

TRABAJO PRÁCTICO N°3:



Docentes: Juan Pablo Ferreyra, Pablo Pioli

Alumno: Matías Battán.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Contenido

Enunciado del problema.....	3
Alcance del problema	5
Arquitectura.....	6
Tecnologías a implementar	7
Requerimientos de software.....	8
Datos con los que tratar	9
Prototipos(web).....	10
Prototipos(Móvil).....	26

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
Nº3

Trabajo Práctico



Enunciado del problema

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, ejemplo:

	<p>Ladrillo Hueco 12x18x33cm 9 tubos Precio por unidad: \$390,00 Descripción: Ladrillo hueco cerámico 12x18x33 cm 9 tubos Ladrillo de cerramiento Uso: Especiales para tabiques divisorios y cerramientos (ambientes interiores y muros de cierre). Cantidad por pallet: 144 unidades</p>
	<p>Viga 4 mts Precio por unidad: \$ 10619 Descripción: Ladrillo hueco cerámico 12x18x33 cm 9 tubos Ladrillo de cerramiento Uso: Especiales para tabiques divisorios y cerramientos (ambientes interiores y muros de cierre). Uso: Son utilizadas para techar en la construcción. Se colocan sobre las paredes y van acompañadas entre viga y viga por ladrillos para techo y malla sima.</p>

**INGENIERÍA EN SISTEMAS DE
INFORMACIÓN**

Diseño de Sistemas de Información
N°3

Trabajo Práctico

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 pallets
- Importe \$ 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.

Alcance del problema

El objetivo en este caso para este problema será agilizar los tiempos de atención a los clientes minoristas, cubriendo cotizaciones y envíos.

Para ello se propone una página web donde los clientes se identificarán y accederán al catálogo de productos y al sistema de cotización ingresando la cantidad de metros cuadrados a construir con los materiales seleccionados. El sistema calcula automáticamente la cantidad de materiales necesarios y genera los pedidos correspondientes. Además, se aplicarán descuentos cuando corresponda.

El pedido es generado por el sistema una vez el cliente acepta la compra de la cotización generada e ingresa la dirección de envío de los materiales. Una vez determinado la forma de pago y su confirmación, el pedido será enviado.

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Arquitectura

Para la estructura del sistema se usaría una arquitectura en capas, dividiendo las responsabilidades de la aplicación en distintas capas que se comunican entre sí. Las capas serían:

Frontend

Es la interfaz web con la que interactuaría el usuario. Figuran datos como el catálogo de productos, datos de los productos, perfil de usuario, carrito de compras, sección de cotización, sección de pago, etc.

Se desarrollará una app móvil multiplataforma utilizando React Native para iOS y Android, con una interfaz adaptada para gestionar pedidos, pagos y visualización de puntos.

En la pantalla de "Carrito", se agregará la opción de subir un archivo de AutoCAD, eliminando la necesidad de ingresar dimensiones manualmente para el cálculo de materiales.

Backend:

Es el encargado de la lógica del negocio, incluye la validación de usuario, el sistema de cotización, aplicación de descuentos, la gestión de pedidos, transacciones de pago, gestión de envíos.

Se implementa una integración con la plataforma de pagos MercadoPago para manejar pagos de pedidos en línea desde la app móvil.

Además se agrega un módulo adicional en el backend que gestionará la acumulación de puntos. Los usuarios recibirán 1 punto por cada \$10,000 gastados, y un administrador podrá ajustar esta regla de conversión desde el backend.

Por otro lado, para el sistema de cotización, el backend recibirá un archivo AutoCAD, que procesará y lo enviará a la API externa, devolviendo los metros cuadrados necesarios. Esta información se usará para calcular los materiales requeridos en la cotización.

Base de datos:

Abarca todos los datos históricos de los clientes, productos, pedidos, pagos, envíos y maneja la integridad entre ellos, es decir, la relación entre productos, clientes y pedidos.

Se incorporarán tablas para registrar puntos acumulados por cada usuario y la configuración de reglas de puntos, permitiendo modificaciones desde el backend.

Tecnologías a implementar

Frontend:

- **HTML, CSS y JavaScript** Para diseñar la interfaz de la página web.
- **React** Para desarrollar una SPA (Single Page Application) que ofrezca una experiencia de usuario fluida y rápida.
- **React Native:** Para desarrollar la app móvil y mantener consistencia entre plataformas (iOS y Android).

Backend:

- **Node.js** para manejar el procesamiento en tiempo real y respuestas rápidas, como el cálculo de cotizaciones y la gestión de pedidos en línea.

se añadirán rutas específicas para:

- Integrar la API de MercadoPago para pagos seguros y en tiempo real.
- Procesar archivos AutoCAD, enviarlos a la API externa y recibir el cálculo de metros cuadrados.

Base de datos:

- **MySQL** para almacenar datos de clientes, productos, stock, pedidos, y transacciones.

Se añade una extensión del esquema para incluir información de puntos de los clientes.

**INGENIERÍA EN SISTEMAS DE
INFORMACIÓN**

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Requerimientos de software

Requerimientos funcionales:

- RF1 – El sistema debe permitir registrar nuevos usuarios.
- RF2 – El sistema debe permitir que los clientes inicien sesión con su usuario y contraseña.
- RF3 – El sistema debe permitir la gestión de perfiles, incluyendo la actualización de datos personales y direcciones de envío.
- RF4 – El sistema debe permitir filtrar búsquedas de productos agrupados por catálogo.
- RF5 – El sistema(web) debe permitir ingresar los metros cuadrados a construir y los tipos de materiales a usar.
- RF6 – El sistema debe calcular la cantidad de material necesario para la construcción basándose en las dimensiones ingresadas y el tipo de material.
- RF7 – El sistema debe aplicar descuentos cuando se cumplan condiciones definidas.
- RF8 – El sistema debe mostrar el importe total de los materiales cotizados, incluyendo los descuentos.
- RF9 – El sistema debe permitir convertir cotizaciones en pedidos cuando el cliente lo confirme.
- RF10 – El sistema debe solicitar dirección de envío y la forma de pago
- RF11 – El sistema debe permitir consultar el estado del pedido de los clientes
- RF12 – El sistema debe gestionar el stock ante los pedidos y envíos.
- RF13 – El sistema(móvil) debe permitir ver la cantidad de puntos acumulados
- RF14 – El sistema(móvil) debe permitir realizar pagos desde la app a través de MercadoPago.
- RF15 – El sistema(móvil) debe acumular puntos por cada compra, con una regla inicial de 1 punto por cada \$10,000 gastados.
- RF16 – El sistema(móvil) debe integrar una API que, a partir de un archivo AutoCAD, calcule automáticamente los metros cuadrados de pared y techo para la cotización.
- RF17 – La regla de conversión de puntos debe ser modificable desde el backend por un administrador.

Requerimientos no funcionales:

- RNF1 – El sistema debe poder manejar grandes cantidades de solicitudes al mismo tiempo sin alterar su desempeño.
- RNF2 – El sistema debe mantener la integridad y seguridad de los datos de cada usuario, así como también la información de pago de cada pedido.
- RNF3 – La interfaz web debe ser intuitiva y fácil de usar para los clientes
- RNF4 – Los tiempos de respuesta frente a solicitudes deben ser rápidos
- RNF5 – El sistema debe ser fácil de mantener y de actualizar sin afectar a su funcionamiento.

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

RNF6 – El sistema debe ser compatible con distintos navegadores y ser accesible desde móviles

RNF7 – La app debe ser multiplataforma y compatible con iOS y Android.

Datos con los que tratar

En un proceso de compra típico, el cliente interesado deberá identificarse en la página web mediante un registro o inicio de sesión en su cuenta. Para esto, se le solicitarán datos de inicio de sesión (usuario y contraseña). Asociado a su cuenta, estarán sus datos de perfil, que incluirán nombre y apellido, correo electrónico, dirección o direcciones de envío, CUIT, teléfono de contacto, entre otros.

Una vez dentro del sistema, el cliente podrá acceder a una lista de catálogos que agrupan los diferentes tipos de productos que ofrece la empresa o bien usar el buscador y aplicar filtros. Al seleccionar un catálogo, se mostrarán los productos asociados, con información detallada como el nombre del producto, imagen del producto, precio unitario, descripción, uso o aplicaciones recomendadas, stock disponible.

El cliente puede ir seleccionando productos y agregándolos a un carrito de compras. En este carrito, se mostrará los mismos datos mencionados anteriormente junto a la cantidad de cada producto y a un lado el sistema de cotización.

Para el sistema de cotización el cliente tendrá la opción de ingresar la cantidad de metros cuadrados a construir, lo cual permitirá al sistema calcular automáticamente la cantidad de materiales necesarios para su construcción. Para esto, el sistema tomará en cuenta las dimensiones y tipo de material seleccionado.

Además, si se cumple con ciertas condiciones, como superar una cantidad específica de productos (por ejemplo, 10.000 ladrillos), el sistema aplicará descuentos automáticos. Estos descuentos se determinan con un porcentaje sobre el valor del producto según la cantidad calculada.

Después de realizar la cotización y revisar el carrito, el cliente podrá confirmar el pedido o cancelarlo, confirmar o editar datos de envío y elegir un método de pago. Una vez que se realiza el pago, el pedido se procesa y se enviará al cliente, usando el propio servicio de envíos de la empresa para hacer la entrega.

En la versión móvil, el proceso es el mismo, con la diferencia de que el proceso de cotización se realiza a partir de un archivo AutoCAD.

Prototipos(web)

Inicio de sesión

The screenshot shows a web browser window with the title 'TP3'. The address bar contains 'https://pagina-ventas.com/login'. The main content area displays a login form titled 'Inicio de sesión'. The form has two input fields: 'Usuario:' with the placeholder text 'usuario', and 'Contraseña:' with the placeholder text 'contraseña'. Below the password field is a blue link labeled 'Registrarse' and a button labeled 'Iniciar sesión'.

Flujo de datos cuando se da click en “Iniciar sesión”

Frontend

Los datos ingresados (Usuario y Contraseña) se envían a través de una petición HTTP (POST) a una API.

La solicitud es en formato JSON de la forma:

```
{  
  "usuario": "Juan",  
  "contrasena": "123"  
}
```

API

La API recibe la solicitud POST y valida los datos comparando el usuario y la contraseña con los almacenados en la base de datos.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

La API envía una consulta SELECT a la base de datos que busca el usuario en la tabla correspondiente, por ejemplo:

```
SELECT usuario, contraseña  
WHERE usuario = 'Juan' and contraseña = '123';
```

Si la consulta no devuelve resultados, significa que el usuario o contraseña no existe en la base de datos. La API responde con un error al frontend.

Si el usuario y contraseña son válidos la API genera una respuesta de éxito al frontend junto a un token de autenticación que el cliente utilizará para futuras solicitudes.

Frontend

Si el inicio de sesión es exitoso, el frontend redirige al usuario a la pagina principal con la lista de los materiales.

Si el inicio de sesión falla, el frontend muestra un mensaje de error indicando que el usuario/contraseña son incorrectos.

Registro de usuario

The screenshot shows a web browser window with the address bar displaying `https://pagina-ventas.com/register`. The page title is "TP3". The main content area displays a registration form titled "Creación de cuenta". The form contains the following fields and controls:

- Usuario:
- Nombre:
- Apellido:
- Email:
- Contraseña: ☐
- Confirmar contraseña: ☐
- Crear cuenta:

Flujo de datos cuando se da click en “Crear cuenta”

Fronted

Los datos ingresados por el usuario en los campos del formulario de registro (Usuario, Nombre, Apellido, Email, Contraseña, Confirmar contraseña) se envían a través de una petición HTTP (POST) a la API.

La solicitud tiene el siguiente formato JSON:

```
{
  "usuario": "usuario",
  "nombre": "Nombre",
  "apellido": "Apellido",
  "email": "email@example.com",
  "contrasena": "password",
  "confirmar_contrasena": "password"
}
```

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

API

La API recibe la solicitud POST con los datos del usuario.

Validación inicial:

La API verifica que todos los campos obligatorios estén completos.

Valida que el email tenga el formato adecuado.

Verifica que la contraseña y su confirmación coincidan.

Consulta a la base de datos:

La API verifica si el nombre de usuario o el email ya están registrados en la base de datos mediante una consulta:

```
SELECT usuario, email
FROM usuarios
WHERE usuario = 'usuario' OR email = 'email@example.com';
```

Si no existen registros previos con ese nombre de usuario o email, la API guarda los datos en la base de datos:

```
INSERT INTO usuarios (usuario, nombre, apellido, email, contraseña)
VALUES ('usuario', 'Nombre', 'Apellido', 'email@example.com',
'contraseña');
```

Base de datos

La base de datos confirma que recibió un nuevo registro y devuelve un mensaje de éxito o error a la API.

API

Si el registro es exitoso, la API devuelve una respuesta de éxito al frontend junto a un token de autenticación.

Si hay algún error (por ejemplo, email o usuario ya registrado), la API responde con un mensaje de error.

Frontend

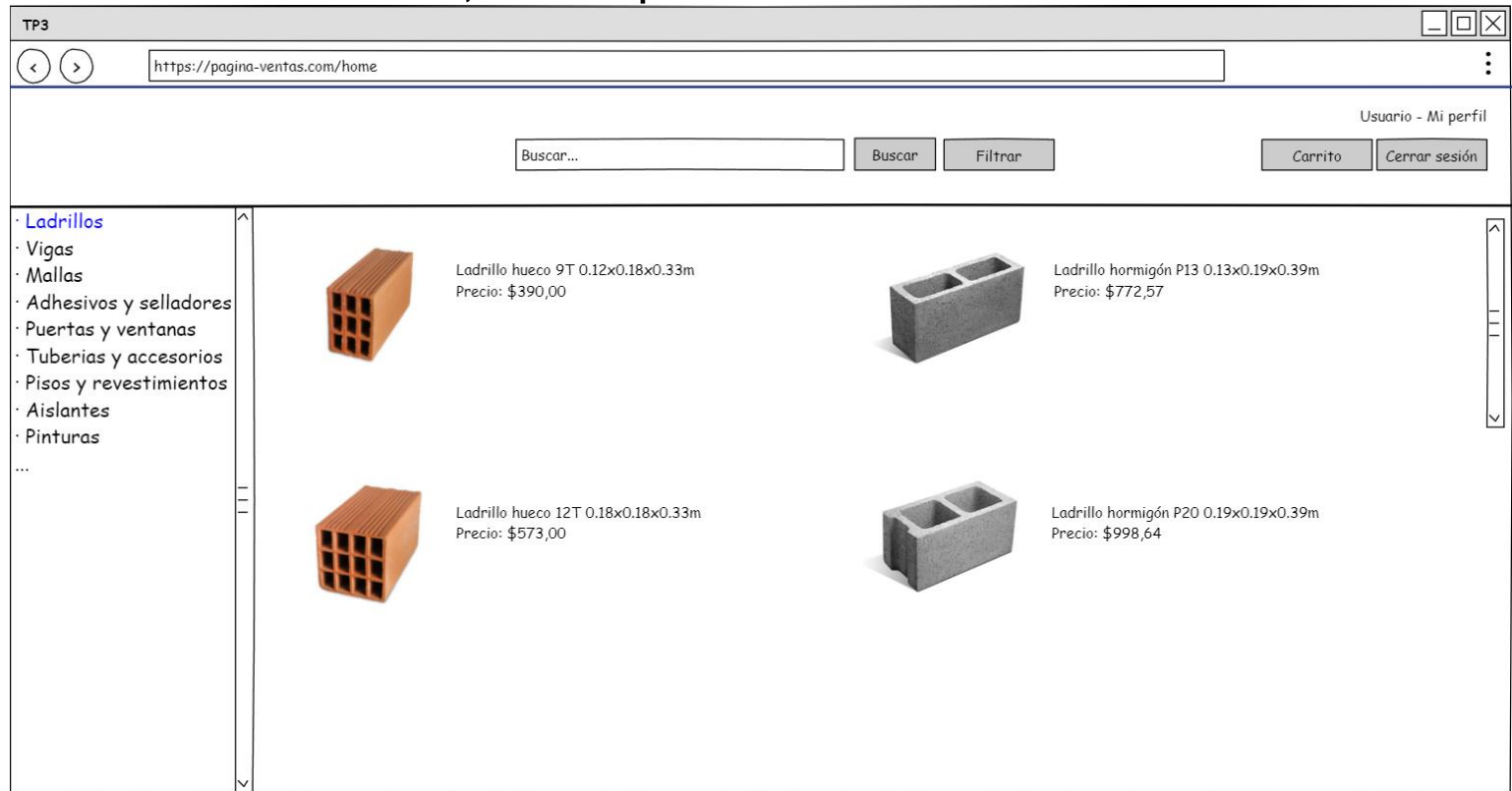
Si el registro es exitoso:

El frontend redirige al usuario a la página principal de la aplicación.

Si el registro falla:

El frontend muestra un mensaje de error indicando el motivo (usuario ya existente, email registrado, error en la validación de contraseñas, etc.).

Pantalla de inicio, listado de productos



Flujo de datos:

1 - Al hacer clic en un título del catálogo:

Frontend

El usuario hace clic en una categoría del catálogo (por ejemplo, "Ladrillos"). Esto envía una solicitud GET a la API con la categoría seleccionada

API

La API recibe el ID de la categoría, consulta la base de datos para obtener todos los productos asociados a esa categoría:

```
SELECT * FROM productos WHERE categoria = 'Ladrillos';
```

La base de datos responde con los productos filtrados y la API devuelve esos datos en formato JSON:

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

```
[
  {
    "id": 1,
    "nombre": "Ladrillo hueco 9T",
    "dimensiones": "0.12x0.18x0.33m",
    "precio": 390.00,
    "imagen": "ladrillo_9T.jpg"
  },
  {
    "id": 2,
    "nombre": "Ladrillo hueco 12T",
    "dimensiones": "0.18x0.18x0.33m",
    "precio": 573.00,
    "imagen": "ladrillo_12T.jpg"
  },
  ...
]
```

Respuesta del Frontend:

Los productos de esa categoría se renderizan en la pantalla principal.

2 - Cuando se escribe en el buscador y se hace clic en "Buscar":

Frontend

El usuario introduce un término de búsqueda (por ejemplo, "Ladrillo hueco") y hace clic en el botón "Buscar". Acá la solicitud GET incluye el término de búsqueda como parámetro

GET /api/products?search=ladrillo

API

La API busca coincidencias de productos en la base de datos basadas en el término ingresado:

```
SELECT * FROM productos WHERE nombre LIKE 'ladrillo' OR descripcion  
LIKE 'ladrillo';
```

La base de datos responde con los productos relevantes y la API devuelve la lista de productos que coinciden.

Respuesta del Frontend:

La interfaz muestra los productos que coinciden con el término de búsqueda.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

3. Al hacer clic en "Filtrar" por nombre o precio:

Frontend

El usuario selecciona un criterio de filtrado (por ejemplo, por precio ascendente) y hace clic en "Filtrar". La solicitud GET se envía con el filtro correspondiente

```
GET /api/products?category=Ladrillos&sort=price_asc
```

API

La API consulta la base de datos y aplica el filtro solicitado:

```
SELECT * FROM productos WHERE categoria = 'Ladrillos' ORDER BY precio  
ASC;
```

La base de datos responde con los productos cumpliendo con el filtrado y la API devuelve una lista de productos ordenados o filtrados por el criterio seleccionado.

Respuesta del Frontend

Los productos filtrados se muestran en la pantalla de acuerdo con el criterio elegido.

4. Al hacer clic en "Carrito":

Frontend

Al hacer clic en "Carrito", Se envía una solicitud GET para obtener los productos en el carrito

```
GET /api/cart
```

API

La API consulta la base de datos para obtener los productos que el usuario ha agregado al carrito, basándose en un identificador:

```
SELECT * FROM carrito WHERE usuario_id = 123;
```

Respuesta del Frontend

Se renderiza una lista de los productos en el carrito, junto con las opciones de cotización.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

5. Al hacer clic en "Cerrar sesión":

Frontend

El usuario cierra sesión, lo que envía una solicitud POST a la API para cerrar la sesión
activa: POST /api/logout

API

La API procesa la solicitud y elimina la sesión del usuario en el servidor

6. Al hacer clic en un producto:

Frontend

El usuario hace clic en un producto, lo que abre los detalles del producto en cuestión.
Se envía una solicitud GET para obtener la información del producto
GET /api/products/1

API

La API recibe el ID del producto, consulta la base de datos para obtener la información completa del producto (nombre, descripción, precio, imágenes, etc.)

```
SELECT * FROM productos WHERE id = 1;
```

La API devuelve:

```
{
  "id": 1,
  "nombre": "Ladrillo hueco 9T",
  "precio": 390.00,
  "descripcion": "Ladrillo hueco cerámico 12x18x33 cm
9 tubos" "Ladrillo de cerramiento",
  "categoria": "Ladrillos",
  "uso": "Especiales para tabiques divisorios y cerramientos
(ambientes interiores y muros de cierre).",
  "cantidad_pallet": 144,
  "imagen_url": "ladrillo_9T.jpg"
}
```

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
Nº3

Trabajo Práctico

Respuesta del Frontend

Se muestra una pestaña de detalles del producto, con la opción de agregarlo al carrito y la cantidad:

TP3

https://pagina-ventas.com/home

Usuario - Mi perfil

Buscar... Buscar Filtrar Carrito Cerrar sesión

- Ladrillos
- Vigas
- Mallas
- Adhesivos y selladores
- Puertas y ventanas
- Tuberías y accesorios
- Pisos y revestimientos
- Aislantes
- Pinturas
- ...

Ladrillo hueco 9T 0.12x0.18x0.33m
Precio: \$390,00

Ladrillo hormigón P13 0.13x0.19x0.39m
Precio: \$772,57

Ladrillo hueco 12T 0.18x0.18x0.33m
Precio: \$573,00

Nombre: Ladrillo Hueco 9T
Precio por unidad: \$390,00

Descripción:
Ladrillo hueco cerámico 12x18x33 cm 9 tubos
Ladrillo de cerramiento

Cantidad por pallet: 144

Uso:
Especiales para tabiques divisorios y cerramientos
(ambientes interiores y muros de cierre).

Cantidad: < 1 > Agregar al carrito

Carrito

TP3


https://pagina-ventas.com/home/Usuario/carrito

Usuario - Mi perfil

Buscar... Buscar Filtrar

Carrito Cerrar sesión

☒




Ladrillo hueco 9T 0.12x0.18x0.33m
Precio: \$390,00

Cantidad Unidades

Eliminar

☐



Viga 4 mts
Precio: \$10.619,00

Cantidad Unidades

Eliminar

☐



Chapa acanalada N°27 1.10 x 3m
Precio: \$29.690,00

Cantidad Unidades

Eliminar

Ancho

Largo

Alto

Calcular

Cantidad Pallet

SUBTOTAL:
\$\$\$

TOTAL: \$\$\$

Aceptar Cancelar

Flujo de datos

1. Seleccionar un producto para cotización

Cuando el usuario abre el carrito, se ve desplegado la lista de los productos que seleccionó.

Si busca cotizar uno de los productos, lo selecciona y el frontend guarda el ID del producto seleccionado para luego enviar una solicitud POST para calcular la cotización cuando el usuario ingrese dimensiones.

2. El usuario ingresa las dimensiones (largo, ancho, alto) y presiona "Calcular".

frontend

Envía una solicitud POST /carrito/cotizacion con los datos del producto seleccionado y las dimensiones ingresadas, por ejemplo:

```
{  
  "productoId": 1,
```

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

```
"dimensiones": {  
  "largo": 5.0,  
  "ancho": 4.0,  
  "alto": 3.0  
}
```

API

La API recibe los datos, consulta la base de datos para obtener las especificaciones del producto y calcula la cantidad necesaria en base a sus dimensiones.

Retorna el resultado de la cotización, incluyendo el descuento si aplica.

```
{  
  "productoId": 1,  
  "nombre": "Ladrillo hueco 9T",  
  "cantidad Necesaria": 50,  
  "Cantidad pallet": 1,  
  "precio": 390.00,  
  "descuento": 10, // 10% de descuento  
  "subtotal": 17550.00 // Precio con descuento aplicado  
}
```

El *frontend* actualiza el subtotal junto con el total y lo muestra al usuario.

3. Se acepta la cotización

El *frontend* envía una solicitud POST con los productos cotizados:

```
{  
  "usuario_id": 12345,  
  "productos": [  
    { "producto_id": 1, "cantidad": 50 },  
    { "producto_id": 2, "cantidad": 1 },  
    { "producto_id": 3, "cantidad": 1 }  
  ],  
  "total": 57859.00  
}
```

API

Registra el pedido en la base de datos mediante una consulta

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

```
INSERT INTO pedidos (id_usuario, total) VALUES (12345, 57959.00);  
INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad,  
subtotal)  
VALUES (54321, 1, 50, 17550);
```

Este ultimo insert se repite tantas veces como productos haya en el carrito.

Respuesta de la API:

```
{  
  "mensaje": "Cotización aceptada",  
  "pedidoId": 54321  
}
```

El *fronted* recibe la respuesta, si todo ha salido bien, se redirige al usuario a la siguiente pantalla donde se muestra el resumen del pedido y se solicita la información de envío

4. El usuario presiona "Cancelar".

Se limpia la cotización actual en el frontend sin enviar datos a la API.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico


Pedido

TP3

https://pagina-ventas.com/home/usuario/pedido/54321


Usuario - Mi perfil

Buscar... Buscar Filtrar Carrito Cerrar sesión



Nombre: Ladrillo Hueco 9T
Precio por unidad: \$390,00
Descripción: Ladrillo hueco cerámico 12x18x33 cm 9 tubos
Ladrillo de cerramiento
Cantidad por pallet: 144
Uso: Especiales para tabiques divisorios y cerramientos (ambientes interiores y muros de cierre).

Cantidad: 50
Subtotal: \$17.550,00



Nombre: Viga 4mts
Precio por unidad: \$10.619,00
Descripción: Viga metálica 4mts
Uso: Son utilizadas para techar en la construcción. Se colocan sobre las paredes y van acompañadas entre viga y viga por ladrillos para techo y malla sima.

Cantidad: 1
Subtotal: \$10.619,00

Datos de envío

Nombre:

Apellido:

Email:

Teléfono:

CUIT:

C. P.: Ciudad:

Provincia:

Calle:

Nro.:

TOTAL: \$57.859,00

Volver Confirmar pedido

Flujo de datos

1 – Cuando se confirma el pedido.

Antes de proceder con el pago, el sistema valida que todos los campos en "Datos de envío" (nombre, apellido, email, teléfono, CUIT, código postal, ciudad, provincia) estén completos.

Si falta algún campo o no es válido, el sistema muestra un mensaje de error indicando que debe completarse correctamente.

Si los datos están completos, el frontend enviará un JSON que contiene:

- ID del pedido (ya generado).
- Datos de envío (nombre, apellido, email, teléfono, CUIT, código postal, ciudad, provincia, calle, Nro.).

Este JSON será enviado a la API para actualizar el estado del pedido en la base de datos, indicando que el pedido está "confirmado" y listo para proceder al pago. Luego el frontend redirige a la pestaña de pago para finalizar la compra.

2 – Cuando se da click en “Volver”.


Redirige a la pantalla anterior (carrito) sin modificar el pedido en la base de datos.


Pago

TP3

[←](#) [→](#)

Usuario - Mi perfil

☒ Tarjeta de crédito 

☐ Tarjeta de débito 

☐ Depósito / transferencia

Nombre propietario

Número CVV

Fecha vencimiento

Flujo de datos para pago por tarjeta

Frontend

Se hace un metodo HTTP POST a la API, los datos que se envían incluyen la información del pedido y los detalles de pago:

```
{  
  "nombre": "propietario.nombre",  
  "numero": 5042 5013 5012 5007,  
  "CVV": 245,  
  "fecha_vencimiento": 07/24  
}
```

API

La API se comunica con un servicio de pagos externo (como visa, MercadoPago, etc.) para validar y procesar el pago. Estos datos se envían en formato JSON:

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

```
{  
  "moneda": "ARS",  
  "precio": 57859.00,  
  "descripcion": "Pago de pedido",  
  "nombre": "nombre1",  
  "numero_tarjeta": 5042 5013 5012 5007,  
  "código_seguridad": 245,  
  "fecha_caducidad": 07/24  
}
```

Una vez que el pago ha sido procesado correctamente por el servicio de pago, se carga en la base de datos el estado actualizado del pedido con el pago aceptado

```
INSERT INTO pedidos (54321, descripción_pago)  
VALUES (1, 'Pago confirmado');
```

La API luego responde al frontend, confirmando que el pedido ha sido registrado y que el pago ha sido procesado exitosamente.


Frontend


El frontend recibe esta respuesta y finaliza el proceso de compra, retornando a la pantalla principal que muestra la lista de productos.

TP3

[<](#) [>](#)

Usuario - Mi perfil

☐ Tarjeta de crédito 

☐ Tarjeta de débito 

☒ Depósito / transferencia

Banco: [Nombre del banco]
Titular de la cuenta: [Nombre del titular]
Número de cuenta: [Número de cuenta]
CBU: [Número de CBU]
Alias de cuenta: [Alias]
CUIT: [Número de CUIT]
Monto a pagar: \$57.859,00

Instrucciones adicionales:

- Asegurate de incluir el **número de tu pedido [54321]** en la referencia o motivo de la transferencia para facilitar la identificación del pago.
- El pedido será procesado una vez confirmado el pago. Envíanos el comprobante de la transferencia a **empresa@gmail.com**.

Importante:

- Tienes hasta 72 horas para completar el pago. Pasado este tiempo, el pedido será cancelado automáticamente.

Atención al cliente:
Si tienes alguna duda o necesitas asistencia, puedes contactarnos al **3564-123456** o escribirnos a **empresa@gmail.com**

Flujo de datos para pago por depósito o transferencia bancaria

En este caso si se decide pagar por depósito / transferencia, al confirmar compra se carga al pedido en la base de datos el estado de “pago pendiente”.
Luego, manualmente un administrador de BD deberá modificar el estado del pedido a “Pago confirmado” una vez reciba el pago por parte del usuario que realiza la compra.

El fronted redirige a la pagina de inicio una vez seleccionado “confirmar compra”.

Flujo de datos para “volver”

Redirige a la pantalla anterior (carrito) sin modificar el pedido en la base de datos.

Prototipos(Móvil)

Para la versión móvil del sistema el flujo de datos se desarrolla de manera similar en varios puntos, principalmente porque la arquitectura se mantiene como en la versión web del sistema. Solo cambia que el frontend en lugar de ejecutarse en un navegador se ejecuta en el dispositivo móvil, siguiendo el mismo principio de enviar solicitudes, generalmente en formato JSON, a través de peticiones HTTP.

Inicio de sesión

The image shows a mobile application interface for a 'Página Ventas' (Sales Page). It features a central white card with the title 'Iniciar Sesión' (Login). The card contains two input fields: 'Usuario' (Username) and 'Contraseña' (Password), each with a placeholder text. Below the password field is a blue button labeled 'Iniciar sesión'. At the bottom of the card, there is a blue link labeled 'Registrarse' (Register). The entire interface is set against a light gray background within a black border representing the mobile device frame.

Flujo de datos cuando se da click en “Iniciar sesión”

Frontend

Verifica que los campos Usuario y Contraseña no estén vacíos.
Si algún campo está vacío, muestra un mensaje de error al usuario.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Si los campos están completos, envía una solicitud de inicio de sesión.

Solicitud al Backend

El frontend envía una solicitud POST a la API de inicio de sesión del backend con los datos ingresados.

Datos enviados

usuario: nombre de usuario ingresado.
contraseña: contraseña ingresada

Autenticación en el Backend

El backend recibe la solicitud y busca en la base de datos el usuario con el nombre ingresado.

Si el usuario existe, compara la contraseña ingresada con la almacenada.

Si la autenticación es exitosa, el backend genera un token de autenticación y lo envía al frontend junto con el perfil del usuario.

Respuesta del Backend

Éxito: El backend envía el token de autenticación y los datos básicos del usuario

Error: Si el nombre de usuario o la contraseña son incorrectos, envía un mensaje de error al frontend.

Recepción en el Frontend

Éxito: El frontend guarda el token en el almacenamiento seguro del dispositivo y redirige al usuario a la pantalla principal de la app.

Error: Muestra el mensaje de error en la pantalla de inicio de sesión.

Registro de usuario

The image shows a mobile application interface for user registration. The screen has a light gray background with a white registration form in the center. The form is titled 'Registrarse' and contains the following fields: 'Usuario', 'Correo electrónico', 'Nombre', 'Apellido', 'Contraseña', and 'Confirmar contraseña'. Each field has a placeholder text. At the bottom of the form is a blue button labeled 'Registrarse'. The entire form is set against a background that says 'Página Ventas' at the top.

Flujo de Datos al Presionar "Registrarse"

Frontend:

Valida que todos los campos estén completos y que los valores sean válidos

Si algún campo está incompleto o es incorrecto, el sistema muestra un mensaje de error al usuario.

Si todos los datos son válidos, se envía la solicitud de registro al backend.

El frontend envía una solicitud POST al endpoint de registro de la API del backend.

Datos enviados en formato JSON:

```
{
  "usuario": "usuarioIngresado",
  "correoElectronico": "correo@ejemplo.com",
  "nombre": "NombreIngresado",
  "apellido": "ApellidoIngresado",
  "contraseña": "contraseñaIngresada"
}
```

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Backend:

El backend recibe la solicitud y verifica que el nombre de usuario y el correo electrónico no estén en uso.

Si el nombre de usuario o el correo ya existen en la base de datos, el backend responde con un mensaje de error indicando el conflicto.

Si los datos son válidos, el backend crea un nuevo registro de usuario en la base de datos y guarda la información correspondiente (nombre, apellido, correo, usuario, y contraseña).

Respuesta del Backend:

Éxito: Si el registro se completa con éxito, el backend envía una respuesta de confirmación, que puede incluir un token de autenticación para iniciar sesión automáticamente después del registro.

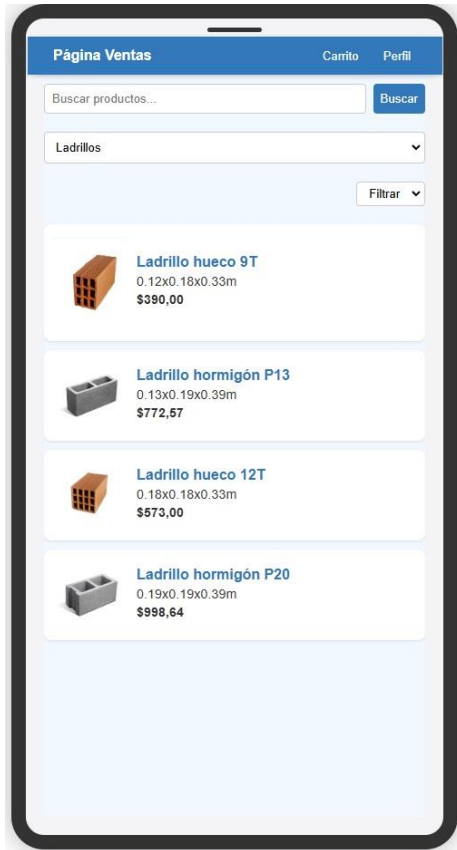
Error: Si ocurre un error (como usuario o correo ya existentes), el backend responde con un mensaje de error, que el frontend mostrará en la pantalla de registro.

Recepción en el Frontend:

Éxito: Si el registro fue exitoso y el backend envió un token de autenticación, el frontend almacena el token en el almacenamiento seguro del dispositivo y redirige al usuario a la pantalla principal de la app.

Error: Si hay un mensaje de error (como "nombre de usuario ya en uso"), el frontend muestra el mensaje para que el usuario pueda corregirlo.

Menú



Flujo de datos:

1. Al Seleccionar una Categoría (ej. "Ladrillos")

El usuario selecciona una categoría desde el menú desplegable de categorías.

Solicitud al Backend:

El frontend envía una solicitud GET a la API del backend con el parámetro de la categoría seleccionada (por ejemplo, categoria=ladrillos).

Ejemplo de solicitud: `GET /api/productos?categoria=ladrillos`

Respuesta del Backend:

El backend recibe el parámetro y filtra los productos según la categoría solicitada.

El backend responde con una lista de productos que pertenecen a la categoría "Ladrillos".

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Actualización del Frontend:

El frontend recibe los datos y muestra los productos filtrados por la categoría seleccionada en la pantalla.

2. Al Escribir en el Buscador y Presionar "Buscar"

El usuario ingresa un término de búsqueda y hace clic en el botón "Buscar".

Solicitud al Backend:

El frontend envía una solicitud GET al backend con el término de búsqueda como parámetro (por ejemplo, `busqueda=ladrillo`).

Ejemplo de solicitud: `GET /api/productos?busqueda=ladrillo`

Respuesta del Backend:

El backend procesa la búsqueda y devuelve una lista de productos que coinciden con el término ingresado.

Actualización del Frontend:

El frontend recibe y muestra los productos que coinciden con la búsqueda realizada.

3. Al Presionar "Filtrar" (por ejemplo, por Precio Ascendente)

El usuario selecciona un criterio de filtro (por ejemplo, ordenar por precio ascendente) y hace clic en el botón "Filtrar".

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Solicitud al Backend:

El frontend envía una solicitud GET al backend con el criterio de filtro seleccionado (por ejemplo, orden=precio_asc).

Ejemplo de solicitud:

```
GET /api/productos?categoria=ladrillos&orden=precio_asc
```

Respuesta del Backend:

El backend organiza los productos de acuerdo con el criterio especificado (precio ascendente en este caso) y envía la lista ordenada al frontend.

Actualización del Frontend:

El frontend recibe los datos y muestra los productos en el orden filtrado.

4. Al Hacer Clic en "Carrito"

El usuario hace clic en el botón "Carrito" en el menú superior.

Redirección a la Pantalla del Carrito

frontend

redirige al usuario a la pantalla Carrito, donde podrá ver los productos que ha agregado para comprar.

Carga de Datos del Carrito

Al acceder a la pantalla del carrito, el frontend puede enviar una solicitud GET para obtener los productos del carrito actual almacenados en el backend o en el almacenamiento local.

5. Al Hacer Clic en un Producto

El usuario selecciona un producto haciendo clic en su nombre o imagen.

Solicitud de Detalles del Producto al Backend:

El frontend envía una solicitud GET al backend con el ID del producto seleccionado para obtener sus detalles completos.

Ejemplo de solicitud: `GET /api/productos/{idProducto}`

Respuesta del Backend

El backend envía los detalles completos del producto, incluyendo su descripción, precio, dimensiones, y cualquier otra información relevante.

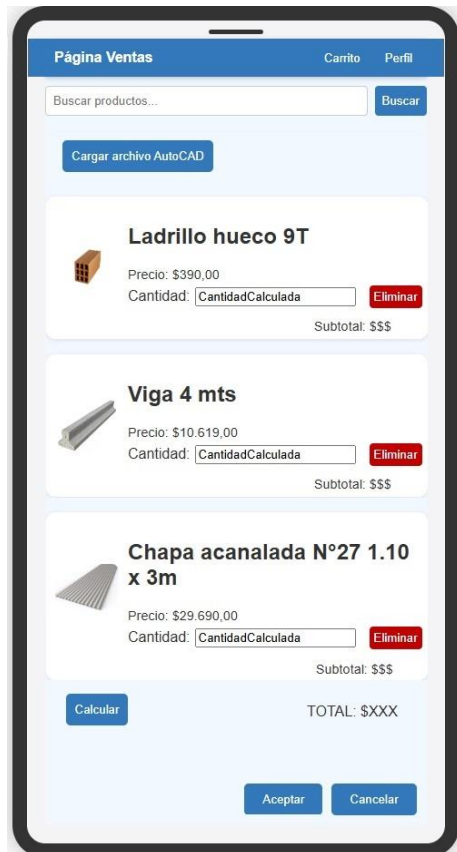
Redirección a la Pantalla de Detalles del Producto:

Frontend

redirige al usuario a una nueva pantalla que muestra los detalles completos del producto seleccionado:



Carrito



Flujo de datos:

1. Cuando se aprieta en "Cargar archivo AutoCAD"

El usuario selecciona un archivo AutoCAD en su dispositivo y hace clic en "Cargar archivo AutoCAD".

Frontend

Se envía una solicitud POST al backend con el archivo AutoCAD adjunto en el cuerpo de la solicitud.

Backend

El backend recibe el archivo y lo envía a la API externa de AutoCAD para su procesamiento.

**INGENIERÍA EN SISTEMAS DE
INFORMACIÓN**

Diseño de Sistemas de Información
N°3

Trabajo Práctico

La API de AutoCAD interpreta el archivo, calcula los metros cuadrados de pared y techo, y devuelve estos valores en formato JSON.

El backend recibe los datos de metros cuadrados y procede a calcular la cantidad de materiales necesarios (ej., ladrillos, vigas, chapas, etc.) basándose en estos metros cuadrados.

Respuesta al Frontend

El backend envía una respuesta en formato JSON al frontend con los productos calculados y la cantidad requerida de cada uno.

El frontend actualiza la pantalla para mostrar la lista de productos junto con la cantidad calculada de cada uno.

2. Cuando se aprieta en "Calcular" posterior a la carga del archivo

Frontend

El usuario hace clic en "Calcular". El frontend envía una solicitud al backend que contiene los productos seleccionados y sus cantidades.

Backend

Recibe la solicitud con los productos y sus cantidades, calcula el subtotal de cada producto, y luego suma todos los subtotales para obtener el total de la cotización.

Respuesta del Backend

Devuelve el total calculado al frontend en un formato JSON.

Frontend

Muestra el total de la cotización en la pantalla en el campo "TOTAL: \$XXX".

3. Cuando se aprieta en "Aceptar" la cotización

Frontend

Al hacer clic en "Aceptar", el frontend envía una solicitud POST al backend para confirmar la cotización y registrar la compra.

La solicitud incluye los detalles de los productos en el carrito (ID, cantidad, precio) y el total calculado.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Backend

El backend procesa la solicitud y registra la información de la cotización en la base de datos como un pedido o presupuesto.
Devuelve una respuesta de confirmación al frontend.

Respuesta al Frontend

El frontend muestra una notificación de confirmación de que la cotización ha sido aceptada y, se redirige al usuario a la pantalla del pedido.

4. Cuando se aprieta en "Cancelar"

Frontend

Al hacer clic en "Cancelar", el frontend borra la información de la cotización actual del carrito y restablece la pantalla de productos.

Pedido

Página Ventas Carrito Perfil

Buscar productos... **Buscar**

Ladrillo Hueco 9T
Precio: \$390,00
Subtotal: \$17,550.00

Viga 4mts
Precio: \$10,619.00
Subtotal: \$10,619.00

Datos de envío

Nombre
Apellido
Email
Teléfono
CUIT
C.P.
Ciudad
Provincia
Calle
Número

TOTAL: \$57,859.00

Confirmar pedido Volver

Flujo de datos:

1. Cuando se aprieta en "Confirmar pedido"

Frontend

El usuario llena todos los campos de "Datos de envío" (nombre, apellido, email, teléfono, CUIT, código postal, ciudad, provincia, calle, y número).

Al hacer clic en "Confirmar pedido", el frontend valida que todos los campos estén correctamente llenos.

Luego, envía una solicitud POST al backend con la información del pedido, que incluye:

- Los productos en el carrito (ID, cantidad, precio unitario, subtotal de cada producto).
- El total del pedido.
- Los datos de envío ingresados por el usuario.

INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

Backend

El backend recibe los datos y guarda la información del pedido en la base de datos, incluyendo los detalles del cliente, los productos, y el total de la compra.

Respuesta al Frontend

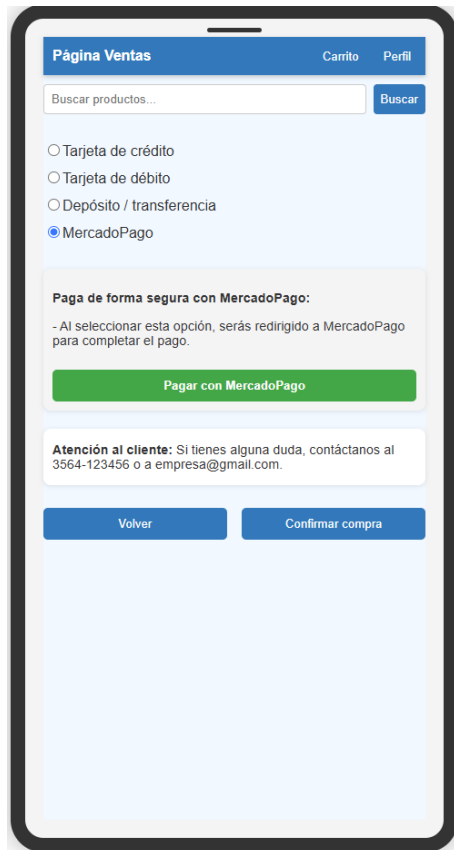
El frontend muestra un mensaje de confirmación al usuario indicando que el pedido ha sido realizado exitosamente y lo redirige a la pantalla de pago.

2. Cuando se aprieta en "Volver"

Frontend

Al hacer clic en "Volver" el frontend ignora los datos ingresados en los campos de envío y regresa a la pantalla anterior (el carrito).

Pago



Flujo de datos:

1. Cuando se aprieta "Pagar con MercadoPago"

El usuario selecciona la opción "MercadoPago" como método de pago.

Frontend

Al hacer clic en "Pagar con MercadoPago", el frontend envía una solicitud al backend para iniciar el proceso de pago con MercadoPago.

Backend

El backend recibe la solicitud y genera una preferencia de pago en MercadoPago con los detalles de la compra.

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Diseño de Sistemas de Información
N°3

Trabajo Práctico

El backend utiliza la API de MercadoPago para crear un enlace de pago y luego devuelve ese enlace al frontend.

Respuesta al Frontend

El frontend redirige al usuario a la página de pago de MercadoPago utilizando el enlace proporcionado.

El usuario completa el pago en la plataforma de MercadoPago. Al finalizar el pago, MercadoPago redirige al usuario a la aplicación con una respuesta de éxito o falla.

2. Cuando se aprieta "Confirmar Compra"

Frontend

El usuario regresa a la pantalla de confirmación de compra después de completar el pago en MercadoPago.

Al hacer clic en "Confirmar compra", el frontend envía una solicitud POST al backend con los datos del pedido y la confirmación del pago.

Backend

El backend verifica el estado del pago con MercadoPago para confirmar que el pago fue exitoso.

Si el pago es exitoso, el backend guarda el pedido en la base de datos.

Se aplica el sistema de puntos: el backend calcula los puntos a otorgar al usuario (1 punto por cada \$10,000 gastados). Estos puntos se agregan al saldo de puntos del usuario en su cuenta.

El backend responde al frontend confirmando que la compra ha sido completada y que los puntos han sido agregados.

Respuesta al Frontend

El frontend muestra un mensaje de confirmación indicando que la compra fue exitosa y el total de puntos ganados en esta transacción

Posteriormente, el usuario es redirigido a la pantalla de inicio(menú)

3. Cuando se aprieta "Volver"

Frontend

Al hacer clic en "Volver", el frontend ignora los datos de pago y regresa a la pantalla anterior (carrito o selección de método de pago).