

Aplicación “La huerta” - Frontend

Datos del estudiante

Creación de pagina de ingreso a la aplicación

Creación de la página principal (Usuario)

Página de la tienda

Página de carrito de compras

Página de notificaciones

Creación de la página principal del administrador

Página de administración de productos

Pantalla de administración de órdenes

Datos del estudiante

Este taller fue elaborado por Luis Danilo Juajinoy Gálvez con código estudiantil 2130341182.

El código fuente se encuentra en el siguiente repositorio

<https://github.com/LuisDanilo/FruverApp/tree/main>

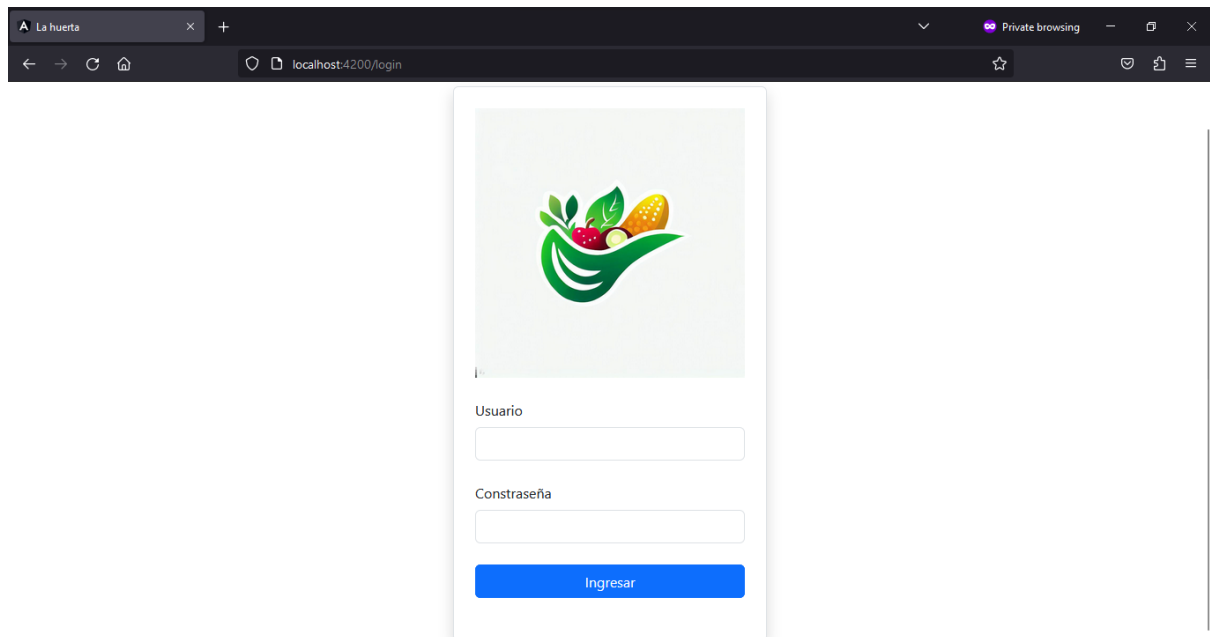
Creación de pagina de ingreso a la aplicación

Uno de los requerimientos es el manejo de roles, donde el administrador es el encargado de la gestión de órdenes y productos y los usuarios son los clientes que pueden crear pedidos a través de un carrito de compras.

Para ello se crea un componente que ofrece un formulario de ingreso con usuario y contraseña, el cual se comunica con el backend para realizar el proceso de autenticación de una persona.

Se emplearon las siguientes instrucciones para la creación de los elementos necesarios

```
# Para crear la estructura del componente Login
ng generate component login
# Para crear la estructura del componente LogoutButton
ng generate component logout-button
# Para crear el servicio de Login
ng generate service login
```



Pantalla de ingreso a la aplicación

```
export class LoginService {  
  
  constructor(private http: HttpClient) { }  
  
  /**  
   * Método que realiza la petición al backend para ejecutar un login.  
   * Retorna datos de autenticación y autorización si el login fue exitoso.  
   */  
  performLogin(credentials: Credentials) {  
    return this.http.post<UserSession>(`${BACKEND_HOST}/login`, credentials)  
  }  
  
  /**  
   * Método que realiza la petición al backend para ejecutar un logout.  
   */  
  performLogout() {  
    return this.http.get(`${BACKEND_HOST}/logout`, {  
      headers: getAuthHeaders()  
    })  
  }  
}
```

Creación de la página principal (Usuario)

Se emplearon los siguientes comandos para la creación de los elementos necesarios

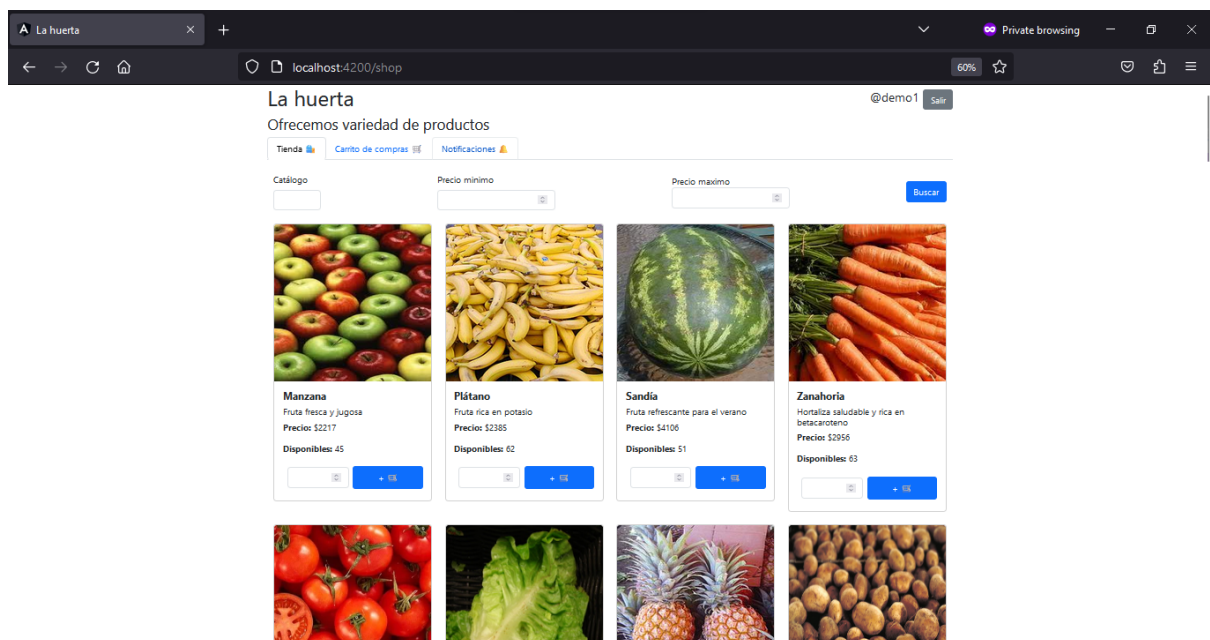
```
# Para generar la estructura del componente Shop  
ng generate component shop  
# para generar la estructura del componente ProductCard  
ng generate component product-card
```

```
# Para generar la estructura del componente ShoppingCartItemCard
ng generate component shopping-cart-item-card
# Para generar la estructura del componente OrderConfirmationModal
ng generate component order-confirmation-modal
# Para generar el servicio de Order
ng generate service order
# Para generar el servicio de Shop
ng generate service shop
# Para generar el servicio de Product
ng generate service product
# Para generar el servicio de Notification
ng generate service notification
```

La página principal del usuario consiste en un apartado dividido en tres secciones

Página de la tienda

La primera es una pantalla de presentación a modo de catálogo de todos los productos ofrecidos en donde se pueden visualizar productos, filtrar productos por catálogo y rango de precios, agregar dichos productos al carrito de compras por las unidades que se deseen



Pantalla de la tienda, visualización y filtrado de productos

```
export class ProductService {

  constructor(private http: HttpClient) { }

  /**
   * Método que ejecuta la petición al backend para obtener productos.
   */
  getProducts(options?: any) {
    const maxPrice = get(options, 'maxPrice', null)
    const minPrice = get(options, 'minPrice', null)
    const catalog = get(options, 'catalog', null)
```

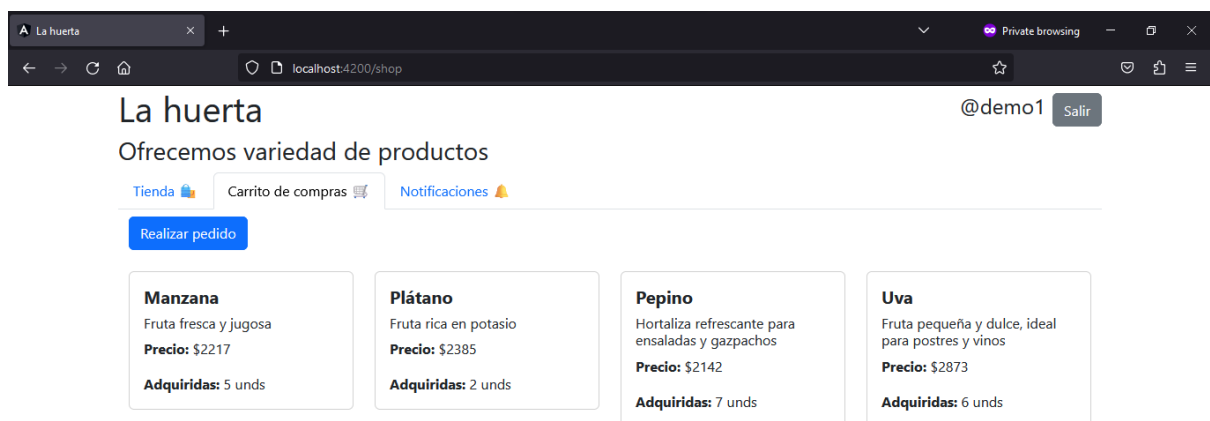
```

let params = new HttpParams()
if (minPrice) {
  params = params.set('min', minPrice)
}
if (maxPrice) {
  params = params.set('max', maxPrice)
}
if (catalog) {
  params = params.set('catalog', catalog)
}
return this.http.get<Product[]>(`${BACKEND_HOST}/products`, {
  headers: getAuthHeaders(),
  params
})
}
}

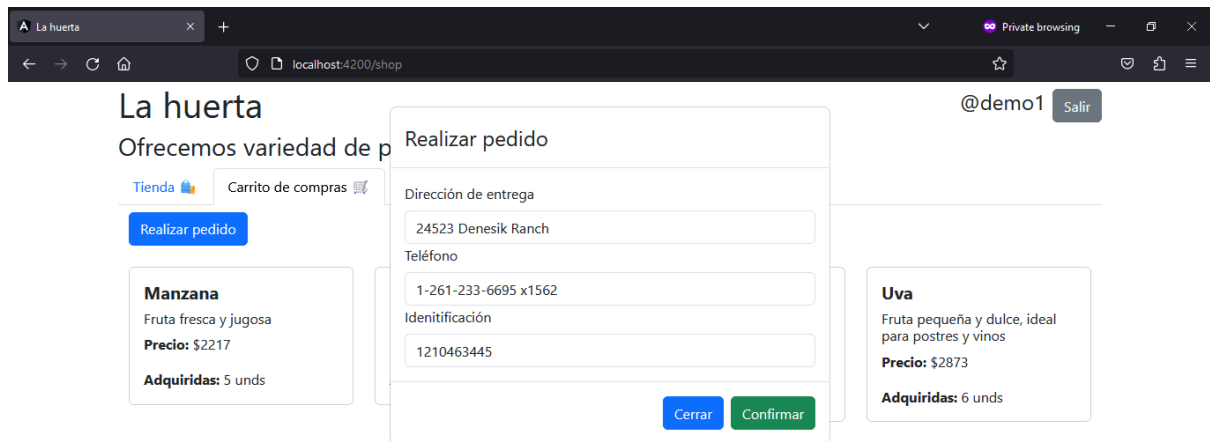
```

Página de carrito de compras

La segunda es una pantalla donde el usuario puede visualizar los productos que se han agregado al carrito de compras, también es donde se pueden generar una compra con dicho carrito



Pantalla con el carrito de compras



Modal de confirmación de orden

```
export class ShopService {

  constructor(
    private http: HttpClient
  ) { }

  /**
   * Método que realiza la petición al backend para agregar un producto al carrito
   */
  addShoppingCartItem(productId: string, desiredUnits: string) {
    return this.http.post(`${BACKEND_HOST}/cart`, { productId, desiredUnits }, {
      headers: getAuthHeaders()
    })
  }

  /**
   * Método que realiza la petición al backend para obtener los productos del carrito
   */
  getShoppingCartItems() {
    return this.http.get<ShoppingCartItem[]>(`${BACKEND_HOST}/cart/items`, {
      headers: getAuthHeaders()
    })
  }
}
```

```
export class OrderService {

  constructor(private http: HttpClient) { }

  /**
   * Método que realiza la petición al backend para obtener el listado de órdenes
   */
  getOrders() {
    return this.http.get<Order[]>(`${BACKEND_HOST}/orders`, {
```

```

        headers: getAuthHeaders()
    })
}

/**
 * Método que realiza la petición al backend para obtener el listado de productos de una orden
 * (Para implementar más adelante)
 */
getOrderItems(orderId: string) {
    const params = new HttpParams()
    return this.http.get<OrderItem[]>(`${BACKEND_HOST}/order/items`, {
        headers: getAuthHeaders(),
        params: params.set('order', orderId)
    })
}

/**
 * Método que realiza la petición al backend para crear una orden
 */
createOrder(orderData: any) {
    return this.http.post(`${BACKEND_HOST}/order`, orderData, {
        headers: getAuthHeaders()
    })
}

/**
 * Método que realiza la petición al backend para aprobar una orden
 */
approveOrder(orderId: string) {
    return this.http.put(`${BACKEND_HOST}/order`,
        { status: 'APPROVED', orderId },
        { headers: getAuthHeaders() }
    )
}

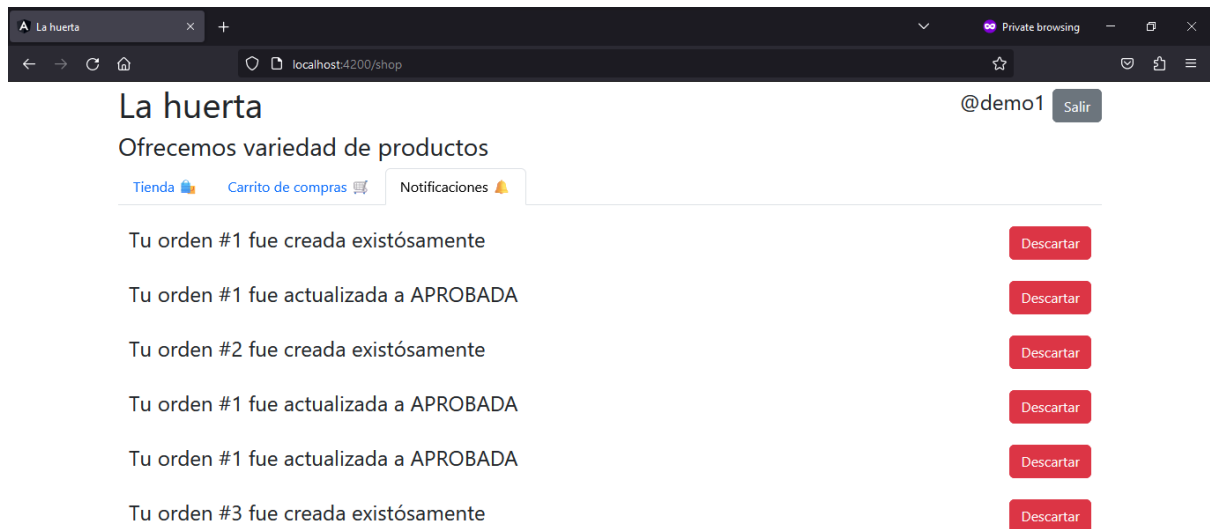
/**
 * Método que realiza la petición al backend para rechazar una orden
 */
rejectOrder(orderId: string) {
    return this.http.put(`${BACKEND_HOST}/order`,
        { status: 'REJECTED', orderId },
        { headers: getAuthHeaders() }
    )
}
}

```

Página de notificaciones

La tercera es la pantalla de notificaciones del usuario donde se podrán visualizar las notificaciones respecto a creación y actualización de órdenes.

También es posible borrar dichas notificaciones.



Pantalla de notificaciones

```
export class NotificationService {  
  
  constructor(  
    private http: HttpClient  
  ) { }  
  
  /**  
   * Método que realiza la petición al backend para obtener el listado de notificaciones  
   */  
  getNotifications() {  
    return this.http.get<Notification[]>(`${BACKEND_HOST}/notifications`, {  
      headers: getAuthHeaders()  
    })  
  }  
  
  /**  
   * Método que realiza la petición al backend para eliminar una notificación  
   */  
  discardNotification(notificationId: string) {  
    const params = new HttpParams()  
    return this.http.delete(`${BACKEND_HOST}/notification`, {  
      headers: getAuthHeaders(),  
      params: params.set('notificationId', notificationId)  
    })  
  }  
}
```

Creación de la página principal del administrador

Se emplearon los siguientes comandos para la creación de los elementos necesarios

```
# Para generar la estructura del componente Admin
ng generate component admin
```

La página principal del administrador consiste en un apartado dividido en dos secciones

Página de administración de productos

En esta pantalla se listan los productos ofrecidos en la tienda.

⚠ Solo esta implementada el listado de productos, la edición de los mismos fue un ejercicio llevado a cabo en clases y NO se incluyó en este taller.

La huerta

+

localhost:4200/admin

☆

🔍

📧

☰

Administración

@demo0

Salir

Productos

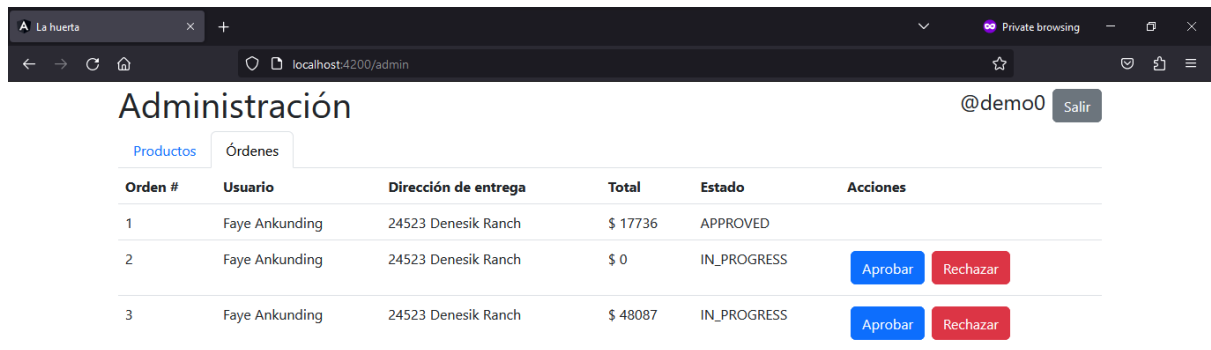
Órdenes

Nombre	Detalle	Precio	Unidades disponibles		
Manzana	Fruta fresca y jugosa	\$ 2217	45 unds	Editar	Eliminar
Plátano	Fruta rica en potasio	\$ 2385	60 unds	Editar	Eliminar
Sandía	Fruta refrescante para el verano	\$ 4106	51 unds	Editar	Eliminar
Zanahoria	Hortaliza saludable y rica en betacaroteno	\$ 2956	63 unds	Editar	Eliminar
Tomate	Fruto versátil para diferentes preparaciones culinarias	\$ 2305	69 unds	Editar	Eliminar
Lechuga	Hojas verdes y crujientes para ensaladas	\$ 2053	72 unds	Editar	Eliminar
Piña	Fruta tropical jugosa y dulce	\$ 3840	59 unds	Editar	Eliminar
Papa	Tubérculo versátil para guarniciones y platos principales	\$ 2465	64 unds	Editar	Eliminar

Pantalla de administración de productos

Pantalla de administración de órdenes

Esta segunda es una pantalla donde el administrador puede visualizar las órdenes creadas por los usuarios así como aprobarlas o rechazarlas.



Pantalla de administración de órdenes