

Proyecto #1

Desarrollo Web Backend

Desarrollen el backend para un servicio de domicilios.

- La API debe crearse como repositorio en Github y debe ser "entregada" al profesor utilizando el enlace, donde se enviará únicamente el enlace al repositorio.
- El proyecto se desarrolla de manera individual, con fecha de entrega Domingo, 30 de abril a las 23:59.

Los endpoints a desarrollar incluyen, pero no están limitados a:

CRUD de usuarios

| Método | Comportamiento Esperado |
|-----------------|--|
| Create | El endpoint crea un usuario en la base de datos con los datos enviados al backend. |
| Read (unidad) | El endpoint retorna los datos del usuario que corresponden a las credenciales (correo y contraseña) o id proveída. |
| Read (cantidad) | N/A |
| Update | El endpoint modifica los datos del usuario que corresponde a la id proveída, usando los datos proveídos. |
| Delete | El endpoint "inhabilita" un usuario que corresponde a la id proveída. |

CRUD de restaurantes

| Método | Comportamiento Esperado |
|-----------------|--|
| Create | El endpoint crea un restaurante en la base de datos con los datos enviados al backend. |
| Read (unidad) | El endpoint retorna los datos del restaurante que corresponde a la id proveída. |
| Read (cantidad) | El endpoint retorna los datos de los restaurantes que correspondan a la categoría proveída y/o su nombre se asemeje a la búsqueda. |
| Update | El endpoint modifica los datos del restaurante que corresponde a la id proveída, usando los datos proveídos. |
| Delete | El endpoint "inhabilita" un restaurante que corresponde a la id proveída. |

CRUD de productos

| Método | Comportamiento Esperado |
|-----------------|---|
| Create | El endpoint crea un producto de un restaurante en la base de datos con los datos enviados al backend. |
| Read (unidad) | El endpoint retorna los datos del producto que corresponde a la id proveída. |
| Read (cantidad) | El endpoint retorna los datos de los productos que correspondan a el restaurante y/o categoría proveída. |
| Update | El endpoint modifica los datos del producto que corresponde a la id proveída, usando los datos proveídos. |
| Delete | El endpoint "inhabilita" un producto que corresponde a la id proveída. |

CRUD de pedidos

| Método | Comportamiento Esperado |
|-----------------|--|
| Create | El endpoint crea un pedido de un usuario a un restaurante en la base de datos con los datos enviados al backend. |
| Read (unidad) | El endpoint retorna los datos del pedido que corresponde a la id proveída. |
| Read (cantidad) | El endpoint retorna los datos de los pedidos realizados por el usuario proveído, enviados por el usuario proveído, pedidos a un restaurante proveído, y/o entre las fechas proveídas. |
| Read (cantidad) | El endpoint retorna los datos de los pedidos enviados, pero sin aceptar. |
| Update | El endpoint modifica los datos del pedido que corresponde a la id proveída, usando los datos proveídos, a menos que este ya haya sido enviado. |
| Delete | El endpoint "inhabilita" un producto que corresponde a la id proveída. |

Flujo de aplicación (cliente)

1. Un usuario abre la aplicación.
2. El usuario realiza su registro usando correo electrónico, nombre(s), contraseña, numero de celular, dirección, y eligiendo la opción "cliente".
3. El usuario inicia sesión con su correo electrónico y contraseña.
4. El usuario busca el restaurante al que desea pedir, sea usando los filtros de categoría, realizando una búsqueda del nombre del restaurante, ambas o ninguna de estas. Los restaurantes se encuentran ordenados según la popularidad del restaurante.
5. El usuario entra a la pagina del restaurante donde encuentra los productos separados por las categorías del restaurante.
6. El usuario crea su pedido agregando los productos que desea y las cantidades que desea.
7. El usuario termina de crear su pedido, y se dirige a enviar el pedido, donde encuentra un resumen de todo el pedido, incluyendo el total monetario a pagar por todos los productos.
8. El usuario envía el pedido.
9. El usuario es informado cuando su pedido es aceptado por otro usuario, quien le llevara el pedido.
10. El usuario es informado cuando el segundo usuario recibe el pedido en el restaurante.
11. El usuario es informado cuando el segundo usuario llegó a la dirección de envío.
12. El usuario es informado cuando el segundo usuario entregó el pedido.

Flujo de aplicación (domiciliario)

1. Un usuario abre la aplicación.
2. El usuario realiza su registro usando correo electrónico, nombre(s), contraseña, numero de celular, dirección, y eligiendo la opción "domiciliario".
3. El usuario inicia sesión con su correo electrónico y contraseña.
4. El usuario encuentra los pedidos enviados por otros usuarios, con opción de ordenarlos según distancia del usuario al restaurante, distancia del restaurante al cliente, o vez del pedido.
5. El usuario selecciona un pedido y lo acepta.
6. El usuario se dirige al restaurante.
7. El usuario espera los productos del pedido.
8. El usuario recibe todos los productos.
9. El usuario se dirige a la dirección del cliente y marca el pedido como en dirección.
10. El usuario realiza la entrega del pedido, y marca el pedido como realizado.

Flujo de aplicación (administrador)

1. Un usuario abre la aplicación.
2. El usuario realiza su registro usando correo electrónico, nombre(s), contraseña, numero de celular, dirección, y eligiendo la opción "administrador de restaurante".
3. El usuario inicia sesión con su correo electrónico y contraseña.
4. El usuario realiza el registro de un restaurante usando nombre, dirección, y categoría.
5. El usuario agrega productos al inventario del restaurante usando nombre, descripción, categoría, y precio.
6. Adicionalmente, puede ver el inventario del restaurante, y agregar/modificar/eliminar los productos.
7. El usuario puede ver los pedidos en curso del restaurante, o los pedidos realizados este día/semana/mes.
8. El usuario, al recibir un pedido, prepara los productos, los entrega al domiciliario y marca el pedido como recibido.

Estados de un pedido

| Palabra clave | Descripción |
|---------------|---|
| Creado | Un usuario esta modificando los productos/cantidades del pedido. |
| Enviado | Un usuario terminó el pedido, los productos o cantidades del pedido no se puede modificar en este estado. Un usuario puede aceptar el pedido. |
| Aceptado | Un usuario aceptó el pedido, y ahora está vinculado a este. |
| Recibido | Un usuario recibió los productos del pedido. |
| En dirección | Un usuario llegó a la dirección de entrega del pedido. |
| Realizado | Un usuario hizo la entrega de los productos de un pedido. |

Requisitos Funcionales

1. Todos los campos de registro de usuario son requeridos.
2. La popularidad de un restaurante está determinada por la cantidad de domicilios realizados por el restaurante.
3. Cada restaurante determina sus propias categorías.
4. El cliente no debe realizar ningún calculo, computo, ni filtración de datos según criterios o categorías, esto es responsabilidad del servidor.
5. Los recursos inhabilitados no deben ser retornados ni modificados con ningún request.
6. Los informes/notificaciones son responsabilidad del frontend, pero el backend debe tener la información necesaria para saber si hay algo que informar o notificar.