Diseño de Sistemas de Información

Integrante:

* Osvaldo Barcos

2024

Profesores:

* Pablo pioli
* Juan Pablo ferreyra

UTN San Francisco

Tabla de contenido

[Enunciado 2](#_Toc179387411)

[Requerimientos No Funcionales 3](#_Toc179387412)

[Requerimientos Funcionales 4](#_Toc179387413)

[Prototipos y flujo de datos 5](#_Toc179387414)

[Flujo de comunicación: 5](#_Toc179387415)

[Registrarte 6](#_Toc179387416)

[Inicio de Sesión 8](#_Toc179387417)

[Catálogo 10](#_Toc179387418)

[Cotización 12](#_Toc179387419)

[Carrito de compras 13](#_Toc179387420)

[Realizar compra 17](#_Toc179387421)

[Diagrama de Arquitectura 19](#_Toc179387422)

[Container 19](#_Toc179387423)

[Diagrama Entidad Relación 20](#_Toc179387424)

[Diagrama de Casos de Uso 21](#_Toc179387425)

# Enunciado

Una empresa dedicada a la fabricación de materiales para la construcción se encuentra distribuida en diferentes 3 plantas productivas, una oficina comercial y vendedores que atienden a clientes mayoristas en diferentes zonas. La sucursal A extrae materia prima que se utiliza como insumo en la planta C. La planta B elabora productos semi-terminados en base a alambres de acero que se utilizan para producir en la planta C. Por su parte, la planta C utiliza elabora ladrillos, vigas de cemento y bloques pre-armados de diferentes medidas. Desde la planta C se realiza el envío de los pedidos directamente al cliente. Cada planta productiva realiza ingresos de stock de materias primas, consulta de stock, generación de órdenes de producción de los diferentes productos y envío de productos a las diferentes plantas. Por decisión de la gerencia se necesita reducir los tiempos de atención a clientes minoristas, para ello se pretende ofrecer la posibilidad de cotizar y generar pedidos directamente en el sitio web de la empresa, para ello, una vez identificados los clientes podrán consultar los productos.

Podrán cotizar, ingresando cantidad de metros cuadrados a construir y tipos de materiales, en base a dicha información se debería poder determinar la cantidad de materiales necesarios, por ejemplo: Para construir un galpón de 40m x 40m, de 6m de altura, con ladrillo de tipo bloques de 18cm x 33cm se necesitaría cubrir una superficie de 960 metros cuadrados, con lo cual la cantidad de ladrillos, considerando una separación de 40 cm entre vigas, se necesitaría:

* 16161 ladrillos, equivalentes a 112,23 pallet.
* sImporte $ 6.302.790.-

Se debería poder gestionar los descuentos por cantidad, por ejemplo, a partir de los 10mil ladrillos ofrecer un 5% de descuento sobre el valor del producto. A partir de dicha cotización el cliente podrá realizar un pedido, debiendo completar información de domicilio de envío. La empresa cuenta con servicio de envío. Una vez aprobado el pedido, se acuerda una forma de pago. Una vez que el cliente realiza el pago se envía el pedido.

# Requerimientos No Funcionales

**1. Eficiencia (Performance Efficiency)**

**RNF1**: El sistema debe tener un tiempo de respuesta de ≤ 2 segundos para el 95% de las solicitudes realizadas (navegación del catálogo, carrito de compras).

**RNF2**: El sistema debe soportar al menos 1000 transacciones por segundo durante períodos de alta demanda.

**2. Escalabilidad**

**RNF3**: El sistema debe soportar un crecimiento hasta 5000 usuarios concurrentes sin afectar significativamente el rendimiento.

**3. Disponibilidad**

**RNF4**: El sistema debe tener una disponibilidad mínima del 99.9%, lo que implica un tiempo de inactividad máximo de 8.77 horas al año.

**RNF5**: El sistema debe contar con mecanismos de redundancia para evitar la pérdida de información crítica (pedidos, pagos, etc.).

**4. Usabilidad**

**RNF6**: Los usuarios deben poder realizar sus primeras cotizaciones en menos de 5 minutos.

**5. Seguridad**

**RNF7**: El sistema debe implementar autenticación mediante tokens (JWT) para asegurar que sólo los usuarios autorizados puedan acceder a sus cuentas y datos.

**6. Mantenibilidad**

**RNF8**: El código debe estar bien documentado y contar con guías de mantenimiento para que futuros desarrolladores puedan trabajar en él sin dificultades.

# Requerimientos Funcionales

**1. Gestión de usuarios**

**RF1:** Los usuarios deben poder registrarse proporcionando datos como nombre, correo electrónico y contraseña.

**RF2:** Los usuarios deben poder iniciar sesión utilizando sus credenciales.

**RF3**: El sistema debe permitir a los usuarios recuperar o restablecer su contraseña en caso de olvido.

**2. Carrito de compras y cotización**

**RF4**: El sistema debe permitir a los usuarios visualizar un catálogo de ladrillos disponibles para la construcción. El catálogo debe mostrar el nombre del producto, precio unitario, y otras especificaciones como tipo de ladrillo (hueco, cocido, etc.).

**RF5**: Los usuarios deben poder buscar productos y filtrar el catálogo según características como el tipo de ladrillo y el precio.

**RF6:** El sistema debe calcular automáticamente la cantidad de ladrillos necesaria según las dimensiones proporcionadas (largo, ancho, altura) del área a construir.

**RF7**: El sistema debe mostrar el carrito con los productos seleccionados, la cantidad calculada de ladrillos, el subtotal de cada producto y el total general.

**RF8**: Si la cantidad total de ladrillos supera un umbral (ej. 10,000 unidades), el sistema debe aplicar un descuento automático (ej. 5%).

**3. Método de pago**

**RF9**: El sistema debe permitir realizar pagos a través de la API de Mercado Pago, mostrando opciones de pago como tarjetas de crédito/débito y otros métodos.

**RF10:** Después de calcular el total del carrito (con o sin descuentos), los usuarios deben poder proceder al pago y obtener una confirmación de este.

**4. Entrega a domicilio**

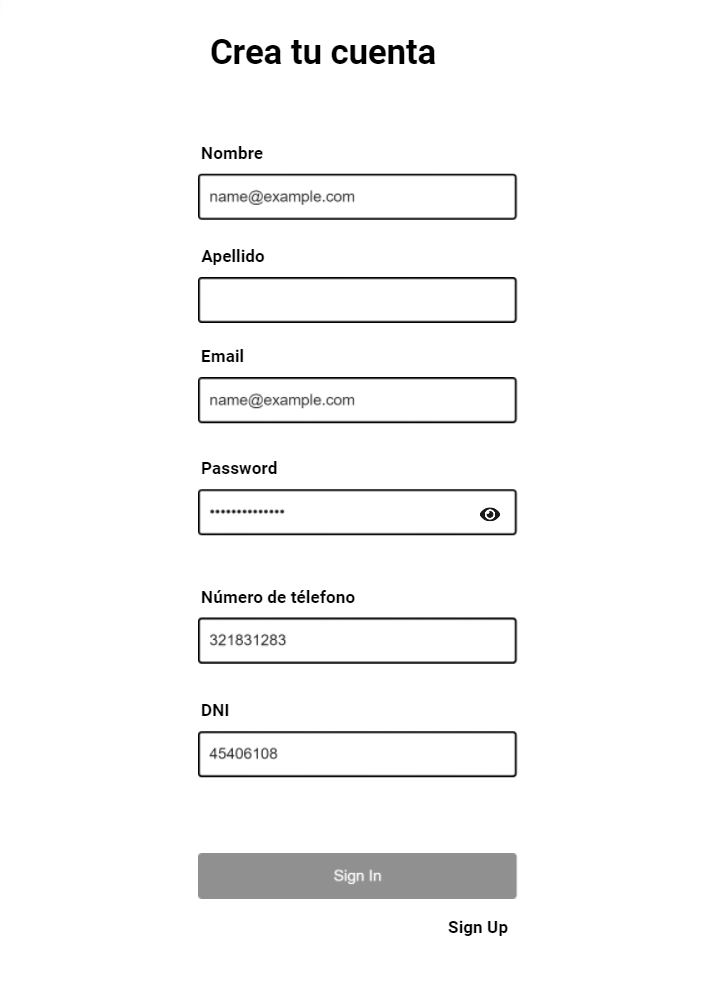
**RF11:** El usuario debe poder introducir o seleccionar su dirección de entrega.

# Prototipos y flujo de datos

## Flujo de comunicación:

1. **El frontend utiliza JavaScript** para crear una solicitud HTTP especificando el método, la URL del endpoint del backend y opcionalmente, un cuerpo de datos en formato JSON.
2. **La solicitud se envía al backend.**
3. **El backend recibe la solicitud,** procesa los datos JSON y realiza las operaciones necesarias.
4. **El backend envía una respuesta al frontend**, generalmente también en formato JSON.
5. **El frontend recibe la respuesta** y la procesa, actualizando la interfaz de usuario o realizando otras acciones según sea necesario.

## Registrarte





**-En caso de éxito:**

**Texto

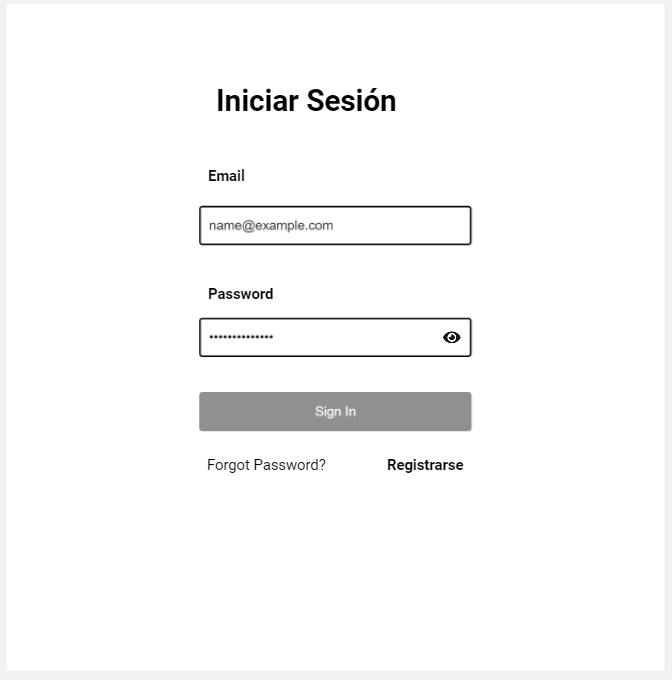
Descripción generada automáticamente**

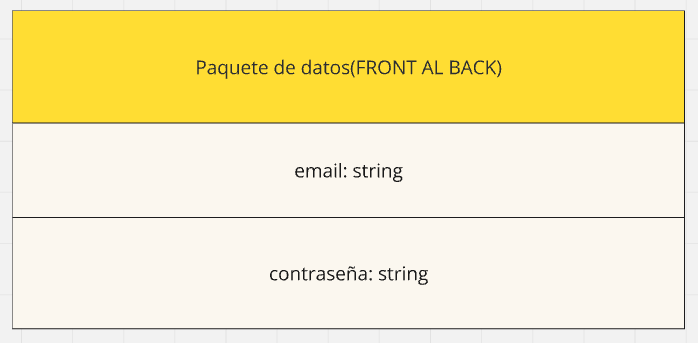
**-En caso de error (por ejemplo, si el email ya existe):**

**Texto

Descripción generada automáticamente**

## Inicio de Sesión





**En caso de éxito:**

**Texto

Descripción generada automáticamente**

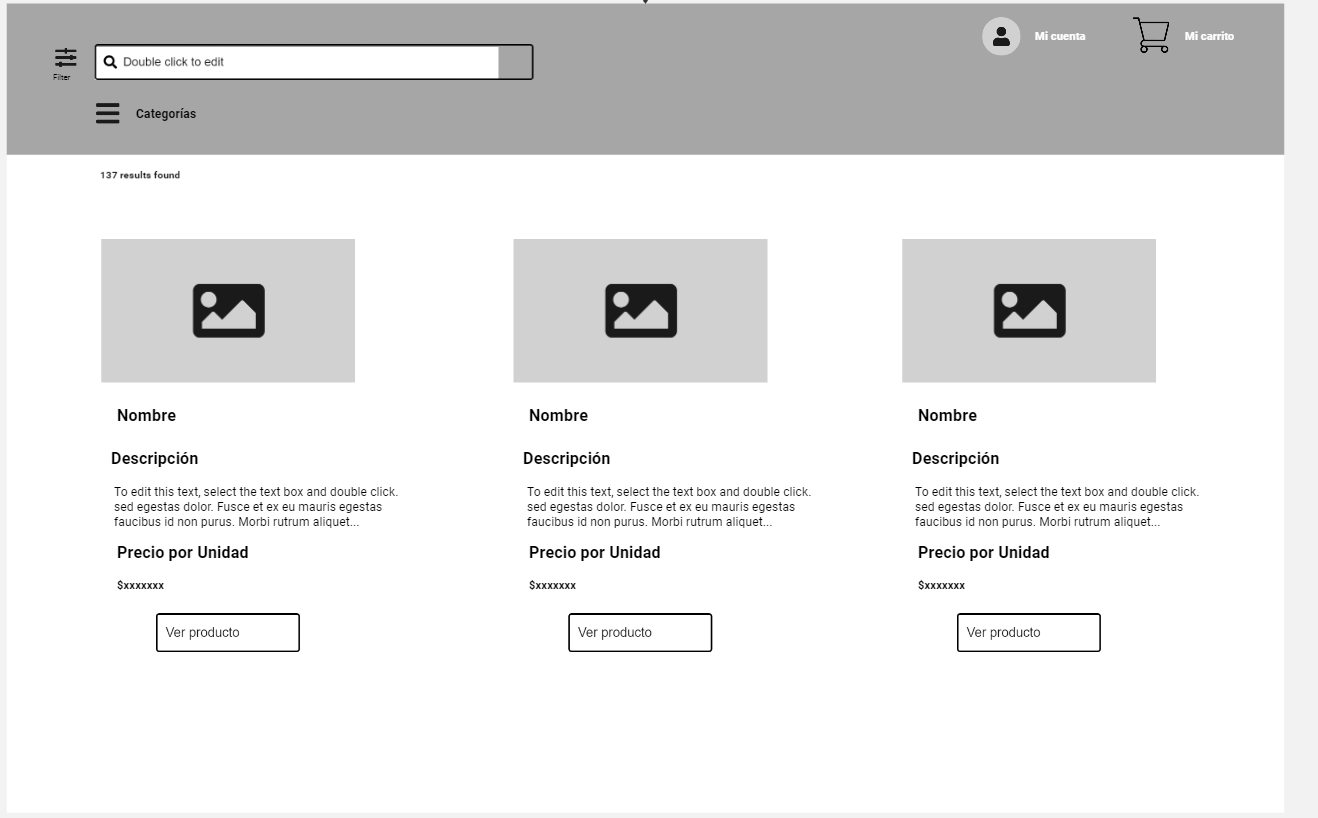
El **token** es el que el frontend utilizará para **mantener la sesión abierta** del usuario. Este token debe ser almacenado en el navegador (en localStorage, sessionStorage o una **cookie segura**).

**En caso de error:**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

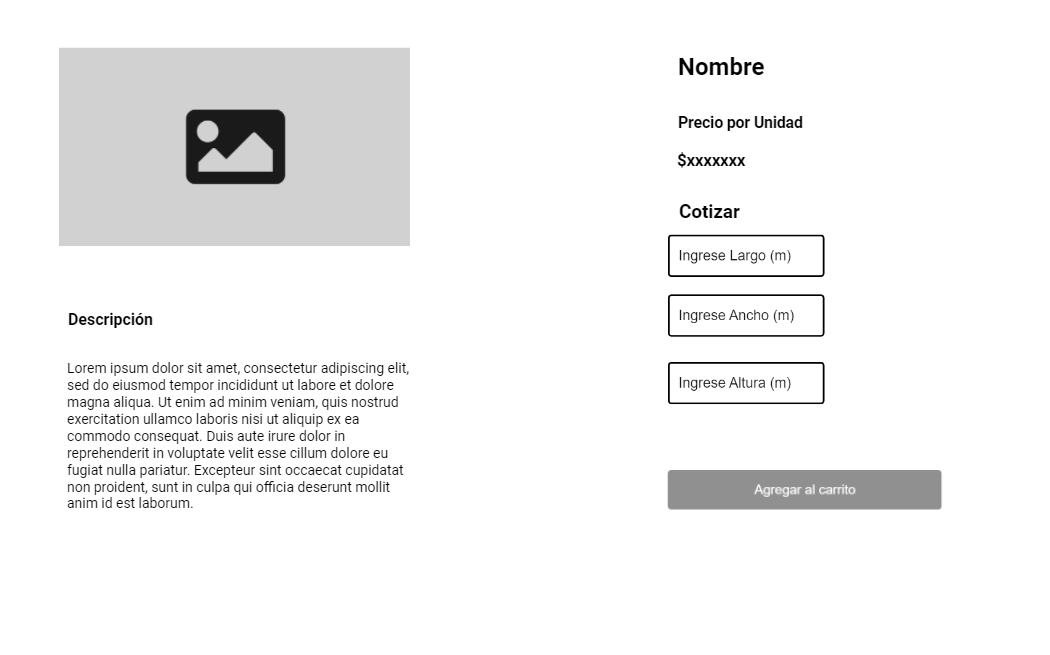
## Catálogo



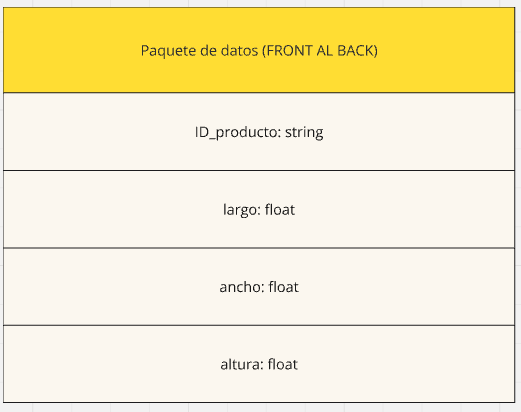


## Cotización

**Pantalla cuando se presiona en el botón ver producto**



**Datos enviados cuando se agrega al carrito el producto con las medidas ingresadas.**

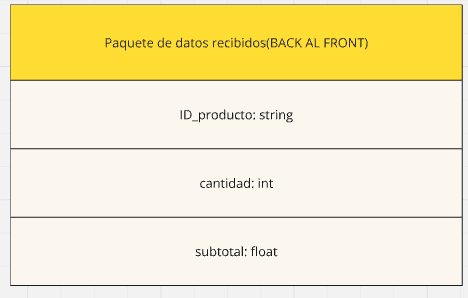


## Carrito de compras

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***- Cotización de cada producto cuando se agrega al carrito de compras.***



* **Cálculos para obtener los datos:**

-perimetro=2×(Largo+Ancho)=2×(40+40)=160 (m)

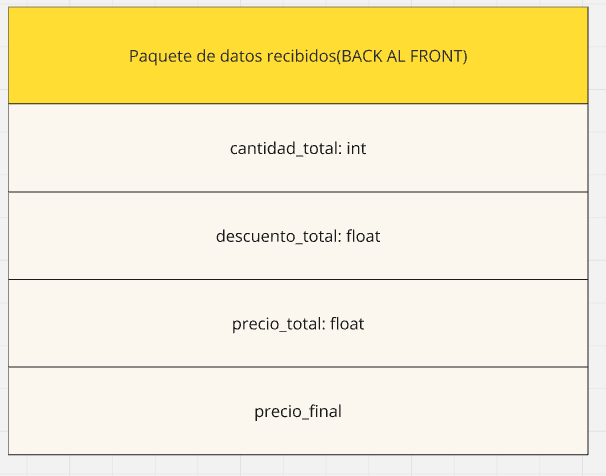
-A\_total = perimetro x altura =160 (m) × 6 (m) = 960 (m^2)

-A\_ladrillo = largo\_ladrillo x altura = 0,0594 (m^2)

-cantidad = A\_total/A\_ladrillo= 960 (m^2) / 0,0594 (m^2) = 16.161 ladrillos

-subtotal = cantidad x precio\_unitario = 16.161 x 390 = 6.302.790

***- Datos del resumen de compra.***



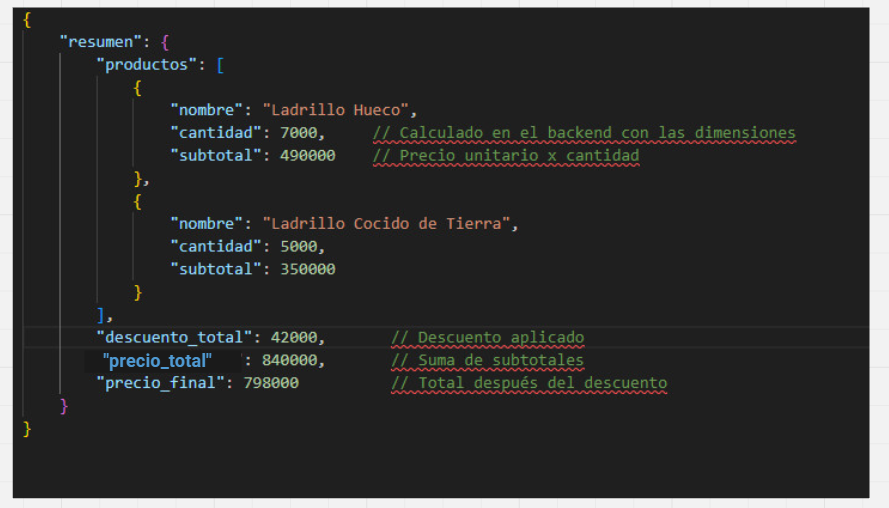
* **Cálculos para obtener los datos:**

**Texto

Descripción generada automáticamente**

***-JSON de ejemplo***

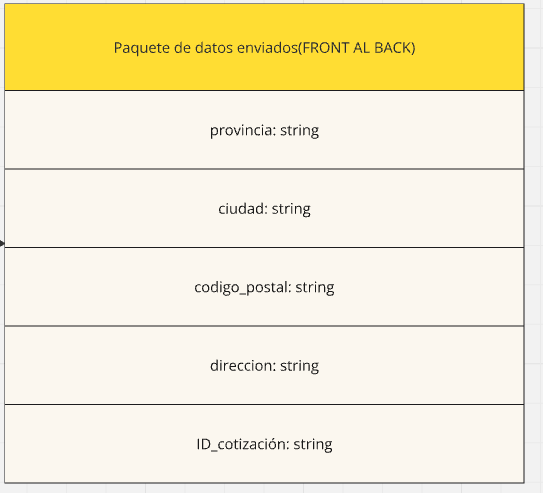
* Envío de todos los datos calculados en el backend.

****

## Realizar compra

***-Cargar datos del domicilio para la entrega***





***-Seleccionar método de pago (por ahora solo tarjetas)***



* **Integración de la API chekout Pro de Mercado pago.**

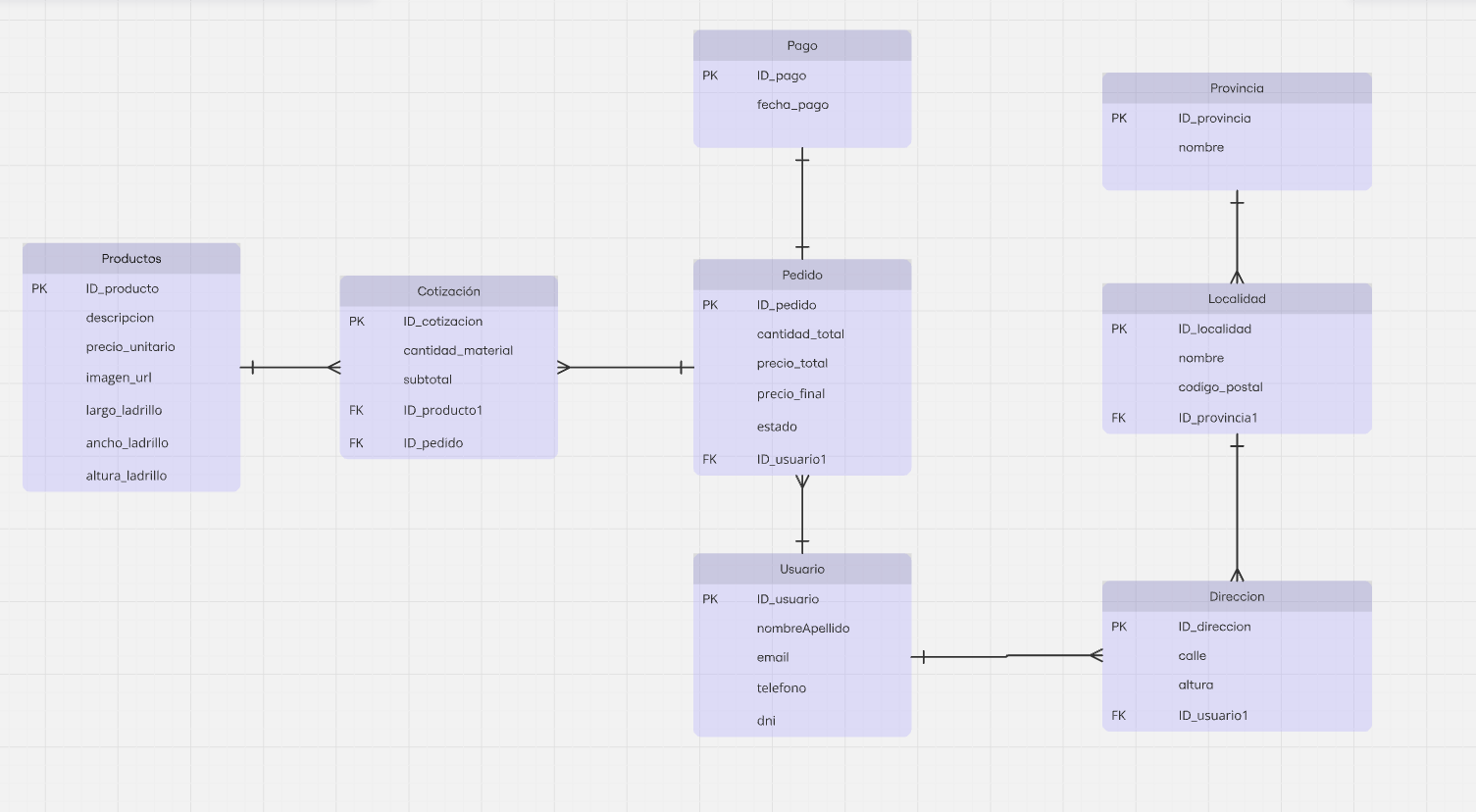
# Diagrama de Arquitectura

## Container

Diagrama

Descripción generada automáticamente

# Diagrama Entidad Relación



# Diagrama de Casos de Uso

