

TALLER PIZZASHOP

Blazor

Trabajo realizado por Luis Moreno-Torres

Introducción	3
Desarrollo del Proyecto	3
2.1 Descripción del Taller	3
2.2 Sesiones de Desarrollo	3
Sesión 1: Introducción a Blazor y configuración del proyecto.....	3
Estructura del proyecto	3
Ejecución de la aplicación	3
Sesión 2: Creación de componentes y manejo de eventos	4
Creación de un nuevo componente.....	4
Manejo de eventos.....	4
Comunicación entre componentes	4
Sesión 3: Mostrar el estado del pedido	5
Agregar enlace de navegación	5
Creación de la página "Mis Pedidos"	5
Obtención de datos del servidor	6
Manejo de estados de carga	6
Visualización detallada del pedido	6
Creación de la página de detalles del pedido	6
Actualización en tiempo real.....	6
Navegación automática tras realizar un pedido	6
Sesión 4: Refactorización de la gestión del estado.....	7
Implementación de un servicio de estado.....	7
Inyección de dependencias	7
Sincronización de datos	7
Sesión 5: Proceso de pago con validación.....	7
Creación del formulario de pago	8
Validación de datos	8
Envío del formulario	8
Sesión 6: Autenticación y autorización	8
Implementación de autenticación.....	9
Restricción de acceso.....	10
Gestión de roles	10

Sesión 7: Seguimiento del pedido en tiempo real.....	10
Integración con mapas.....	10
Actualización dinámica	10
Sesión 8: Uso de componentes genéricos.....	10
Creación de componentes reutilizables	10
Implementación de parámetros de plantilla.....	10
Sesión 9: Convertir la app en una PWA	11
Habilitación de PWA	11
Funcionalidad offline	11
Sesión 10: Despliegue de la aplicación	11
Preparación para producción	11
Publicación en Azure.....	11
Pruebas post-despliegue.....	12
2.3 Diagramas de Flujo	12
2.4 Justificación de Mejoras en UI/UX	13
Personalizaciones Implementadas.....	13
Extras y Mejoras Adicionales	14
Control de Versiones y Comentarios en Código	14
Conclusión y Reflexión	15

Introducción

Este documento presenta el desarrollo del proyecto **Pizzashop**, basado en el taller de Blazor. Se describe el proceso de implementación de la aplicación, personalizaciones añadidas y mejoras realizadas para optimizar la experiencia del usuario.

Desarrollo del Proyecto

2.1 Descripción del Taller

El taller Blazor guía en la creación de una aplicación de pedidos de pizza utilizando el framework Blazor. Se trabajó con componentes, gestión de estado y comunicación entre componentes.

2.2 Sesiones de Desarrollo

Sesión 1: Introducción a Blazor y configuración del proyecto

Estructura del proyecto

- Se exploran las carpetas y archivos principales.
- Se explica la función de cada uno en la aplicación.

Ejecución de la aplicación

- Se compila y ejecuta la aplicación para verificar su correcto funcionamiento.

```

@page "/"
@Inject HttpClient HttpClient

<div class="main">
  <ul class="pizza-cards">
    @if (specials is not null)
    {
      @foreach (var special in specials)
      {
        <li style="background-image: url('@special.ImageUrl')">
          <div class="pizza-info">
            <span class="title">@special.Name</span>
            @special.Description
            <span class="price">@special.GetFormattedBasePrice()</span>
          </div>
        </li>
      }
    }
  </ul>
</div>

@code {
  List<PizzaSpecial>? specials;

  protected override async Task OnInitializedAsync()
  {
    specials = await HttpClient.GetFromJsonAsync("specials", BlazingPizza.OrderContext.Default.Lis
  }
}

```

Sesión 2: Creación de componentes y manejo de eventos

Creación de un nuevo componente

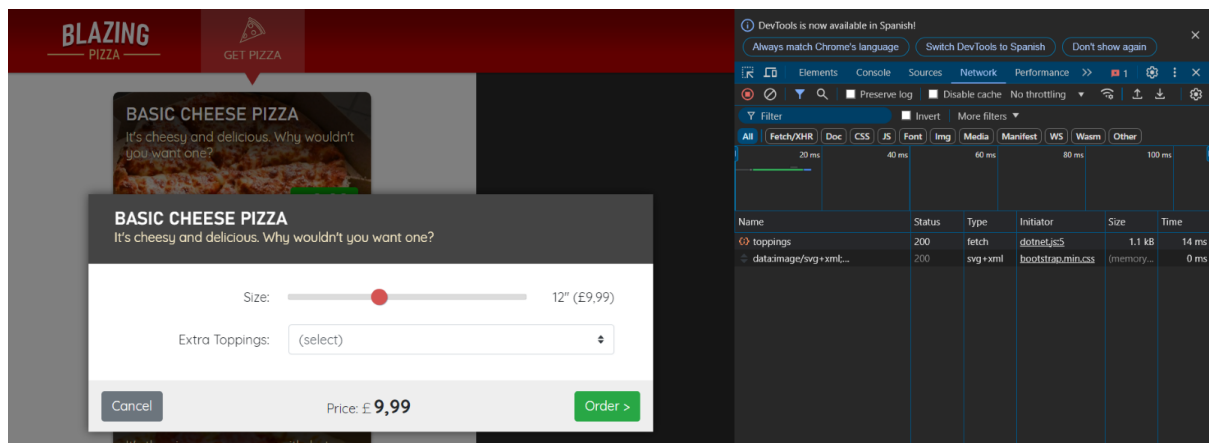
- Se añade `PizzaSpecials.razor` en `Shared` para mostrar las ofertas de pizzas.

Manejo de eventos

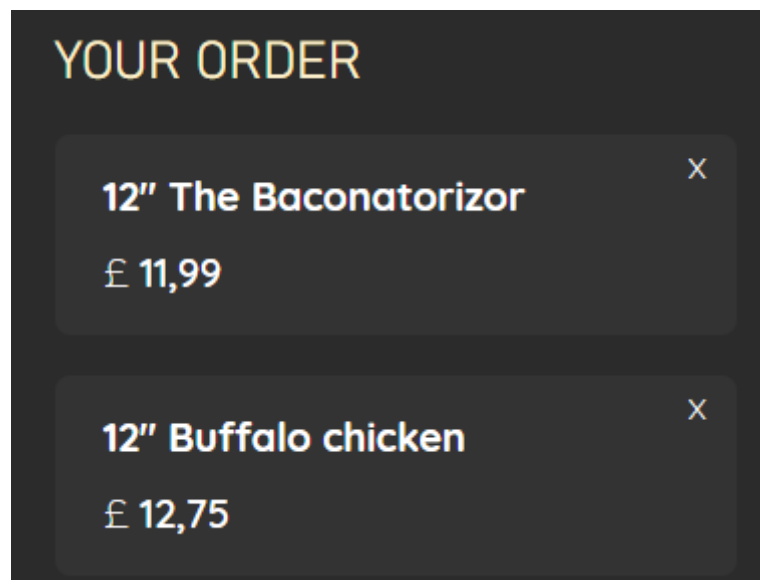
- Se implementa `@onclick` en cada pizza especial para manejar su selección.

Comunicación entre componentes

- Se utilizan parámetros de componentes para pasar datos entre padre e hijo.



Podemos ver cómo se envía una solicitud para mostrar los ingredientes a añadir cuando hacemos clic en la pizza (el nombre por consola no he conseguido que se muestre en ninguno de los navegadores que tengo).



(Cesta de compra)

Sesión 3: Mostrar el estado del pedido

Agregar enlace de navegación

- Se incluye un enlace en "Mis Pedidos" que dirige a `/myorders`.

Creación de la página "Mis Pedidos"

- Se crea `MyOrders.razor` con la ruta `/myorders`.
- Se muestra un mensaje de "Cargando..." mientras se obtienen los datos.

Obtención de datos del servidor

- Se inyecta `HttpClient` y se realiza una solicitud GET a la API para obtener los pedidos.

Manejo de estados de carga

- Se implementan condiciones para mostrar mensajes según el estado:
 - **Cargando**
 - **Sin pedidos**
 - **Lista de pedidos**

Visualización detallada del pedido

- Se muestra la fecha, cantidad de artículos, precio total y estado del pedido.
- Se añade un botón "Rastrear" para ver detalles.

A green rectangular button with rounded corners containing the text "Track >" in white.

Creación de la página de detalles del pedido

- Se crea `OrderDetails.razor` con la ruta `/myorders/{orderId:int}`.
- Se muestra la información del pedido seleccionado.

Actualización en tiempo real

- Se consulta el servidor cada pocos segundos para actualizar el estado del pedido.
- Se reflejan cambios como "Preparando", "En camino" o "Entregado".

ORDER PLACED SÁBADO, 15 DE FEBRERO DE 2025		Status: Delivered
12" Buffalo chicken (£12,75)		
12" The Baconatorizer (£11,99)		
Total price: £24,74		

Navegación automática tras realizar un pedido

- Se redirige automáticamente a los detalles del pedido recién creado.

BLAZING
PIZZA

GET PIZZA

MY ORDERS

ORDER PLACED SÁBADO, 15 DE FEBRERO DE 2025

Status: **Preparing**

12" Buffalo chicken (£12,75)

12" The Baconatorizer (£11,99)

Total price: £24,74

Sesión 4: Refactorización de la gestión del estado

Implementación de un servicio de estado

- Se crea un servicio para gestionar el estado de la orden.

Inyección de dependencias

- Se inyecta el servicio en los componentes que lo necesitan.

Sincronización de datos

- Se asegura que todos los componentes reflejen los cambios en la orden en tiempo real.

Sesión 5: Proceso de pago con validación

BLAZING
PIZZA

GET PIZZA

MY ORDERS

luismtmdam@gmail.com

Sign out

Review order

12" The Brit (£10,25)

Total price: £10,25

Deliver to...

Name:

Line 1:

Line 2:

City:

Region:

Postal code:

Place order

Deliver to...

Name:

The Name field is required.

Line 1:

The Line1 field is required.

Line 2:

City:

The City field is required.

Region:

The Region field is required.

Postal code:

The PostalCode field is required.

Creación del formulario de pago

- Se añade un formulario en Checkout .razor para ingresar información de entrega.

Validación de datos

- Se implementan validaciones en el formulario para evitar datos incorrectos.

Envío del formulario

- Se envían los datos al servidor tras validarlos correctamente.

Sesión 6: Autenticación y autorización

REGISTER

LOGIN

Register

Create a new account.

Email

Password

Confirm Password

Register

Use another service to register.

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services](#).

Log in

Use a local account to log in.

Email

Password

☐ Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)

[Resend email confirmation](#)



Implementación de autenticación

- Se configura el sistema de autenticación para iniciar sesión y registrar usuarios.

Restricción de acceso

- Se protege la página de pedidos para que solo sea accesible a usuarios autenticados.

Gestión de roles

- Se definen permisos según el tipo de usuario.

Sesión 7: Seguimiento del pedido en tiempo real



Integración con mapas

- Se usa un servicio de mapas para mostrar la ubicación del repartidor.

Actualización dinámica

- Se refresca la posición en tiempo real sin necesidad de recargar la página.

Sesión 8: Uso de componentes genéricos

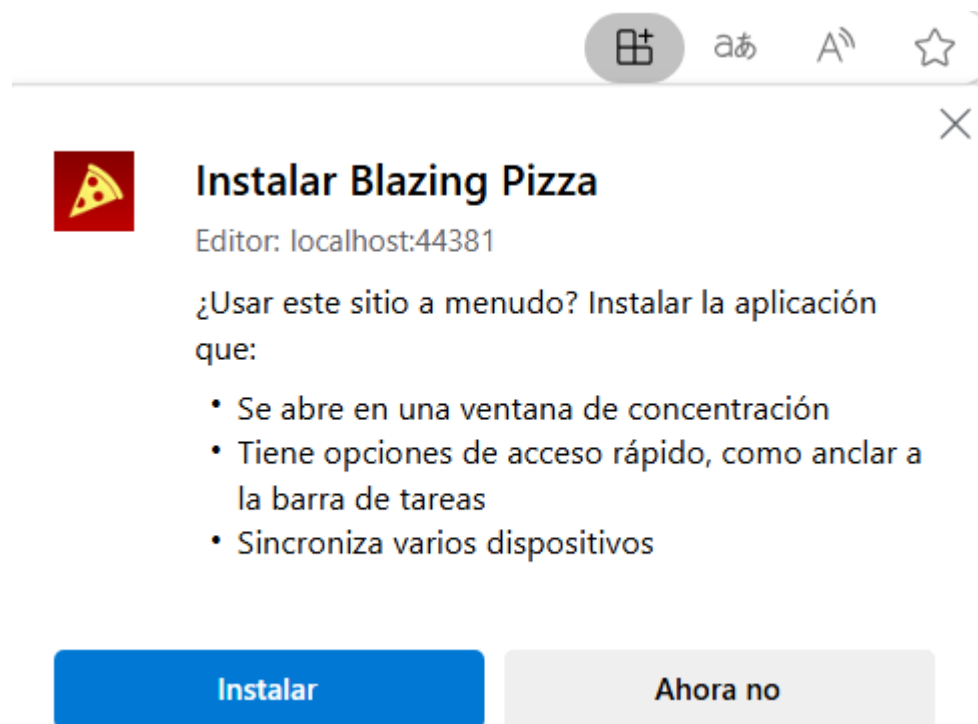
Creación de componentes reutilizables

- Se desarrollan componentes que pueden aceptar diferentes tipos de datos.

Implementación de parámetros de plantilla

- Se utilizan parámetros genéricos para personalizar la funcionalidad del componente.

Sesión 9: Convertir la app en una PWA



Habilitación de PWA

- Se configuran archivos de servicio y manifest para permitir instalación en dispositivos móviles.

Funcionalidad offline

- Se implementa almacenamiento en caché para permitir el uso sin conexión.

Sesión 10: Despliegue de la aplicación

Preparación para producción

- Se realizan ajustes de compilación y optimización.

Publicación en Azure

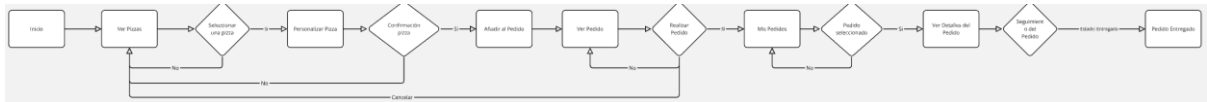
- Se sube la aplicación a Azure y se configura el servidor.

Pruebas post-despliegue

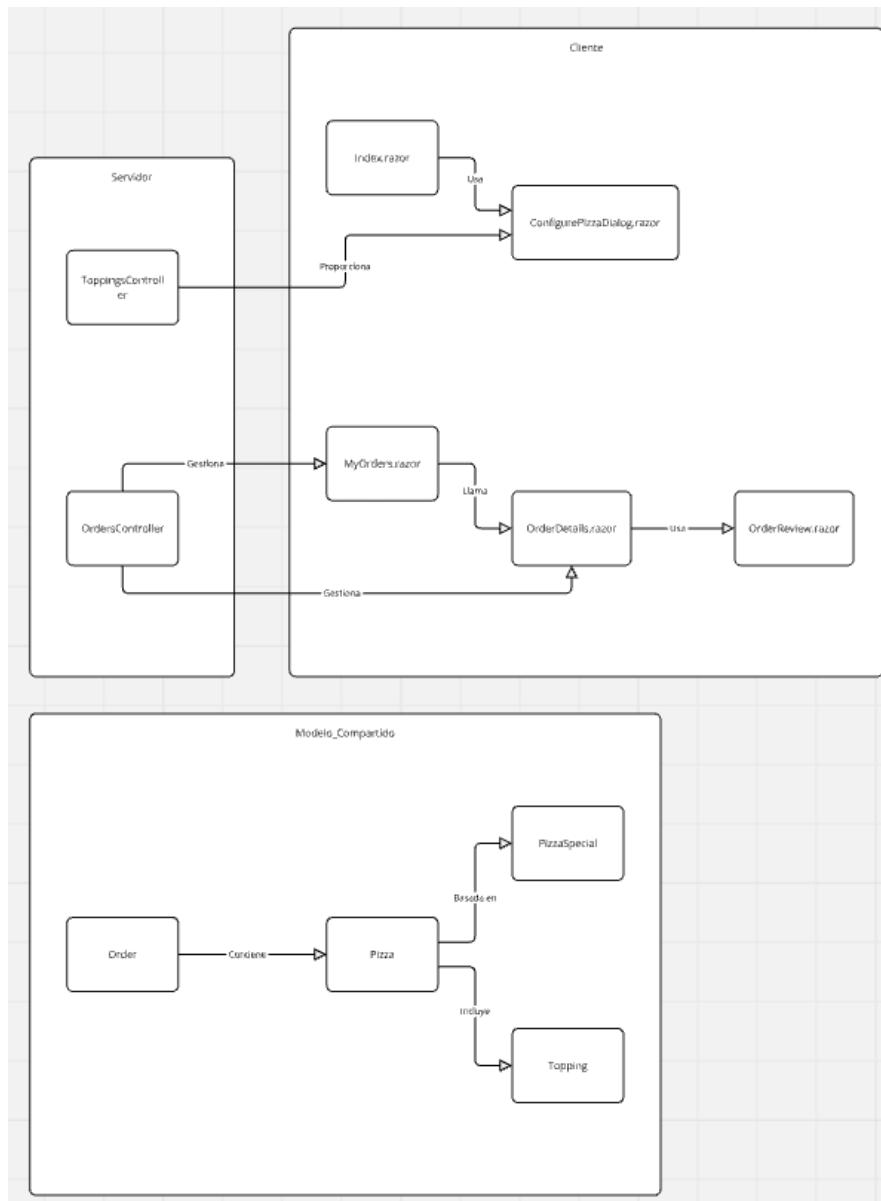
- Se verifica que todas las funcionalidades operen correctamente en producción.

2.3 Diagramas de Flujo

Diagrama de Flujo de la Aplicación



Diagramas de Clases y Componentes



2.4 Justificación de Mejoras en UI/UX

- Se ha mejorado la interfaz en el inicio de sesión para hacerla más dinámica y atractiva.

Personalizaciones Implementadas

- **Personalización de colores:** Se han cambiado los colores para un diseño más atractivo.
- Sistema de suscripción a notificaciones

- **Imágenes y nombres personalizados:** Se han modificado imágenes y nombres de la página.

```
UPDATE Specials SET imageUrl = 'img/pizzas_luis/bomba.svg' WHERE Id = 1;
UPDATE Specials SET Name = 'Bomba' WHERE Id = 1;
UPDATE Specials SET BasePrice = 1.00 WHERE Id = 1;
UPDATE Specials SET Description = 'You are about to explode' WHERE Id = 1;

UPDATE Specials SET imageUrl = 'img/pizzas_luis/pizza_humana.svg' WHERE Id = 2;
UPDATE Specials SET Name = 'Human' WHERE Id = 2;
UPDATE Specials SET BasePrice = 19.99 WHERE Id = 2;
UPDATE Specials SET Description = 'There u go, take a bit of carnivalism' WHERE Id = 2;

UPDATE Specials SET imageUrl = 'img/pizzas_luis/kilometrica.svg' WHERE Id = 3;
UPDATE Specials SET Name = 'Kilometric' WHERE Id = 3;
UPDATE Specials SET BasePrice = 1000.00 WHERE Id = 3;
UPDATE Specials SET Description = 'You will never finish this' WHERE Id = 3;

UPDATE Specials SET imageUrl = 'img/pizzas_luis/nutella.svg' WHERE Id = 4;
UPDATE Specials SET imageUrl = 'img/pizzas_luis/precongelada.svg' WHERE Id = 5;

SELECT * FROM Specials;
```

Para modificar la información de las pizzas, se ha de acceder a Specials en la base de datos SQLite (en mi caso utilicé DB Browser).

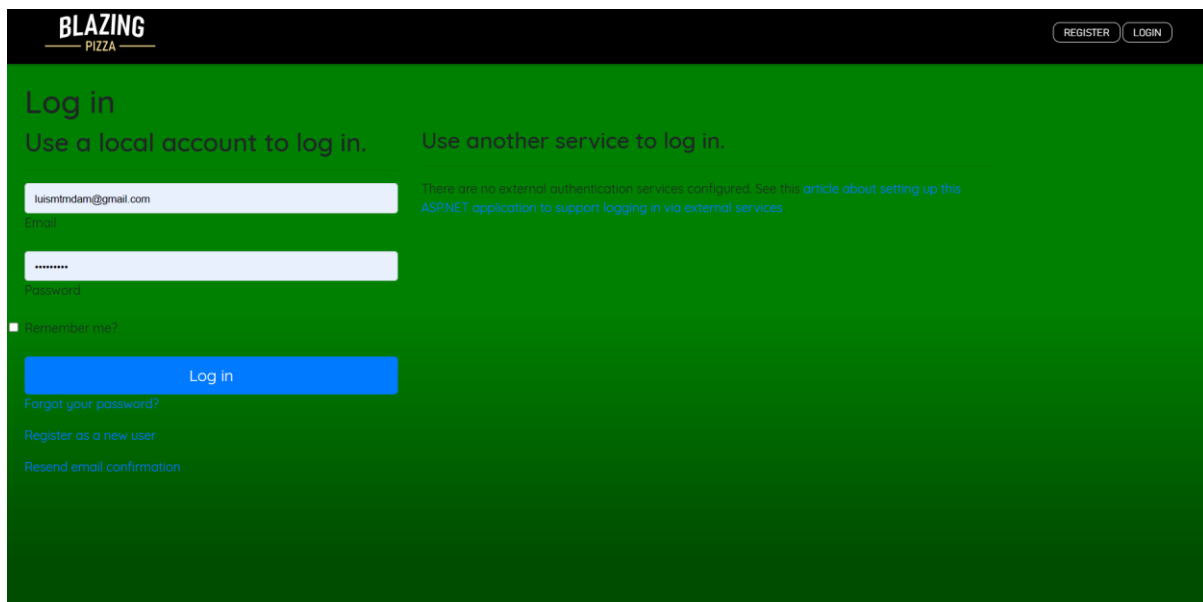
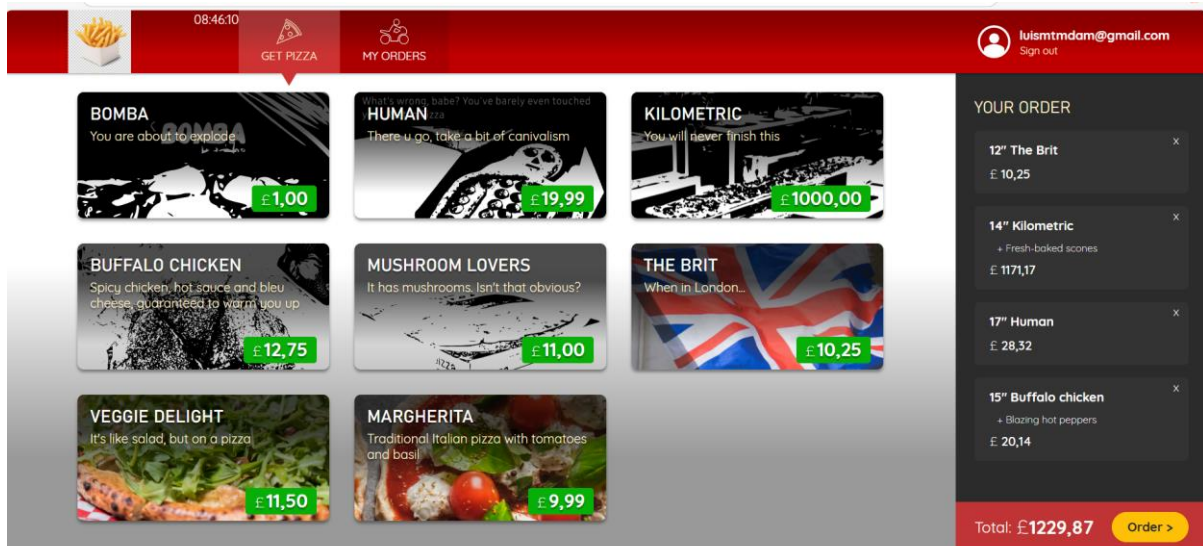
Extras y Mejoras Adicionales

- **Hora en tiempo real:** Se ha añadido la hora actual al lado del logo, permitiendo al usuario tener en cuenta el tiempo a la hora de pedir y saber si queda poco para la hora de cierre de la pizzería.

Control de Versiones y Comentarios en Código

- Se utilizó **GitHub** para el control de versiones.
- Se añadieron comentarios en el código explicando la funcionalidad de cada componente.

Resultado:



Conclusión y Reflexión

Este proyecto permitió explorar las capacidades de Blazor en el desarrollo web, mejorando habilidades en diseño de interfaces, gestión de estado y personalización de aplicaciones. Se lograron implementar mejoras significativas en la usabilidad y experiencia del usuario.