

Proyecto Lenguajes de Programación

Título del Proyecto:

Aplicación de compra/venta de bienes o servicios - TiendaSegura

Integrantes del Grupo: G10

- Sebastian Holguin Vargas
- Derian Baque Choez
- Carlos Ronquillo Brush

1. Problemática a resolver

La mayoría de los locales comerciales / tiendas están siendo forzadas a cerrar debido a la ola de inseguridad y extorsiones (vacunas) que se viven día a día en el país. Los negocios más afectados son las tiendas de barrio o locales para las comunidades por su fácil exposición a la inseguridad ya que no cuentan con seguridad privada y el modo de atención al cliente es el tradicional abierto al público, esto expone al tendero a ser identificado, amenazado y por último una víctima más de inseguridad y extorsión. Actualmente estas tiendas no pueden operar bajo pedido o modalidad de “tienda oculta a puerta cerrada”.

Este proyecto busca preservar la economía barrial y proteger la vida de los comerciantes y clientes. Al trasladar toda la interacción de compra / venta a un entorno digital y seguro, se elimina la necesidad de tener la reja abierta y reduce el riesgo de extorsión.

2. Objetivos Propuestos

Objetivo General:

- Ofrecer una solución digital segura y adaptable que permita a los comercios de barrio operar bajo la modalidad de “tienda oculta a puerta cerrada”, minimizando la exposición a la inseguridad y extorsión.

Objetivos Específicos:

- Elaborar un sistema de gestión de catálogo que permita al tendero cargar sus productos y mantener la oferta de su negocio actualizada, este proceso no deberá tomar más de 1 minuto por producto.
- Desarrollar funcionalidades de búsqueda, secciones de detalle y compra de los productos en la aplicación del cliente.
- Implementar un sistema de autenticación para tendero, vecinos y/o clientes, garantizando el acceso a la plataforma de manera segura.

- Simular un botón de emergencia en caso de que ocurra alguna amenaza para el tendero.

3. Alcance del Proyecto

Incluye:

- App para el Tendero: Gestiona pedidos, catálogo, botón de pánico al 911.
- App para el Vecino / Cliente: Visualiza catálogo y pedidos, incluye carrito de compras y simulación de pagos.

No incluye:

- Integración de sistemas de cobros bancarios.
- Lógica de mapa para delivery.

4. Características de la Solución

1. Gestor de productos e inventario para tiendas pequeñas.
2. Búsqueda automática de imágenes de productos con base al nombre y marca del mismo.
3. Botón de pánico al 911.

5. Requerimientos Funcionales Asignados

Requerimiento	Descripción	Responsable
Crear un producto (escritura)	Esta funcionalidad permite al tendero crear productos y cargarlos en el catálogo disponible de la tienda con la siguiente información x producto: stock, nombre, categoría, precio, imagen del producto.	Sebastian Holguin
Ver pedidos de los clientes (lectura y escritura)	Esta función permite al tendero ver los pedidos de los clientes por orden de llegada (fecha) y aceptar o rechazar el pedido si así el tendero lo considera.	Sebastian Holguin
Buscar imágenes de productos automáticamente (scraping)	Esta opción permite al tendero cargar una imagen del producto automáticamente con base al nombre y marca del mismo. Además de tener la opción de aceptar la sugerencia de la	Sebastian Holguin

	imagen o tomar por sí mismo una fotografía del producto.	
Usar carrito de compra (escritura)	Esta función permite al cliente agregar productos al carrito de compras y almacenar dicha información en la base de datos MongoDB, validando un límite máximo de 25 ítems por carrito.	Derian Baque
Visualizar y actualizar el total de la lista de productos carrito de compra (lectura y escritura)	Esta función permite consultar el carrito almacenado en la base de datos, calcular dinámicamente el total de los productos y actualizar el valor total del carrito en la base de datos, mostrando al cliente el precio final.	Derian Baque
Consultar productos y tiendas cercanas. (lectura)	Esta función permite buscar productos y tiendas disponibles. El cliente podrá buscar los productos a través de una barra de búsqueda ingresando el nombre del producto, marca o categoría.	Carlos Ronquillo
Guardar historial de búsqueda y compras (lectura y escritura)	El usuario puede consultar su historial de compras clasificando las compras por fecha y tienda.	Carlos Ronquillo

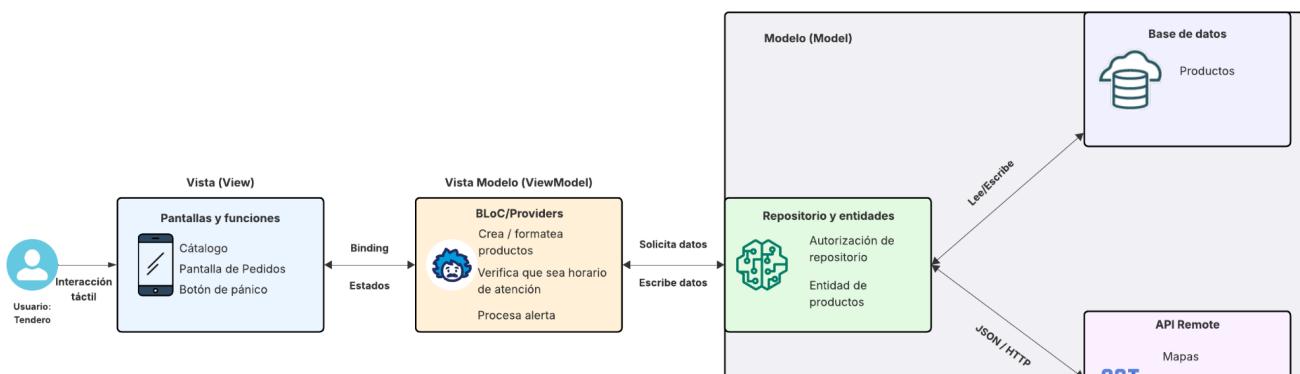
7. Arquitectura de la Aplicación

Tipo de Aplicación: Móvil

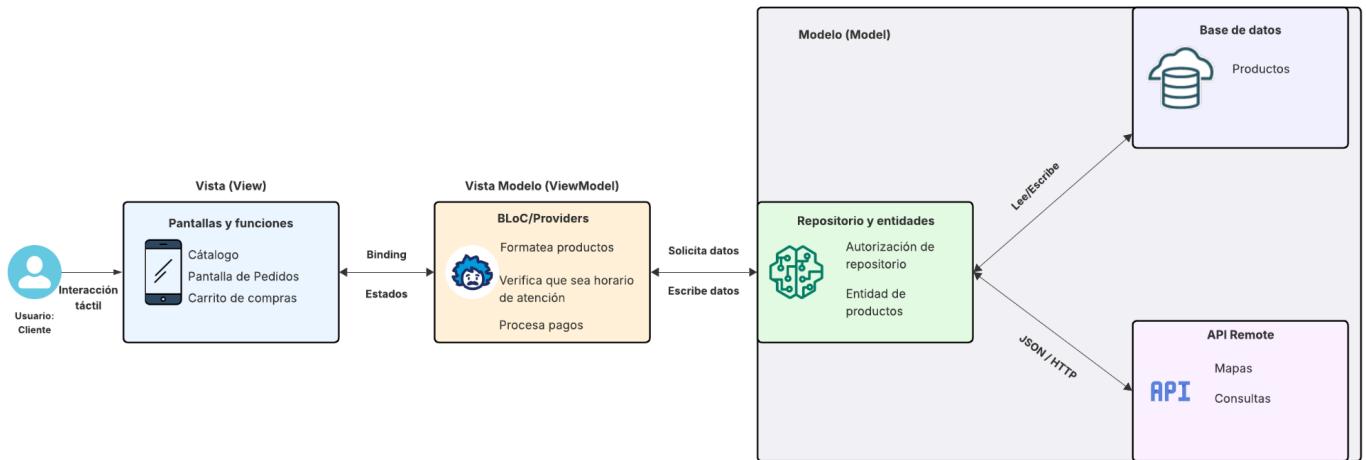
Las tiendas de barrios, especialmente de comunidades marginadas, los tenderos suelen tener celulares de gama media-baja y requieren de movilidad constante dentro del local además de que incluye la facilidad de tomar fotos. Una aplicación móvil permite el uso de notificaciones a los usuarios por lo que el tipo de aplicación móvil es la ideal.

Patrón de Arquitectura: MVVM

Tendero



Cliente/Vecino



Herramientas y Frameworks:

- Frameworks:
 - Flutter: Se debe desplegar la aplicación en Android que es la marca de celulares dominante en las comunidades y también para desplegar la app en simultáneo en sistemas iOS.
 - NodeJs: Indispensable para el backend con base en JavaScript/TypeScript.
- API:
 - Google, para integrar consultas de Google Maps e incluso el uso de Gemini y búsqueda de imágenes integradas.
- Servicios:
 - Autenticación: Firebase para autenticación de credenciales.
 - Base de datos: MongoDB (NoSQL) para la flexibilidad en la subida y cambios de productos en la tienda.

8. Lenguajes de Programación

Back-end:

- **TypeScript - JavaScript (Conocido)**
 - **Pros:**
 - Fácil de leer
 - Tipo de datos:** TypeScript usa tipado fuerte sobre JavaScript. Esto facilita definir interfaces estrictas.

- **Confiable:** Como es de tipado fuerte detecta los errores de datos antes de compilar.
- Fácil de escribir
- **Abstracción:** El lenguaje posee un nivel alto de abstracción mediante clases y módulos, además que cuenta con el ecosistema NPM.
- o **Contras:**
- Dificultad para leer
 - Al ser un superconjunto de JavaScript, a veces permite referencias mutables que pueden llevar a confusiones si no se configuran las reglas en TSConfig.
 - Simplicidad: El manejo de promesas asíncronicas

Front-end:

- **Dart (no conocido)**
 - o **Pros:**
 - Fácil de escribir
 - Expresividad:** Dart es muy expresivo al momento de escribir interfaces. Al declarar los componentes esto permite describir visualmente la jerarquía de la pantalla en el propio código.
 - Ortogonalidad:** Dart es muy consistente, la declaración de componentes/widgets es reutilizable durante todo el código.
 - **Fácil de leer:**
 - Tipo de datos:** Dart utiliza seguridad nula sólida, esto garantiza que las variables no puedan ser nulas, eliminando errores comunes.
 - o **Contras:**
 - Dificultad para leer:
 - Diseño de sintaxis:** Debido a su estructura de árbol de widgets, el código tiene indentación profunda y muchos paréntesis de cierre, lo cual afecta un poco en la legibilidad.
 - Simplicidad:** Para nuestro grupo que viene de ver la materia de Desarrollo Web Y Aplicaciones Móviles puede afectar un poco en la curva de aprendizaje ya que Dart mezcla lógica y diseño en el mismo archivo.

9. Prototipo de Baja Fidelidad

Herramienta de Diseño Utilizada: Figma

Capturas:

Cliente

The screenshots illustrate the user flow for a client in the TiendaSegura app:

- Inicio (Home Screen):** Shows a large placeholder for a profile picture. Below it are fields for "correo" (email) and "contraseña" (password), and checkboxes for "Recuérdame" (Remember me) and "Olvidé la contraseña" (Forgot password). A "Iniciar sesión" (Log in) button is at the bottom.
- Selección Rol (Role Selection):** A modal dialog titled "Seleccione Rol" (Select Role) offers two options: "Tendero" (Seller) and "Cliente" (Client). Each option has a description and a "Vamos a" button ("Let's go").
- Tienda Cercanas (Nearby Stores):** A search bar at the top shows "Panamá, 090313 Guayaquil". Below it is a section titled "Tienda Cercanas" (Nearby stores) featuring two store cards: "DISTRIBUIDORA ROSITA" (A 15 metros a la redonda) and "Tienda don Leo" (A 10 metros a la redonda). Each card includes a "Ver Catálogo" (View Catalog) button.
- Tienda Rosita (Rosita Store):** A store page for "Tienda Rosita" in Panamá, 090313 Guayaquil. It features a search bar and tabs for "Todos (2)", "Bebidas (1)", and "Snacks (1)". The "Bebidas" tab is selected, showing items like "Agua" (600 ml) for \$0.50 and "Coca Cola" (300 ml) for \$0.80, each with an "Agregar al carrito" (Add to cart) button.
- Detalle Producto (Product Detail):** A zoomed-in view of a "Salticas" snack product. The product image is a bag of Salticas cookies. The price is \$0.50. Below the image is a description: "Descripción: Galletas saladas crujientes ideales para snacks rápidos." A "Agregar al carrito" (Add to cart) button is centered below the price.
- Carrito de Compras (Shopping Cart):** A summary screen showing the total amount of \$2.00. It lists the items added from the previous screens: "Agua" (1 item) and "Salticas" (1 item). At the bottom are navigation icons for "Home", "Cart", and "Profile".

Tendero

The image displays four screenshots of the Tendero mobile application, showing the login screen, role selection, product management interface, and a product registration form.

1. Inicio (Login Screen): Shows a large user icon placeholder. Below it are fields for 'correo' (email) and 'contraseña' (password), along with 'Iniciar sesión' (Sign in) and 'Recuérdame' (Remember me) buttons.

2. Inicio (Role Selection): Shows a 'Selección Rol' (Role Selection) screen with two options: 'Tendero' (Seller) and 'Cliente' (Client). Each option has a description and a button: 'Vengo a vender' (I come to sell) for Seller and 'Vengo a comprar' (I come to buy) for Client.

3. Mis Productos (Product Management): Shows a list of products under categories 'Bebidas' (Drinks) and 'Snacks'. Each product card includes an image, name, price, and an 'Editar Producto' (Edit Product) button.

Categoría	Producto	Imagen	Precio	Opción
Bebidas	Agua		\$ 0.50	Editar Producto
Bebidas	Coca Cola		\$ 0.80	Editar Producto
Snacks	Salticas		\$ 0.50	Editar Producto
Snacks	Ruffles		\$ 0.50	Editar Producto

4. Registrar nuevo producto (New Product Registration): Shows a form with fields for 'Nombre del producto:' (Product name), 'Precio:' (Price), 'Categoría' (Category), 'Descripción:' (Description), and 'Imagen del producto:' (Product image). It includes a search bar ('Búsqueda auto.') and a 'Agregar producto' (Add product) button at the bottom.

10. Implementación

10.1 Backend

Implementación	Responsable	Estado
Crear un producto	Sebastian Holguin	✓ Hecho
Ver pedidos de los clientes	Sebastian Holguin	✓ Hecho
Buscar imágenes (scraping)	Sebastian Holguin	✓ Hecho
Agregar productos al carrito de compra	Derian Baque	✓ Hecho
Visualizar y actualizar el total del carrito de compra	Derian Baque	✓ Hecho
Consultar productos y tiendas cercanas. (lectura)	Carlos Ronquillo	✓ Hecho
Guardar historial de búsqueda y compras (lectura y escritura)	Carlos Ronquillo	✓ Hecho

10.1.1 Evidencia

Crear un Producto y agregarlos al catálogo

HTTP TiendaSegura / New Request

POST http://localhost:3000/api/products/create

Body **raw**

```
1 {
2   "name": "Laptop Dell Inspiron",
3   "description": "Laptop de alto rendimiento con procesador Intel i7",
4   "brand": "Dell",
5   "price": 899.99,
6   "stock": 15,
7   "imageUrl": "https://example.com/images/laptop-dell.jpg",
8   "isActive": true
9 }
```

Body Cookies Headers (8) Test Results

201 Created 220 ms 604 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "name": "Laptop Dell Inspiron",
3   "description": "Laptop de alto rendimiento con procesador Intel i7",
4   "brand": "Dell",
5   "price": 899.99,
6   "stock": 15,
7   "imageUrl": "https://example.com/images/laptop-dell.jpg",
8   "isActive": true,
9   "_id": "694b41bc84f347478cd085ce",
10  "createdAt": "2025-12-24T01:28:28.259Z",
11  "updatedAt": "2025-12-24T01:28:28.259Z",
12  "__v": 0
13 }
```

POST para crear un producto

HTTP TiendaSegura / New Request

POST http://localhost:3000/api/products/694a7787297261bd755fe50b/add-to-catalog/694b41bc84f347478cd085ce

Body **raw**

```
1 {
2   "name": "Laptop Dell Inspiron",
3   "description": "Laptop de alto rendimiento con procesador Intel i7",
4   "brand": "Dell",
5   "price": 899.99,
6   "stock": 15,
7   "imageUrl": "https://example.com/images/laptop-dell.jpg",
8   "isActive": true
9 }
```

Body Cookies Headers (8) Test Results

200 OK 386 ms 493 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "_id": "694a7787297261bd755fe50b",
3   "products": [
4     "694a75ad297261bd755fe500",
5     "694a75ad297261bd755fe502",
6     "694b41bc84f347478cd085ce"
7   ],
8   "isActive": true,
9   "shop": "694a7a20297261bd755fe512",
10  "updatedAt": "2025-12-24T02:16:06.170Z",
11  "__v": 1
12 }
```

POST para agregar un producto al catálogo

Crear una orden de compra

The screenshot shows a Postman interface with the following details:

- HTTP Method:** POST
- URL:** http://localhost:3000/api/orders/create
- Body:** raw JSON (selected)
- Request Body:**

```
1 {  
2   "clientId": "694a758f297261bd755fe4fc",  
3   "shoppingCartId": "694a8642af633aa8a1fcfdc2a"  
4 }
```

- Response Status:** 201 Created
- Response Time:** 507 ms
- Response Size:** 490 B
- Response Body:**

```
1 {  
2   "_id": "694b5ae1578cc715bb7acb16",  
3   "client": "694a758f297261bd755fe4fc",  
4   "shoppingCart": "694a8642af633aa8a1fcfdc2a",  
5   "status": "Pending",  
6   "createdAt": "2025-12-24T03:15:45.055Z",  
7   "updatedAt": "2025-12-24T03:15:45.055Z",  
8   "__v": 0  
9 }
```

POST para agregar un orden de compra

Ver una orden de compra de un cliente

The screenshot shows a Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:3000/api/orders/client/694a758f297261bd755fe4fc
- Params:** (selected)
- Response Status:** 200 OK
- Response Time:** 159 ms
- Response Size:** 487 B
- Response Body:**

```
1 [  
2   {  
3     "_id": "694b5ae1578cc715bb7acb16",  
4     "client": "694a758f297261bd755fe4fc",  
5     "shoppingCart": "694a8642af633aa8a1fcfdc2a",  
6     "status": "Pending",  
7     "createdAt": "2025-12-24T03:15:45.055Z",  
8     "updatedAt": "2025-12-24T03:15:45.055Z",  
9     "__v": 0  
10   }  
11 ]
```

Get para consultar la orden de los clientes

Visualizar productos, agregar productos al carrito de compra y actualizar el total del carrito de compra

The screenshot shows the Postman interface with a collection named "TiendaSegura". A GET request is made to `http://localhost:3000/api/lists/`. The response status is 200 OK, and the response body is a JSON object representing a shopping cart:

```
1 [  
2   {  
3     "_id": "694a7d4caf633aa8a1fc1f",  
4     "products": [  
5       "694a75ad297261bd755fe500",  
6       "694b41bc84f347478cd085ce"  
7     ],  
8     "price": 901.24,  
9     "isActive": true,  
10    "updatedAt": "2025-12-24T02:47:31.735Z",  
11    "__v": 7  
12  }  
13 ]
```

Get para obtener información del carrito de compras

The screenshot shows the Postman interface with the same collection. A POST request is made to `http://localhost:3000/api/products/694a7d4caf633aa8a1fc1f/add-to-list/694a75ad297261bd755fe500`. The response status is 200 OK, and the response body is the updated shopping cart:

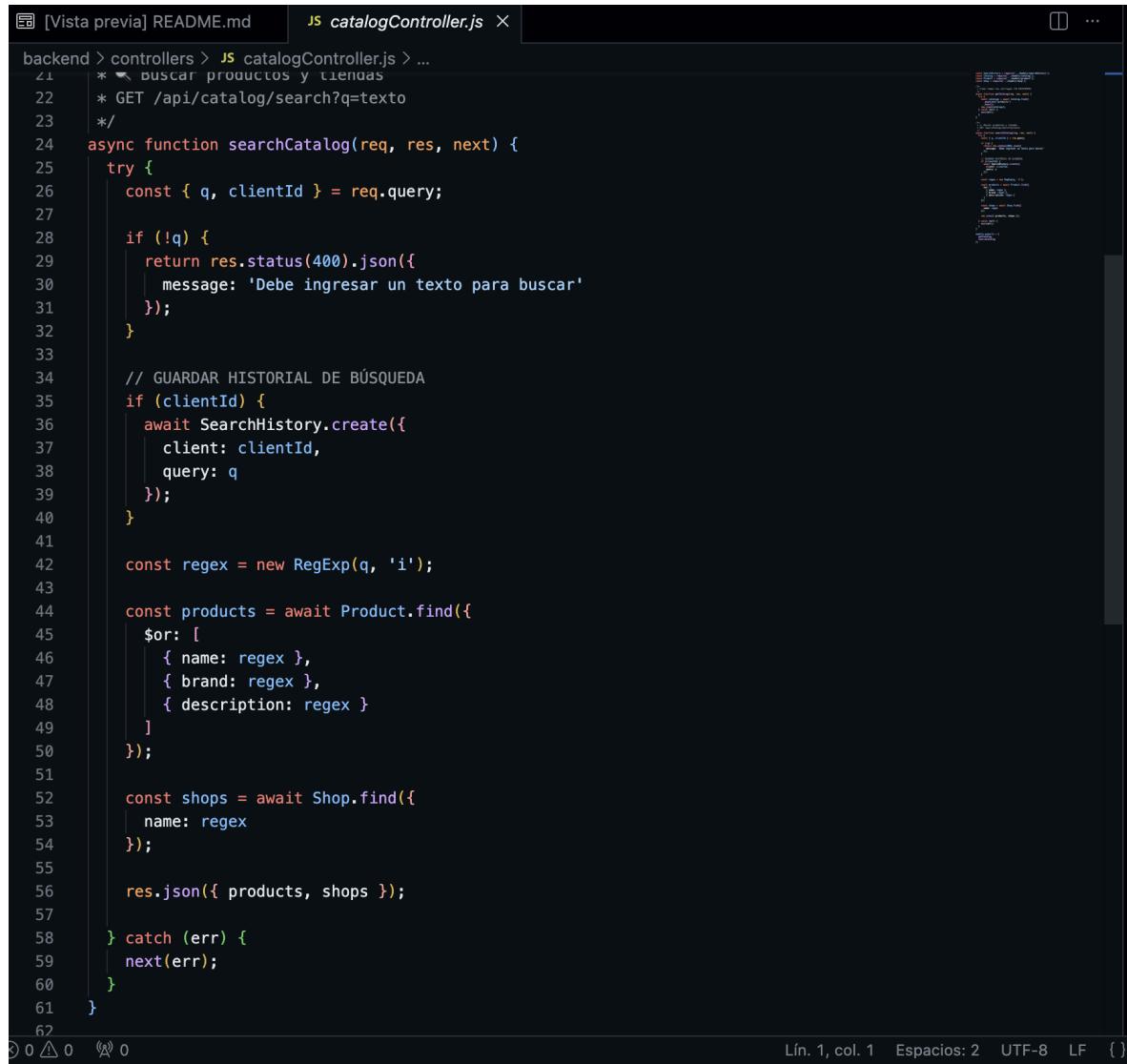
```
1 {  
2   "_id": "694a7d4caf633aa8a1fc1f",  
3   "products": [  
4     "694a75ad297261bd755fe500",  
5     "694b41bc84f347478cd085ce",  
6     "694a75ad297261bd755fe501"  
7   ],  
8   "price": 901.24,  
9   "isActive": true,  
10  "updatedAt": "2025-12-24T02:51:09.138Z",  
11  "__v": 8  
12 }
```

Post para agregar un producto mediante id al carrito de compras

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (Derian Fernando Baque Choez's W...), 'Environments' (My Collection, TiendaSegura selected), 'History', 'Flows', and 'Files (BETA)'. The main area has tabs for 'Overview', 'GET https://pokeapi.co/api/v2/pokemon?limit=10', 'GET productos' (selected), and 'GET productos'. A search bar at the top says 'Search Postman' and there are 'Ctrl + K' and 'Invite' buttons. The central part shows a 'TiendaSegura / productos' request with a status of '9:53 PM X'. It's a 'GET' request to 'http://localhost:3000/api/lists/'. Below it are tabs for 'Docs', 'Params' (selected), 'Authorization', 'Headers (6)', 'Body', 'Scripts', and 'Settings'. Under 'Params', there's a table for 'Query Params' with columns 'Key', 'Value', 'Description', and 'Bulk Edit'. The 'Body' tab shows a JSON response with a status of '200 OK' and a timestamp of '9:53 PM'. The JSON data is as follows:

```
1 [  
2 {  
3   "_id": "694a7d4caf633aa8a1fcfdc1f",  
4   "products": [  
5     "694a75ad297261bd755fe500",  
6     "694b41bc84f347478cd085ce",  
7     "694a75ad297261bd755fe501"  
8   ],  
9   "price": 903.34,  
10  "isActive": true,  
11  "updatedAt": "2025-12-24T02:53:57.125Z",  
12  "_v": 8  
13 }]  
14 ]
```

Get para obtener información actualizada del carrito de compras



```

backend > controllers > JS catalogController.js > ...
  * 🔍 buscar productos y tiendas
  * GET /api/catalog/search?q=texto
  */
24 async function searchCatalog(req, res, next) {
25   try {
26     const { q, clientId } = req.query;
27
28     if (!q) {
29       return res.status(400).json({
30         message: 'Debe ingresar un texto para buscar'
31       });
32     }
33
34     // GUARDAR HISTORIAL DE BÚSQUEDA
35     if (clientId) {
36       await SearchHistory.create({
37         client: clientId,
38         query: q
39       });
40     }
41
42     const regex = new RegExp(q, 'i');
43
44     const products = await Product.find({
45       $or: [
46         { name: regex },
47         { brand: regex },
48         { description: regex }
49       ]
50     });
51
52     const shops = await Shop.find({
53       name: regex
54     });
55
56     res.json({ products, shops });
57
58   } catch (err) {
59     next(err);
60   }
61 }
62

```

Lín. 1, col. 1 Espacios: 2 UTF-8 LF {}

Implementacion metodo searchCatalog

10.2 FrontEnd - Visualización

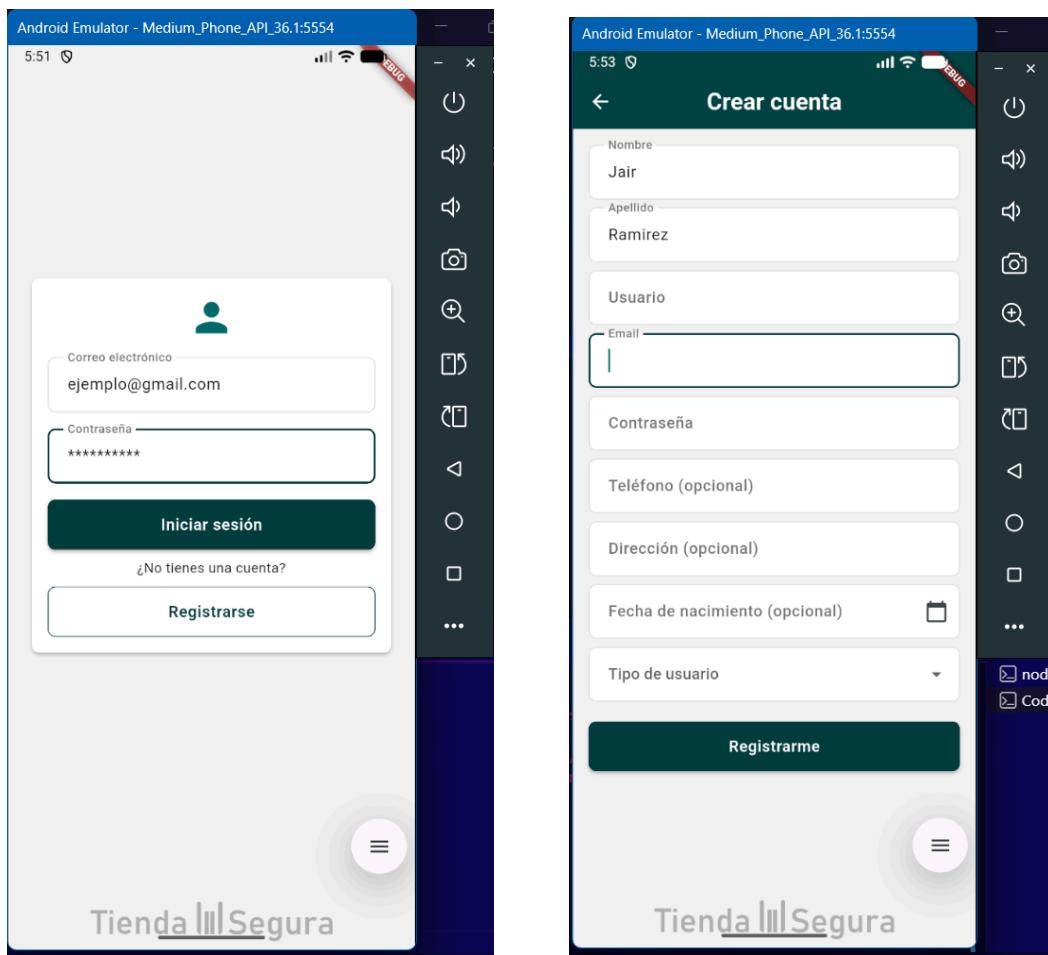
Implementación	Responsable	Estado
Login	Sebastian Holguin	✓ Hecho
Registro de usuario	Sebastian Holguin	✓ Hecho
Catálogo de productos	Derian Baque	✓ Hecho
Carrito de compras	Derian Baque	✓ Hecho

Registro de productos

Carlos Ronquillo

✓ Hecho

10.2.1 Evidencia



Login y Registro de usuario



Bebidas



Coca Cola
\$1.25

Agregar



Agua
\$0.60

Agregar



Coca Cola
\$1.25

Agregar



Agua
\$0.60

Agregar

Snacks



Ruffles
\$0.75

Agregar



Salticas
\$0.50

Agregar

Snacks



Categoría de productos

Tienda Segura

 Agua Cantidad: 1 \$0.60	- 1 +
 Coca Cola Cantidad: 1 \$1.25	- 1 +

Total a pagar: \$1.85

Tienda Segura

 Agua Cantidad: 4 \$2.40	- 4 +
 Coca Cola Cantidad: 2 \$2.50	- 2 +

Total a pagar: \$4.90

Carrito de Compras

Tienda Segura Tiendasegura

Bebidas

Registrar Producto

Nombre

Marca (opcional)

Precio (opcional)

Cantidad neta (opcional)

Unidad

URL imagen (opcional)

Crear producto

11 Link del repositorio

<https://github.com/Sebhvarg/tiendasegura.git>

12. Referencias

Flutter Team. (2024). *Flutter documentation*. Google.
<https://docs.flutter.dev>

Dart Team. (2024). *Dart language tour*. Google.
<https://dart.dev/guides>

Node.js Foundation. (2024). *Node.js documentation*. OpenJS Foundation.
<https://nodejs.org/en/docs>

Microsoft. (2024). *TypeScript handbook*.
<https://www.typescriptlang.org/docs>

MongoDB Inc. (2024). *MongoDB documentation: NoSQL database*.
<https://www.mongodb.com/docs>

Firebase Team. (2024). *Firebase authentication documentation*. Google.
<https://firebase.google.com/docs/auth>

Google Developers. (2024). *Google Maps Platform documentation*.
<https://developers.google.com/maps>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.

Fowler, M. (2003). *Patterns of enterprise application architecture*. Addison-Wesley.

Richards, M., & Ford, N. (2020). *Fundamentals of software architecture*. O'Reilly Media.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Statista Research Department. (2023). *Mobile operating system market share worldwide*.
<https://www.statista.com>

OWASP Foundation. (2023). *OWASP mobile top 10: Mobile security risks*.
<https://owasp.org/www-project-mobile-top-10/>