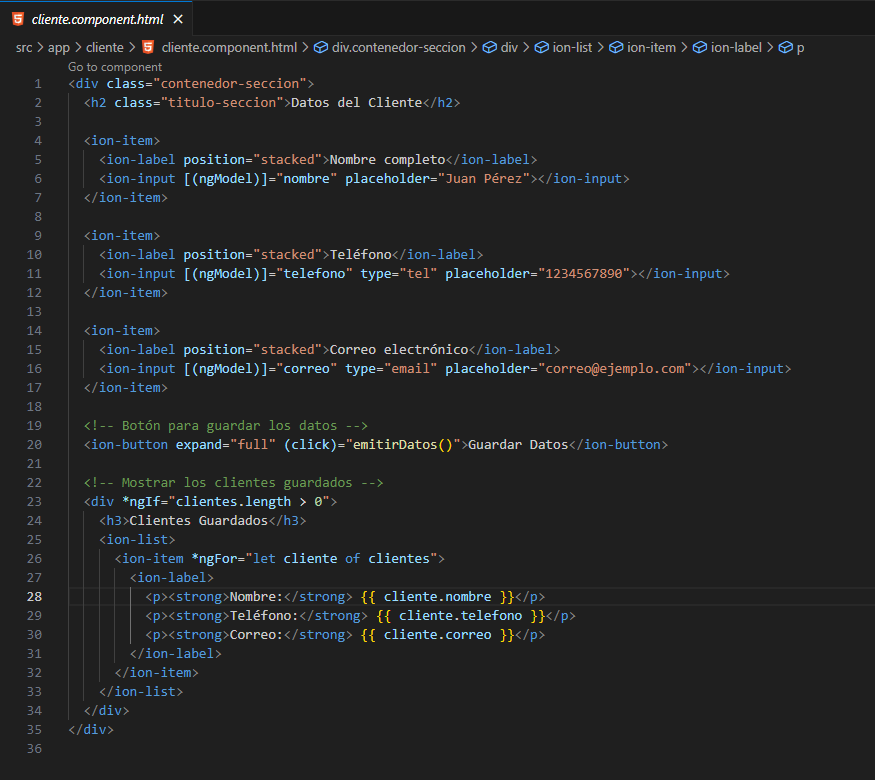
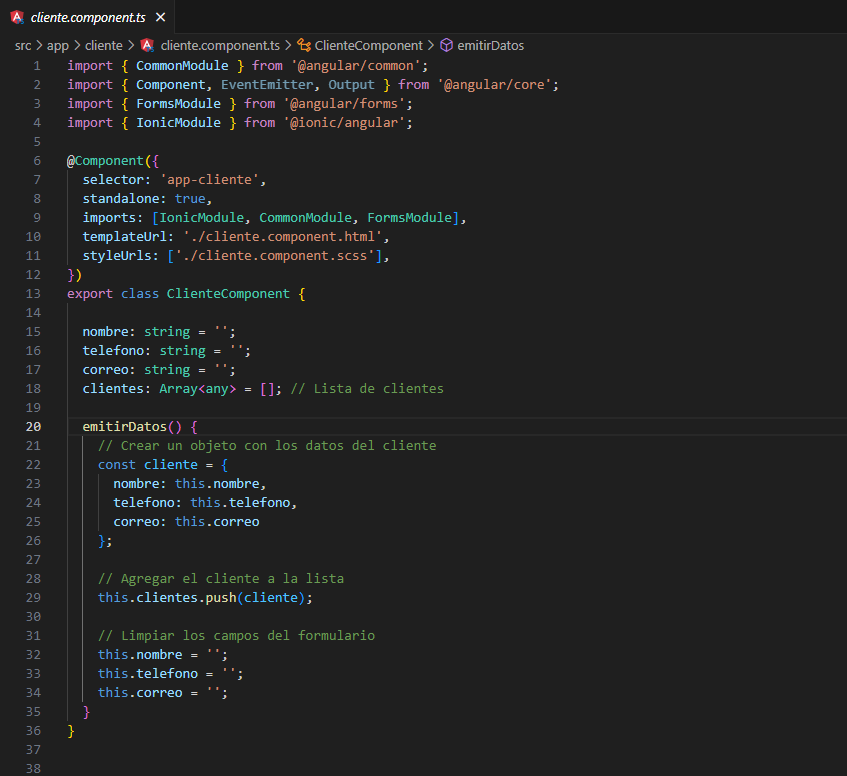
MANUEAL DE USUARIO

CLIENTE:  
  


Este componente de Angular llamado [ClienteComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") es un formulario interactivo que permite al usuario ingresar y guardar datos de clientes. A continuación, se explica cada parte del componente:

**1. Archivo TypeScript (**[cliente.component.ts](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**)**

Este archivo contiene la lógica del componente:

* **Propiedades:**
  + [nombre](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [telefono](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [correo](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Variables enlazadas al formulario para capturar los datos del cliente.
  + [clientes](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Un arreglo que almacena los datos de todos los clientes ingresados.
* **Método**[emitirDatos()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**:**
  + Crea un objeto con los datos ingresados ([nombre](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [telefono](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [correo](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)).
  + Agrega este objeto al arreglo [clientes](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
  + Limpia los campos del formulario para permitir nuevos datos.

**2. Archivo HTML (**[cliente.component.html](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**)**

Este archivo define la estructura visual del componente:

* **Formulario:**
  + Contiene tres campos de entrada (ion-input) para capturar el nombre, teléfono y correo del cliente.
  + Cada campo está enlazado a las propiedades del componente mediante [(ngModel)].
* **Botón Guardar:**
  + Un botón (ion-button) que ejecuta el método [emitirDatos()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") al hacer clic, guardando los datos ingresados.
* **Lista de Clientes:**
  + Si hay clientes guardados ([clientes.length > 0](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")), se muestra una lista (ion-list) con los datos de cada cliente.
  + La lista utiliza \*ngFor para iterar sobre el arreglo [clientes](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y mostrar cada cliente.

**3. Archivo SCSS (**[cliente.component.scss](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**)**

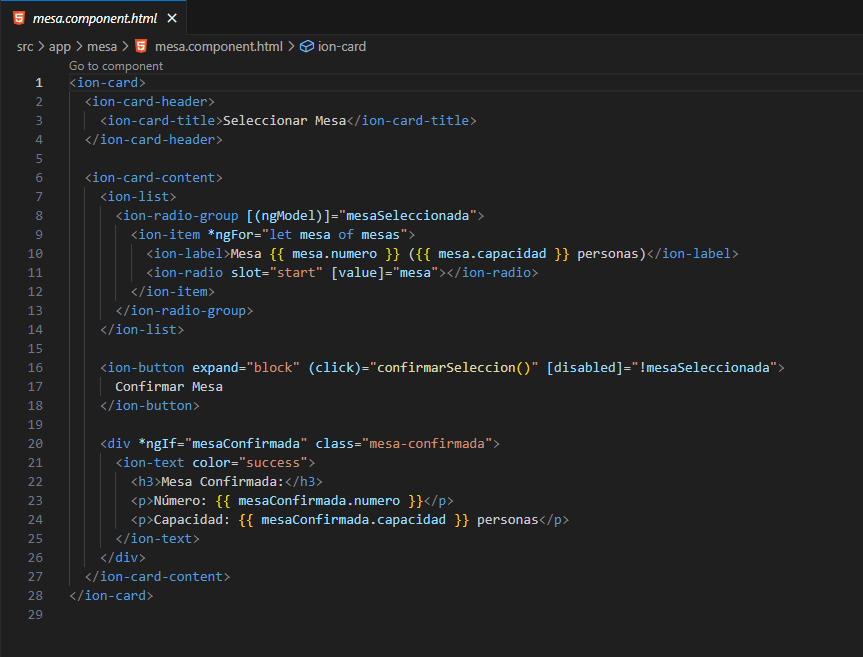
Actualmente, este archivo está vacío, pero puede ser utilizado para agregar estilos personalizados al componente.

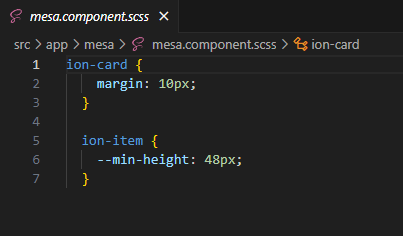
**Flujo del Componente:**

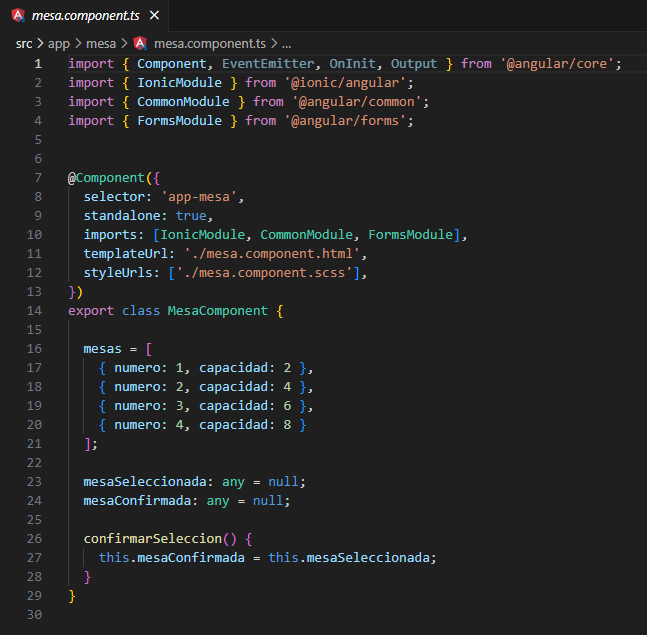
1. El usuario ingresa los datos en los campos del formulario.
2. Al hacer clic en "Guardar Datos", los datos se guardan en el arreglo [clientes](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
3. Los datos guardados se muestran dinámicamente en una lista debajo del formulario.

**Tecnologías Utilizadas:**

* **Angular:** Para la estructura del componente.
* **Ionic Framework:** Para los elementos visuales (ion-item, ion-input, ion-button, etc.).
* **Two-Way Data Binding ([(ngModel)]):** Para enlazar los datos del formulario con las propiedades del componente.
* **Directivas Angular (\*ngIf, \*ngFor):** Para mostrar u ocultar elementos dinámicamente.

Este componente es útil para aplicaciones que requieren capturar y mostrar datos de manera sencilla, como un sistema de gestión de clientes.  
  
MESA:  






Este componente de Angular llamado [MesaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") permite al usuario seleccionar una mesa de una lista y confirmar su elección. A continuación, se explica cada parte del componente:

**1. Archivo TypeScript (mesa.component.ts)**

Este archivo contiene la lógica del componente:

* **Propiedades:**
  + [mesas](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Un arreglo que contiene información sobre las mesas disponibles, incluyendo su número y capacidad.
  + [mesaSeleccionada](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Variable que almacena la mesa seleccionada por el usuario.
  + [mesaConfirmada](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Variable que almacena la mesa confirmada después de que el usuario haga clic en el botón de confirmación.
* **Método**[confirmarSeleccion()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**:**
  + Asigna la mesa seleccionada ([mesaSeleccionada](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")) a la variable [mesaConfirmada](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), indicando que la selección ha sido confirmada.

**2. Archivo HTML (mesa.component.html)**

Este archivo define la estructura visual del componente:

* **Lista de Mesas:**
  + Utiliza un ion-radio-group para permitir al usuario seleccionar una mesa de la lista.
  + Cada mesa se representa como un ion-item con un ion-radio que permite la selección.
  + La directiva \*ngFor itera sobre el arreglo [mesas](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) para generar dinámicamente los elementos de la lista.
* **Botón Confirmar:**
  + Un botón (ion-button) que ejecuta el método [confirmarSeleccion()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") al hacer clic.
  + El botón está deshabilitado ([disabled]="!mesaSeleccionada") hasta que el usuario seleccione una mesa.
* **Confirmación de Mesa:**
  + Si hay una mesa confirmada ([mesaConfirmada](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")), se muestra un mensaje con los detalles de la mesa seleccionada (número y capacidad).

**3. Archivo SCSS (mesa.component.scss)**

Este archivo contiene estilos personalizados para el componente:

* **ion-card:** Define un margen de 10px para las tarjetas.
* **ion-item:** Ajusta la altura mínima de los elementos de la lista.

**Flujo del Componente:**

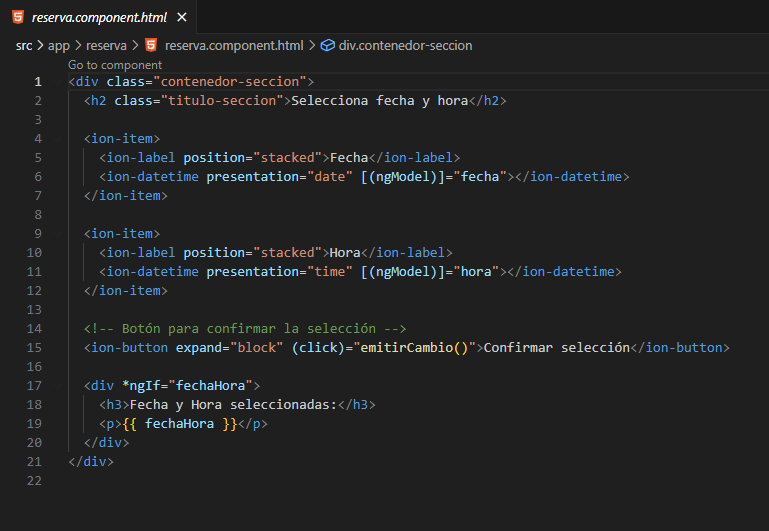
1. El usuario ve una lista de mesas disponibles con su número y capacidad.
2. Selecciona una mesa utilizando los botones de radio (ion-radio).
3. Hace clic en el botón "Confirmar Mesa" para confirmar su selección.
4. Los detalles de la mesa confirmada se muestran en la parte inferior del componente.

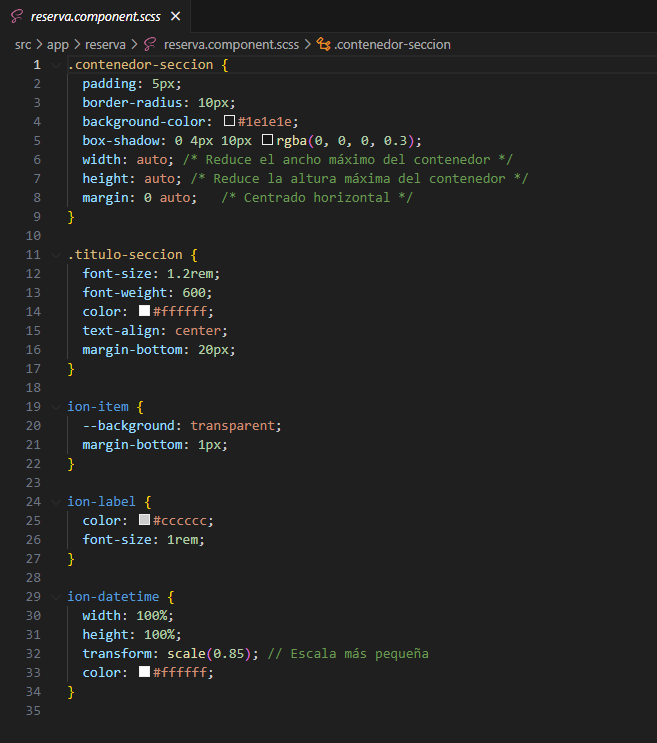
**Tecnologías Utilizadas:**

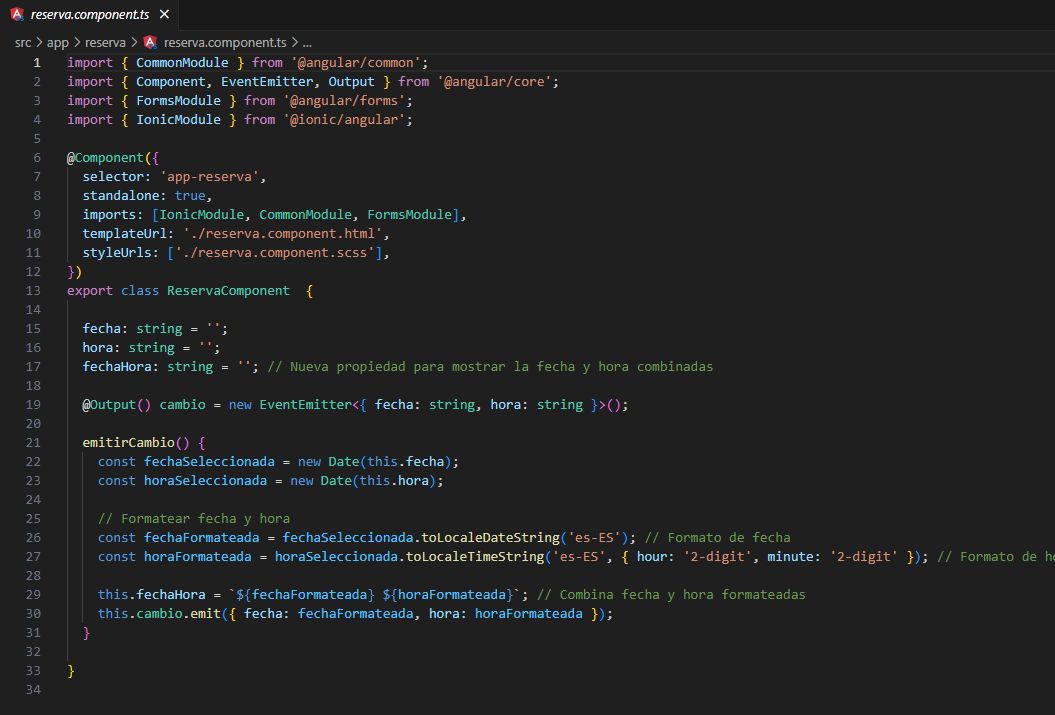
* **Angular:** Para la estructura del componente.
* **Ionic Framework:** Para los elementos visuales (ion-card, ion-radio-group, ion-button, etc.).
* **Two-Way Data Binding ([(ngModel)]):** Para enlazar la selección de la mesa con la propiedad [mesaSeleccionada](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
* **Directivas Angular (\*ngFor, \*ngIf):** Para generar dinámicamente la lista de mesas y mostrar u ocultar elementos según las condiciones.

Este componente es útil para aplicaciones que requieren la selección de opciones, como un sistema de reservas de mesas en un restaurante.

RESERVA:







El componente [ReservaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") es un formulario interactivo que permite al usuario seleccionar una fecha y una hora, combinarlas y emitir un evento con los datos seleccionados. A continuación, se explica cada parte del componente:

**1. Archivo TypeScript (reserva.component.ts)**

Este archivo contiene la lógica del componente:

* **Propiedades:**
  + [fecha](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Variable enlazada al selector de fecha (ion-datetime) para almacenar la fecha seleccionada.
  + [hora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Variable enlazada al selector de hora (ion-datetime) para almacenar la hora seleccionada.
  + [fechaHora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Propiedad que combina la fecha y la hora seleccionadas en un formato legible.
* **Decorador @Output:**
  + [cambio](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Un evento que emite un objeto con la fecha y hora seleccionadas al componente padre.
* **Método**[emitirCambio()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**:**
  + Convierte las variables [fecha](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y [hora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) en objetos [Date](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
  + Formatea la fecha y la hora en un formato legible (es-ES).
  + Combina la fecha y la hora formateadas en la propiedad [fechaHora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
  + Emite un evento con los datos formateados ([fecha](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y [hora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)) al componente padre.

**2. Archivo HTML (reserva.component.html)**

Este archivo define la estructura visual del componente:

* **Selector de Fecha:**
  + Un campo de entrada (ion-datetime) con presentación de tipo date para seleccionar una fecha.
  + Está enlazado a la propiedad [fecha](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) mediante [(ngModel)].
* **Selector de Hora:**
  + Un campo de entrada (ion-datetime) con presentación de tipo time para seleccionar una hora.
  + Está enlazado a la propiedad [hora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) mediante [(ngModel)].
* **Botón Confirmar:**
  + Un botón (ion-button) que ejecuta el método [emitirCambio()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") al hacer clic.
  + Este botón confirma la selección de fecha y hora.
* **Visualización de Fecha y Hora Seleccionadas:**
  + Si la propiedad [fechaHora](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") tiene un valor, se muestra un mensaje con la fecha y hora seleccionadas.

**3. Archivo SCSS (reserva.component.scss)**

Este archivo contiene estilos personalizados para el componente:

* **contenedor-seccion:**
  + Define un contenedor con bordes redondeados, fondo oscuro, y sombra para un diseño moderno.
  + Centra el contenido horizontalmente y ajusta el tamaño automáticamente.
* **titulo-seccion:**
  + Estiliza el título con un tamaño de fuente mediano, color blanco, y alineación centrada.
* **Estilos de ion-item, ion-label, y ion-datetime:**
  + Personalizan los elementos de entrada para que se integren con el diseño oscuro.
  + Reducen el tamaño de los selectores de fecha y hora para un diseño más compacto.

**Flujo del Componente:**

1. El usuario selecciona una fecha y una hora utilizando los selectores (ion-datetime).
2. Al hacer clic en "Confirmar selección", el método [emitirCambio()](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") combina y formatea los datos.
3. La fecha y hora seleccionadas se muestran en pantalla y se emite un evento al componente padre con los datos.

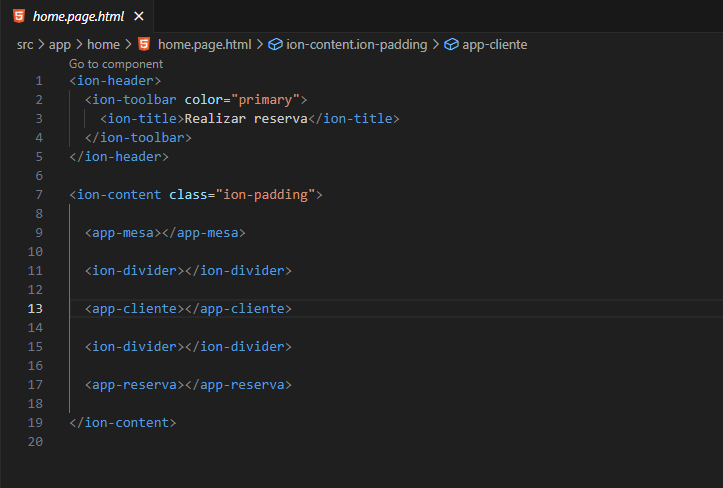
**Tecnologías Utilizadas:**

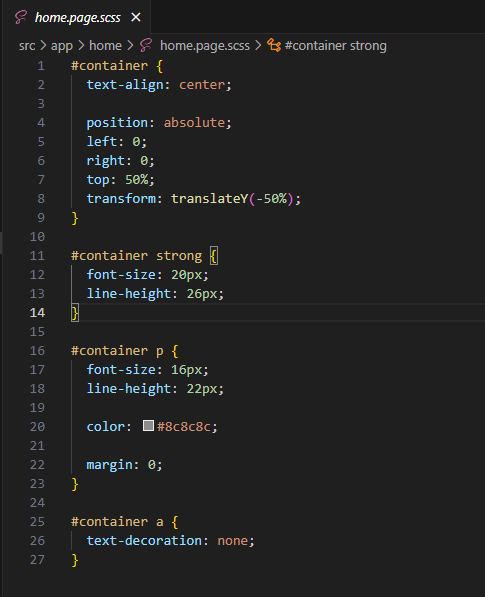
* **Angular:** Para la estructura del componente.
* **Ionic Framework:** Para los elementos visuales (ion-datetime, ion-item, ion-button, etc.).
* **Two-Way Data Binding ([(ngModel)]):** Para enlazar los selectores de fecha y hora con las propiedades del componente.
* **EventEmitter (@Output):** Para emitir eventos al componente padre.

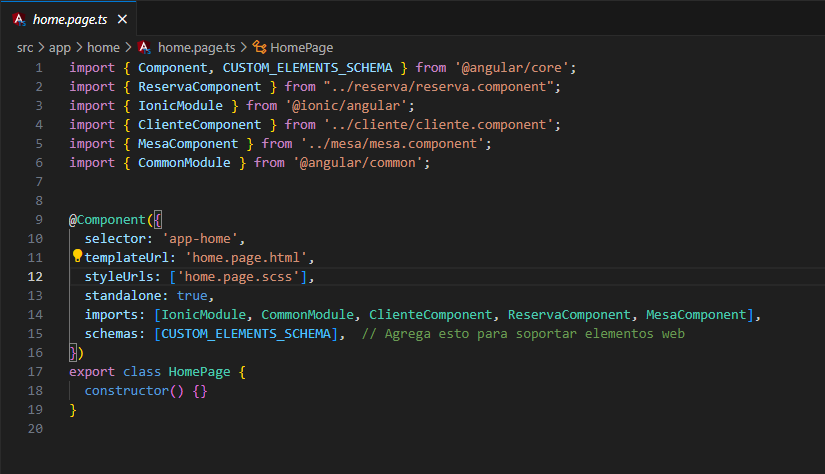
**Uso del Componente:**

Este componente es ideal para aplicaciones que requieren la selección de una fecha y hora, como sistemas de reservas, citas o eventos. Su diseño moderno y funcionalidad lo hacen fácil de usar e integrar en aplicaciones más grandes.

HOME:







El componente [HomePage](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") es la página principal de la aplicación, que actúa como un contenedor para los componentes [MesaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [ClienteComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") y [ReservaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "). Su propósito es integrar estos componentes para que el usuario pueda realizar una reserva completa (selección de mesa, ingreso de datos del cliente y selección de fecha y hora). A continuación, se explica cada parte del componente:

**1. Archivo TypeScript (home.page.ts)**

Este archivo define la lógica básica de la página:

* **Importaciones:**
  + Importa los componentes [MesaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [ClienteComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y [ReservaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) para utilizarlos dentro de esta página.
  + También importa módulos esenciales como [IonicModule](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") y [CommonModule](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
* **Decorador @Component:**
  + Define el selector app-home, que se utiliza en el archivo HTML.
  + Especifica los estilos (home.page.scss) y la plantilla (home.page.html) de la página.
  + Incluye los componentes importados en la propiedad [imports](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") para que puedan ser utilizados en la plantilla.
  + Utiliza [CUSTOM\_ELEMENTS\_SCHEMA](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) para permitir el uso de componentes personalizados como app-mesa, app-cliente y app-reserva.
* **Clase**[HomePage](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")**:**
  + Actualmente, no contiene lógica adicional, ya que su propósito principal es servir como un contenedor para los componentes.

**2. Archivo HTML (home.page.html)**

Este archivo define la estructura visual de la página:

* **Encabezado (ion-header):**
  + Contiene una barra de herramientas (ion-toolbar) con el título "Realizar reserva".
* **Contenido Principal (ion-content):**
  + Incluye los tres componentes principales:
    1. **<app-mesa>:** Renderiza el componente para seleccionar una mesa.
    2. **<app-cliente>:** Renderiza el componente para ingresar los datos del cliente.
    3. **<app-reserva>:** Renderiza el componente para seleccionar la fecha y hora de la reserva.
  + Los componentes están separados por divisores (ion-divider) para mejorar la organización visual.

**3. Archivo SCSS (home.page.scss)**

Este archivo contiene estilos personalizados para la página:

* **#container:**
  + Define un contenedor centrado verticalmente y horizontalmente.
  + Aunque no se utiliza directamente en el archivo HTML actual, puede ser útil para centrar contenido adicional en la página.
* **Estilos de texto (#container strong, #container p, #container a):**
  + Estilizan elementos de texto con tamaños de fuente específicos y colores.
  + Estos estilos no están aplicados a los componentes actuales, pero podrían ser utilizados en futuras modificaciones.

**Flujo del Componente:**

1. El usuario selecciona una mesa utilizando el componente [MesaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
2. Ingresa los datos del cliente en el componente [ClienteComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
3. Selecciona la fecha y hora de la reserva en el componente [ReservaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
4. Cada componente funciona de manera independiente, pero juntos forman un flujo completo para realizar una reserva.

**Tecnologías Utilizadas:**

* **Angular:** Para la estructura de la página y la integración de componentes.
* **Ionic Framework:** Para los elementos visuales (ion-header, ion-toolbar, ion-content, etc.).
* **Componentes Personalizados:** Los componentes [MesaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [ClienteComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) y [ReservaComponent](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) son reutilizados dentro de esta página.

**Propósito del Componente:**

El componente [HomePage](vscode-file://vscode-app/c:/Users/Molina211/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") sirve como una interfaz principal para que el usuario pueda realizar una reserva completa. Su diseño modular permite que cada parte del proceso (selección de mesa, datos del cliente, fecha y hora) sea manejada por un componente independiente, lo que facilita la escalabilidad y el mantenimiento del código.

