoque de esta.

Stack tecnologico.

Para realización de la app el stack tecnológico elegido es:

- Lenguaje de programación: Se usará Java tanto para el desarrollo en la aplicación móvil en Android como también para la API mediante el Framework Spring Boot.
- Base de datos: Se usará MySQL debido a la gran compatibilidad con Android mediante conectores JDBC, también por su facilidad de uso con PHPMyAdmin
- JWT: JSON Web Token se usará para la autenticación y autorización de usuarios, con este sistema se generará un token seguro tras el inicio de sesión del usuario, así no se gestionan las sesiones abiertas en el servidor mejorando el rendimiento de este.
- Picasso: Para la subida y visualización de imágenes, se ha optado por la librería Picasso ya que permite mostrar tanto imágenes de url como de rutas del servidor.
- Jakarta Servlet: Se usará como base para manejar las peticiones HTTP, también permite implementar controladores y filtros para la seguridad de estas, siendo la evolución de javax.servlet.
- TLS/SSL: Para la seguridad de la transmisión de datos como tokens y contraseñas se habilitará el soporte para conexiones https mediante protocolo TLS/SSL, así evitar ataques de capturas de paquetes en la transmisión de estos.

Objetivos y Requisitos

La aplicación tendrá 5 apartados principales, sin contar paneles de configuración o de administradores:

- Feed: Un apartado el que los usuarios podrán acceder a una lista de tarjetas de restaurantes con imágenes representativas o imagen de la misma carta pudiéndose visualizar ampliándose si pulsamos en la misma como también veremos nota media, precio medio, tiempo de espera medio, solo datos resumidos y relevantes, también podremos pulsar en la tarjeta donde se nos abrirá una ventana donde podremos ver todas los platos añadidos con anterioridad en ese restaurante.
- Búsqueda: Un apartado el que los usuarios podrán acceder a una lista de tarjetas de platos de todos los restaurantes pudiendo visualizar los datos de los platos que se han añadido pudiendo asignar por ejemplo si son platos favoritos
- Añadir platos: En este apartado los usuarios añadirán el plato vinculándolo al restaurante, proporcionando un nombre del plato, imagen, nota, valoración, precio y minutos de espera
- Mapa: Sera un mapa geográfico donde se cargarán todos los restaurantes visitados para poder abrir Google maps en la dirección del mismo así poder navegar hasta él.
- Favoritos: : Un apartado el que los usuarios podrán acceder a una lista de tarjetas de platos marcados como favoritos de todos los restaurantes con la información anteriormente menciona

REQUISITOS FUNCIONALES	REQUISITOS NO FUNCIONALES
La APP debe mostrar todos los restaurantes y platos	Los restaurantes solo podrán modificarse por administradores, tendrán un apartado específico para esto
La app debe poder almacenar los datos de restaurantes, platos y usuarios	Los platos solo podrán modificarse por los autores de las publicaciones, se habilitará un botón en la tarjeta del plato
La app debe poder asignar favoritos a los platos	La autenticación estará protegida por SSL y token
La app debe poder modificar datos de la cuenta del usuario	Existirán filtros de solicitudes a la API
La app debe mostrar un mapa con los restaurantes visitados	Se diferenciarán a los diferentes tipos de usuarios por roles, siendo asignado el rol usuario automáticamente y rol administrador por otros administradores en su sección habilitada

Administradores

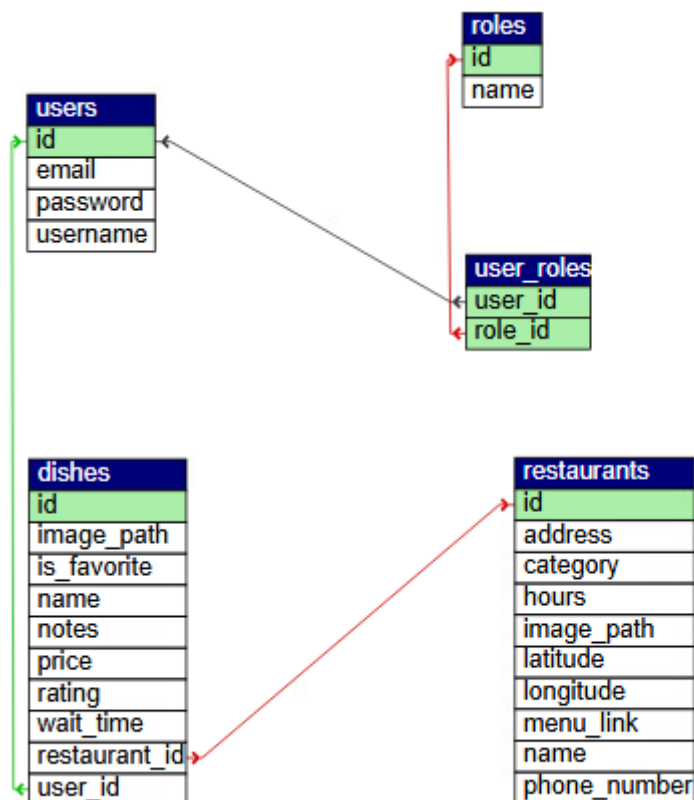
Los administradores tienen una función clara y sencilla, aparte de las funciones de los usuarios normales, tienen la tarea mas importante que es añadir los restaurantes y modificarlos, nunca jamás un usuario puede realizar estas tareas, así evitamos que los usuarios puedan duplicar un restaurante, será otorgado el rol de administrador a usuarios escogidos específicamente con el deber de comprobar y evitar las duplicaciones de los restaurantes añadidos como también la verificación de los datos de estos.

Cuando un usuario se crea automáticamente tiene el rol de usuario, sin embargo, los administradores pueden otorgar el permiso de administradores a otros usuarios.

Caso de uso

Descripción	El sistema debe de comportarse de la siguiente manera para poder subir un plato
Precondición	El usuario tiene que disponer de un nombre de usuario y una contraseña para poder acceder y deberá de tener asignado un rol
Secuencia Normal	<ol style="list-style-type: none">1- El usuario accede al sistema mediante autenticación2- El sistema autentifica al usuario y le concede el acceso3- El usuario selecciona el botón de añadir plato4- El usuario selecciona restaurante e ingresa foto, nombre, minutos, precio, nota y valoración.5- Pulsa el botón de subir6- El sistema lo devuelve a la pantalla principal
Excepciones	<ol style="list-style-type: none">2- El usuario introduce nombre de usuario o contraseña incorrectos. Ir al paso 16- El sistema solicita completar los campos requeridos. Ir al paso 4.

Modelo DB



Dishes

Columna	Tipo	Atributos	Null	Predeterminado	Extra	Enlaces a
id	bigint(20)		No		auto_increment	
image_path	varchar(255)		Sí	NULL		
is_favorite	bit(1)		Sí	NULL		
name	varchar(255)		Sí	NULL		
notes	varchar(255)		Sí	NULL		
price	double		Sí	NULL		
rating	int(11)		Sí	NULL		
wait_time	int(11)		Sí	NULL		
restaurant_id	bigint(20)		No			-> restaurants.id ON UPDATE RESTRICT ON DELETE RESTRICT
user_id	bigint(20)		No			-> users.id ON UPDATE RESTRICT ON DELETE RESTRICT

Restaurants

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a
id	bigint(20)		No		auto_increment	
address	varchar(255)		Sí	NULL		
category	varchar(255)		Sí	NULL		
hours	varchar(255)		Sí	NULL		
image_path	varchar(255)		Sí	NULL		
latitude	double		Sí	NULL		
longitude	double		Sí	NULL		
menu_link	varchar(255)		Sí	NULL		
name	varchar(255)		Sí	NULL		
phone_number	varchar(255)		Sí	NULL		

Roles

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a
id	int(11)		No		auto_increment	
name	enum('ROLE_ADMIN', 'ROLE_MODERATOR', 'ROLE_USER')		Sí	NULL		

Users

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a
id	bigint(20)		No		auto_increment	
email	varchar(255)		Sí	NULL		
password	varchar(255)		Sí	NULL		
username	varchar(255)		Sí	NULL		

Users_roles

Columna	Tipo	Atributos	Null	Predefinido	Extra	Enlaces a
user_id	bigint(20)		No			-> users.id ON UPDATE RESTRICT ON DELETE RESTRICT
role_id	int(11)		No			-> roles.id ON UPDATE RESTRICT ON DELETE RESTRICT