

# Exámen Final

## Descripción

Desarrollar una aplicación que permita tomar los pedidos de los clientes de un café

## Requerimientos funcionales

- El backend de esta aplicación ya está programado, debemos desarrollar sólo el front.
- El acceso a ciertas partes de la aplicación debe estar protegido por usuario y contraseña
- Deben existir tres tipos de usuarios: guest, user y admin
  - El **guest** es un usuario sin acceder. Sólo puede ver el menú.
  - El **user** puede hacer lo mismo que **guest**, además de tomar pedidos e ingresarlos
  - El **admin** puede hacer lo mismo que el **user**, además de ver los user y admin registrados
- La aplicación frontend se compone de las siguientes vistas:
  - **Página de inicio:** Menú de productos y sus valores. Contiene un formulario de ingreso por usuario y contraseña
  - **Toma de pedidos:** Listado de productos donde podemos señalar la cantidad que el cliente desea de cada uno. Al final del listado indica el precio total que se actualiza cada vez que agregamos o quitamos un producto.
  - **Vista de usuarios:** Listado de usuarios tipo user y admin. Se muestra fotografía, nombre y rol.

# Diseño y vistas

## Mockup



El mockup es sólo una guía, no es necesario que sean usados los mismos estilos. En el siguiente enlace pueden verlo en Figma, herramienta utilizada por la mayoría de diseñadores y UX/UI.

<https://www.figma.com/file/VPKBhy0R54Aw7ww9RPVFUL/Mockup-para-Entrega-Final-Gen6-Javascript-Master?node-id=2380%3A31>

## Requerimientos de diseño

Los requerimientos de diseño son obligatorios.

### Menú



- Se deben mostrar las imágenes y nombres de productos. Máximo 3 productos por sección del menú, con la posibilidad de clicar un botón u enlace que contenga el texto “Ver más”, el cual nos mostrará el resto de la carta de esa sección en particular.
- Debe tener un enlace al inicio de sesión

### Inicio de sesión



- Permitirá un máximo de 3 intentos de contraseña. Al 3to impedirá el acceso durante 15 minutos. Este bloqueo será solo en el frontend.
- Una vez el usuario se logue exitosamente, será redirigido a Órdenes

### Órdenes o pedidos

- Se mostrará un listado de los pedidos, con la imagen y nombre de la persona que tomó el pedido, además de la fecha de creación de este y su ID.
- Se debe facilitar un formulario para la creación de pedidos. Este constará de un combobox para indicar la mesa **que sólo mostrará las disponibles**, uno o más combobox con el listado de productos, acompañados con un input que permita indicar el número que desea de dicho producto. Además, tendrá dos botones, uno para agregar nuevas líneas al pedido, y otro para enviar el pedido a backend.

### Usuarios

- Se mostrará nombre, correo, teléfono, rol y fecha de cumpleaños (en formato humano), acompañados de la fotografía del usuario.

## Requerimientos de entrega

- La entrega se debe realizar mediante classroom, ya sea un fichero .zip conteniendo los archivos del proyecto o un link a un repositorio.
- Se tienen 4 semanas de plazo para entrega, finalizando este período el 11 de Julio de 2022.

Una vez concluído el periodo de entrega, en un plazo máximo de 2 semanas se entregarán

las evaluaciones mediante classroom, finalizando este periodo el 1 de Agosto del 2022.

# Documentación API

**Base URL:** <https://maf2qxs1f6.execute-api.us-east-1.amazonaws.com/prod>

## Endpoints

/oauth/token

Métodos permitidos: POST

Endpoint destinado a conseguir los tokens necesarios para utilizar la aplicación.

/api/users

Métodos permitidos: GET

Endpoint que entrega una lista de los usuarios registrados en la aplicación, solo es visible para usuarios con el rol **admin**.

Ejemplo de respuesta

```
[
  {
    "id": 1,
    "name": "Madonna",
    "username": "madonna",
    "birthday": -359078400000,
    "email": "madonna@restaurant.cl",
    "phone": "+56 9 68615452",
    "roles": ["admin"],
    "img": "https://ramosmerino.cl/gen6/madonna.jpg"
  }
  ...
]
```

/api/menus

Métodos permitidos: GET

Endpoint que entrega un listado de platos disponibles con sus respectivos precios. Es accesible por cualquier usuario.

Ejemplo de respuesta

```
{
```

```

    "entradas": [ // nombre de sección
      {
        "id": 1, // id de producto
        "name": "Papas Bravas", // nombre producto
        "price": "5400", // precio producto
        "description": "Porción de papas.", // descripción producto
        "img": "https://ramosmerino.cl/gen6/papas-bravas.jpg" // imagen
      }
    ]
  }
  ...

```

## /api/orders

Métodos permitidos: GET, PUT

Endpoint utilizado para tomar órdenes y ver las que ya existen. Es accesible por usuarios con el rol **user**.

### Ejemplo de respuesta

```

[
  {
    "id": 326, // id de pedido
    "table": 4, // id de mesa
    "created_at": 1655184271000, // epoch de creación de pedido
    "waiter": 3, // id de usuario/trabajador
    "order": [ // orden
      {
        "product": 1, // id de producto
        "quantity": 2 // cantidad
      },
      {
        "product": 9,
        "quantity": 2
      },
      {
        "product": 28,
        "quantity": 1
      },
      {
        "product": 30,
        "quantity": 1
      }
    ]
  }
]
...

```

/api/tables

Métodos permitidos: GET

Endpoint utilizado para obtener las mesas. Es accesible por usuarios con el rol **user**.

Ejemplo de respuesta

```
[{
  "id": 1, // id de mesa
  "name": "Mesa 1", // nombre de la mesa
  "available": true // disponibilidad de la mesa
}
...]
```

## Usuarios y contraseñas

Administrador

**Usuario:** madonna

**Contraseña:** mad0nna

Usuarios

**Usuario:** dualipa

**Contraseña:** du4321

**Usuario:** britney

**Contraseña:** britNEY