

COOK-IT-APP

• Explicación de la aplicación.

Se trata de un restaurante que tiene camareros y cocineros de forma que se sirven platos que se registran en una comanda y llegan al cocinero. Existe un almacén de ingredientes a partir de los cuales se hacen los platos.

Esta aplicación está basada en un restaurante y primeramente se compone de una ventana de inicio y una ventana de inicio de sesión.

El restaurante está formado por seis mesas y cada mesa puede tener como máximo seis comensales.

La ventana inicial está formada por una imagen de fondo, el nombre del restaurante “El Rincón de la Comida” y un botón de “Entrar” que permite el acceso inicial a la web y da paso a la ventana de login.

Para la implementación de dicha ventana, se ha utilizado un archivo perteneciente al frontend de la aplicación llamado **entrada.js**.

<pre><Typography component="h1" variant="h3" className="titulo" color="white" sx={{ mt: 3, mb: 2, marginTop: 20, }} > Restaurante "El Rincón de la Comida" </Typography></pre>	<pre><Button type="submit" fullWidth variant="contained" sx={{ mt: 3, mb: 2, marginTop: 5, }} > Entrar </Button></pre>
---	--

De la misma manera, para la ventana de login se ha utilizado un archivo en la misma ubicación que el anteriormente mencionado y llamado **login.js** donde se introducen los datos del usuario que quiere iniciar sesión como se muestra a continuación.

Usuario

Contraseña

```
<TextField
  margin="normal"
  required
  fullWidth
  name="password"
  label="Contraseña"
  type="password"
  id="password"
```

```

<TextField
    margin="normal"
    required
    fullWidth
    id="email"
    label="Usuario"
    name="email"
    autoComplete="email"
    autoFocus
  />

```

La ventana de inicio de sesión está formada por un título “Acceso de usuario”, dos cajas donde se debe indicar el nombre de usuario y la contraseña y dos botones, “Acceder” y “Atrás” por si el usuario requiere retroceder al paso anterior.

Botón Acceder

```

<Button
    type="submit"
    fullWidth
    variant="contained"
    sx={{ mt: 3, mb: 2 }}
  >
    Acceder
</Button>

```

Botón Atrás

```

<Button
    type="submit"
    fullWidth
    variant="contained"
    sx={{ mt: 3, mb: 2 }}
    onClick={()=>navigate("/")}
  >
    Atrás
  </Button>

```

Existen tres tipos de usuarios, los camareros, los cocineros y el usuario de tipo administrador.

A continuación se detallan las acciones que realiza cada uno de los usuarios y el código que las implementa.

Una vez iniciada la sesión, se abre una ventana que representa el comedor y donde se muestran las seis mesas con sus respectivos números identificativos y de color verde si estan libres y rojo si estan ocupadas. Además cada mesa en la parte inferior tiene dos botones, uno para tomar una comanda y otro para cambiar una mesa ocupada a libre.

Si se accede al botón inferior izquierdo de cada mesa “Tomar comanda” se abre una ventana en la que se indica el número de la mesa elegida y donde se debe introducir en

primer lugar el número de comensales de dicha mesa y posteriormente el usuario de tipo camarero deberá introducir los platos que hayan decidido los comensales.

Una vez que se termina de registrar todos los platos, el camarero debe hacer click en el botón inferior “Terminar comanda” y los platos se registran en la base de datos.

Una vez se haya tomado la comanda, el camarero podrá acceder a un botón que aparecerá en la mesa “Ver comanda”.

Botón Terminar comanda

```
<Button
  type="button"
  fullWidth
  variant="contained"
  sx={{ mt: 3, mb: 2 }}
  onClick={() => {
    subeComanda();
  }}
>
  Terminar Comanda
</Button>
```

Hacer una comanda

```
<Box sx={{ bgcolor: 'background.paper', pt: 8, pb: 6 }}>
  <Container maxWidth="md">
    <Typography
      component="h1"
      variant="h2"
      align="center"
      color="text.primary"
      gutterBottom
    >
      COMANDA MESA {mesa}
    </Typography>
    <Box component="form" /* onSubmit={handleSubmit}*/ noValidate sx={{ mt: 1 }}>
      <Autocomplete
        options={personas}
        autoHighlight
        style={{ width: 300 }}
        value={numeroPersonas}
        inputValue={numeroPersonas}
        onChange={(event, newInputValue) => {
          setNumeroPersonas(newInputValue);
          comensales(newInputValue);
        }}
        renderInput={(params) => <TextField {...params} label="Comensales"/>}
      />
    </Box>
  </Container>
</Box>
```

Página 10 de 10

Se puede deducir que el camarero se encarga de tomar, ver y eliminar las comandas de las mesas como así lo indican las funciones implementadas en el archivo **camarero.js** del frontend.

Tomar una comanda

```
function tomarComanda(nMesa){
  sessionStorage.setItem("mesa", nMesa);
  navigate('/'+camarero+'/comanda/'+nMesa);
}

    3);
}
```

Eliminar una comanda

```
async function borrarComanda(nMesa){
  axios.post('http://localhost:3053/'+camarero+'/liberar', {
    mesa: nMesa,
  }).then(response =>{
    alert("Mesa " + nMesa + " queda libre");
  });
}
```

Desde la ventana del comedor mencionada unos párrafos atrás, se puede acceder a un botón en la parte superior llamado “Administrador” y a “Salir”.

El botón de salir lleva al usuario que acceda a él a la ventana inicial.

Si accedemos a “Administrador” aparece una ventana donde hay un encabezado “Administrador” y tres botones para añadir ingrediente, añadir plato y añadir usuario.

En el archivo Admin.js se implementan dichos botones como se muestra a continuación.

Botón Añadir ingrediente

```
<Button
  type="submit"
  fullWidth
  variant="contained"
  sx={{ mt: 3,
  |onClick={()=>navigate("/user/admin/nuevoIngrediente")}
  >
  Añadir ingrediente
</Button>
```

Botón Añadir Plato

```
<Button
  type="submit"
  fullWidth
  variant="contained"
  sx={{ mt: 3, mb: 2 }}
  |onClick={()=>navigate("/user/admin/nuevoPlato")}
  >
  Añadir plato
</Button>
```

```

<Button
    type="submit"
    fullWidth
    variant="contained"
    sx={{ mt: 3, mb: 2 }}
    onClick={()=>navigate("/user/admin/nuevoUser")}
>
    Añadir usuario
</Button>

```

• Tecnologías.

Para la realización de este proyecto hemos utilizado las siguientes tecnologías:

- Visual Studio Code como entorno de desarrollo de la práctica.
- Tecnologías utilizadas del lenguaje de programación:
 - React.js propone una arquitectura basada en componentes o piezas de código, lo que llamamos Snippets (pequeñas partes reutilizable de código fuente), en las que se utiliza HTML, CSS y Javascript, de modo que contienen tanto la lógica como la presentación.
 - La utilización de Node.js en el backend que es un motor de plantillas. De esa manera, es el servidor con Node.js quien interpreta React y entrega al navegador HTML, CSS y JavaScript como en un sitio web tradicional.
 - Se utiliza Axios para la comunicación. Es una librería Javascript capaz de ejecutarse tanto en el frontend como en el backend, que facilita todo tipo de operaciones como cliente HTTP. Con Axios se pueden realizar solicitudes contra un servidor o consumo de APIs, completamente configurables, y recibir la respuesta de una manera sencilla.
 - Express es un marco de aplicación web de Node js que proporciona amplias funciones para crear aplicaciones web y móviles. Es utilizado para crear una aplicación web híbrida, de varias páginas y de una sola página. Es una capa construida en la parte superior de Node js que ayuda a administrar servidores y rutas.
 - React Router es una biblioteca estándar para el routing en React que permite la navegación entre vistas de varios componentes, permite cambiar la URL del navegador y mantiene la interfaz de usuario sincronizada con la URL. El routing es cómo los puntos finales de una aplicación (URI) responden a las solicitudes de los clientes. Se define el routing usando métodos del objeto de aplicación Express que corresponden a métodos HTTP; por ejemplo, aplicación. get() para manejar solicitudes GET y app. post para manejar las solicitudes POST.
- Utilizamos Photoshop para la creación de las imágenes que se pueden ver dentro de la aplicación cuando un camarero va a realizar una comanda.
- Heroku es una plataforma de servicios en la nube que nos permite manejar los servidores y sus configuraciones, escalamiento y la administración. Como consecuencia se puede lanzar la aplicación sin tener complicaciones de

infraestructura, administrar servidores, las bases de datos y la seguridad que estos deben de tener entre otras cosas.

- MongoDB es una base de datos orientada a documentos, la cual guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON. MongoDB es un sistema para la gestión de datos NoSQL o no relacional.

Github es un repositorio online gratuito que nos permite gestionar proyectos y controlar versiones de código, subiendo nuestros repositorios de código para almacenarlo en el sistema de control de versiones Git.

● Diagrama de arquitecturas

El diagrama de arquitecturas que se muestra en esta sección es una representación de alto nivel del frontend y el backend de la aplicación Cook.It-App.

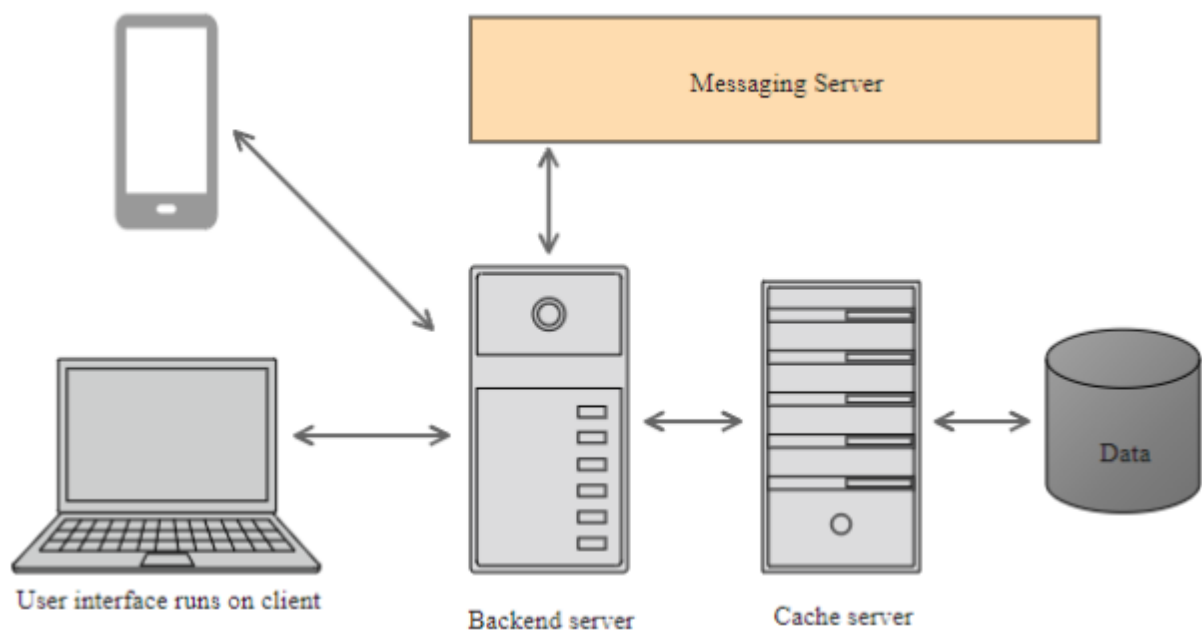
Este diagrama aunque no es muy grande pero proporciona información útil y práctica sobre cómo se ha desarrollado el sitio web de dicho restaurante.

La aplicación web Cook-It-App igual que todas las aplicaciones de base de datos basadas en la web tiene tres componentes primarios, un navegador web (cliente), un servidor de aplicaciones web y un servidor de bases de datos.

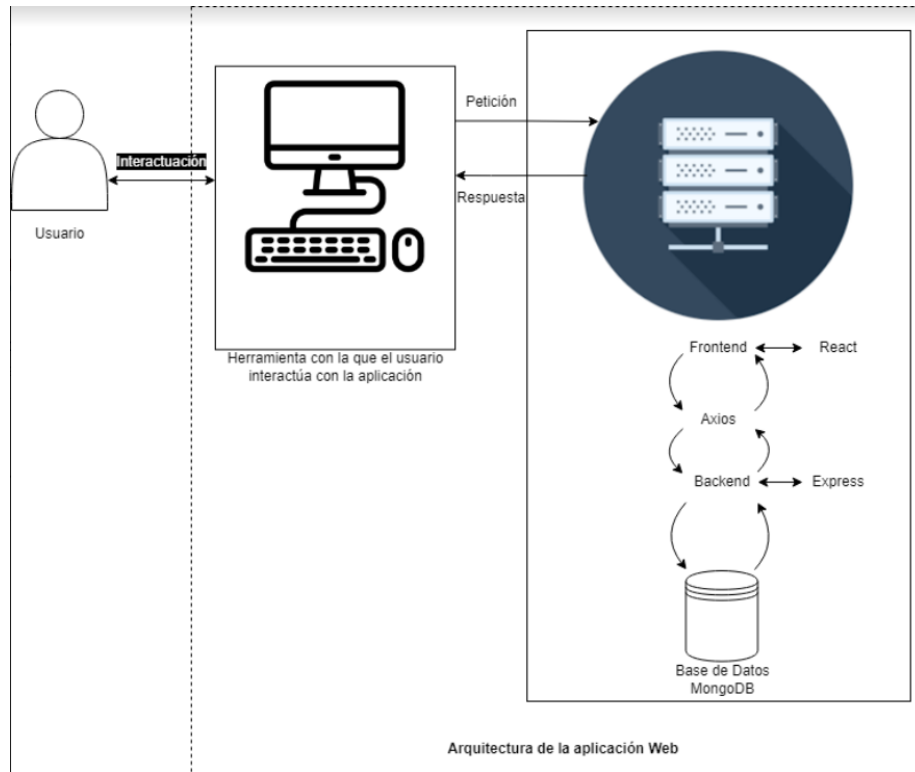
La aplicación web es proporcionada por un servidor Web y se puede utilizar en cualquier punto vía clientes web (navegadores).

El servidor web distribuye las páginas de información a los clientes que la solicitan.

Los requisitos son hechos a través de la conexión de red y para ello se utiliza el protocolo HTTP.

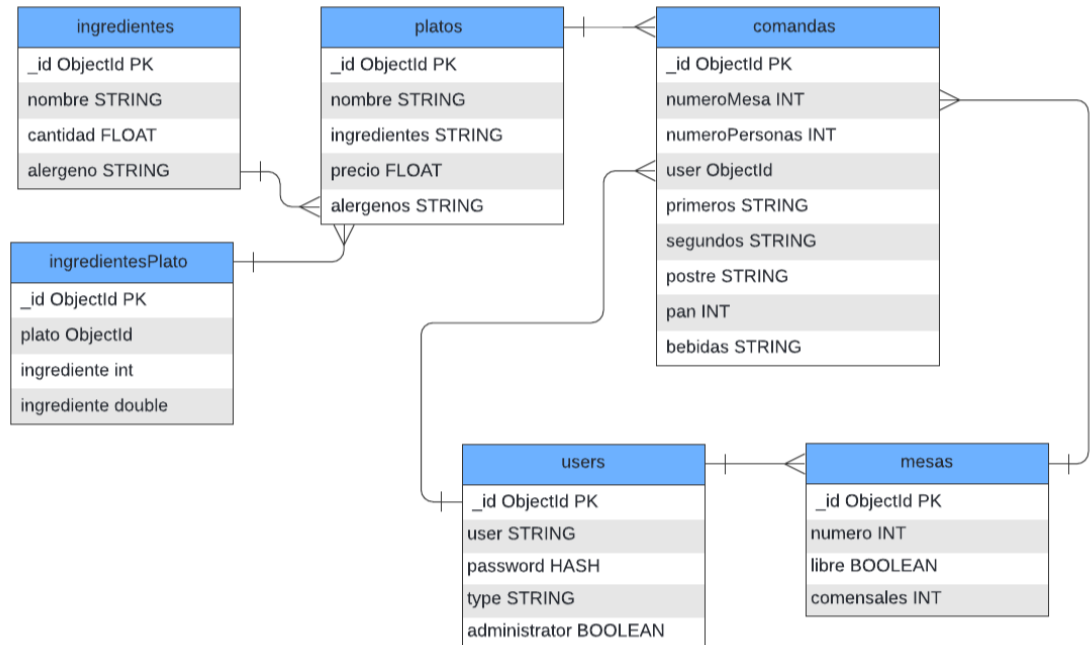


El componente principal de la aplicación web es el servidor que es desde donde el backend envía la información que termina en la base de datos. Desde el punto de vista de la interfaz de usuario, es decir, del lado del cliente, la aplicación web muestra la información que recibe del backend.

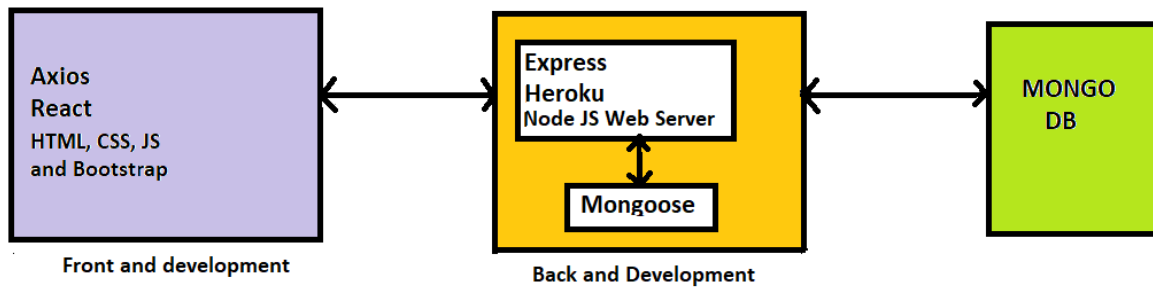


- **Diagrama de modelo de la base de datos**

La base de datos de Cook-It-App está formada por 6 tablas: ingredientes, ingredientes Plato, platos, users, mesas y comandas.

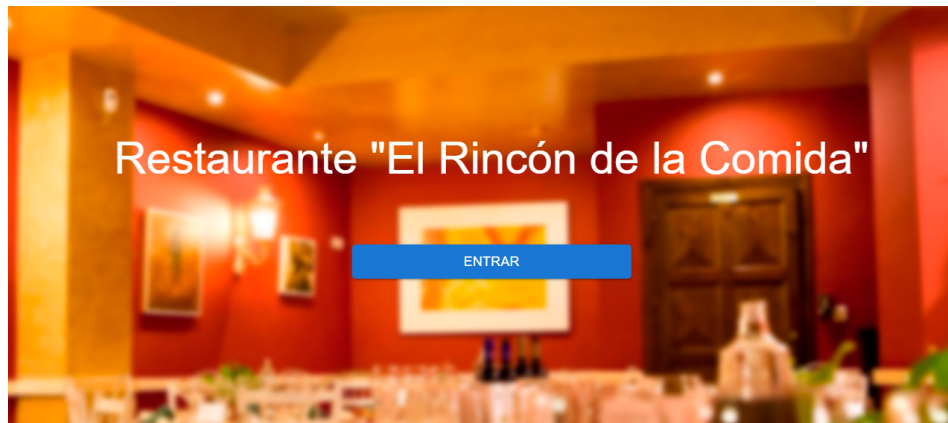


- **Diagrama de desarrollo**



- **Manual de utilización**

Lo primero que se debe hacer al arrancar la aplicación es acceder al primer botón que aparece debajo del nombre del restaurante, “Entrar”.



Una vez estamos dentro de la web del restaurante, se debe hacer es iniciar sesión introduciendo el nombre de usuario y la contraseña, como por ejemplo: nick, contraseñanick.



Acceso de usuario

Usuario *

Contraseña *

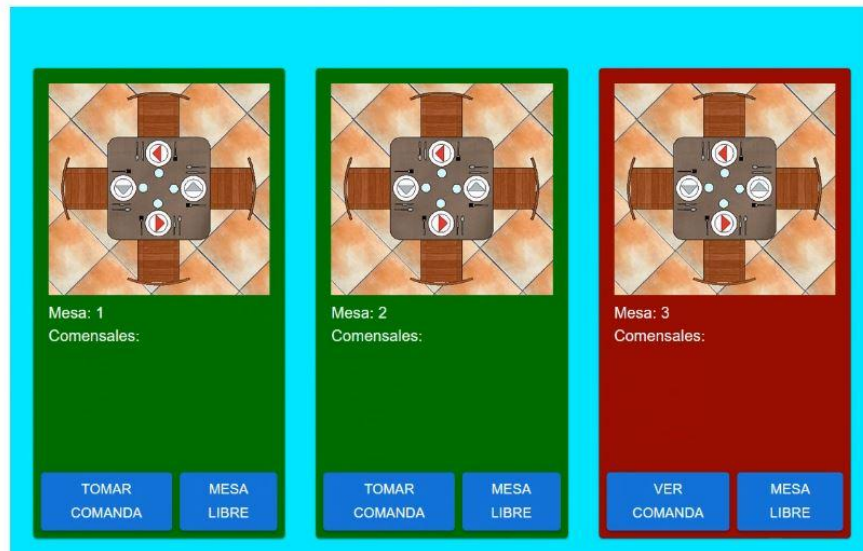
ACCEDER

ATRÁS

Copyright © www.grupoinnova6d.com 2022.

Una vez hecho esto aparece la ventana del comedor donde el usuario podrá ver las mesas, su disponibilidad y tomar comandas de los comensales que haya en cada mesa, los cuales tendrá que indicar como se muestra a continuación.

Una vez se toma la comanda, hacer click en el botón terminar comanda.



COMANDA MESA 1

Comensales:

Primero

Segundo

Postre

☐ Pan

Bebida

intolerancias

Aclaraciones

Primero

Segundo

Postre

☐ Pan

Bebida

intolerancias

Aclaraciones

Se deberán incluir los primeros, segundos, postres y bebidas de cada comensal incluyendo también el pan, las intolerancias y alguna aclaración si se precisa.

El usuario podrá ver las comandas que ha tomado en cada mesa accediendo al botón “Ver comanda” que automáticamente aparecerá en cada mesa de la ventana del comedor.

COMANDA MESA 3

Primeros:

Segundos:

Postres:

Pan:

Bebidas:

VOLVER

El usuario podrá ser de tipo Administrador de forma que podrá añadir platos, ingredientes y usuarios accediendo al botón de la parte superior “Administrador”.

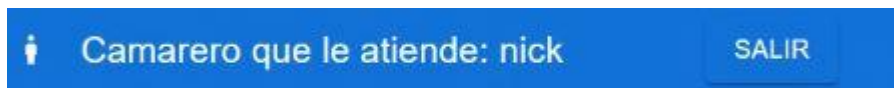
The screenshot shows a web form titled "Nuevo plato" (New dish). The form includes several input fields and a table for ingredients. At the top, there is a purple navigation bar with a home icon and a user profile icon. The form fields are: "Nombre del Plato *" (Dish Name), "Tipo de Plato" (Dish Type) with a dropdown arrow, "Precio del Plato *" (Dish Price), and a checkbox labeled "Contiene alérgenos" (Contains allergens). Below these is a section for "Ingredientes" (Ingredients) which features a search bar containing the number "3" and a table with two columns: "Ingrediente" (Ingredient) and "Cantidad" (Quantity). The table has two rows, each with a dropdown menu for the ingredient and a dropdown menu for the quantity.

Nuevo plato

Nombre del Plato *	Tipo de Plato ▼
Precio del Plato *	<input type="checkbox"/> Contiene alérgenos
Ingredientes	
3 × ▼	
Ingrediente ▼	Cantidad ▼
Ingrediente ▼	Cantidad ▼

Se podrá salir de la página en cuanto el usuario lo desee en el botón “Salir” de la parte superior.

Como se puede ver en la imagen posterior, la aplicación indica al camarero que ha iniciado sesión mostrando su nombre de usuario.



Una vez terminada la comanda, los comensales solo tendrán que esperar a que lleguen los platos pedidos a sus mesas.

.