

Trabajo práctico 3

DISEÑO DE SISTEMAS

UTN Facultad Regional San Francisco.

Ing. En sistemas de información

Gianotti Ornella

ÍNDICE

ENUNCIADO:	3
PROPUESTA DE SOLUCIÓN:	4
REQUERIMIENTOS:	4
PROTOTIPOS DE INTERFAZ DE USUARIO:	5
REGISTRO DE USUARIOS:	5
CREACIÓN DE CUENTA:	6
BÚSQUEDA DE PRODUCTOS:	7
REALIZAR COTIZACIÓN:	9
REALIZAR PEDIDO:	13
REALIZAR PAGO:	14
PAGO CON TARJETA:	15
PAGO CON TRANSFERENCIA:	16
PAGO CON MERCADO PAGO:	18
MIS PUNTOS:	19
INICIO DE SESIÓN PARA ADMINISTRADORES:	21
GESTIÓN DE PUNTOS:	21



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, ejemplo:

	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
	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.

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.

PROPUESTA DE SOLUCIÓN:

Dado que la empresa se dedica a la fabricación y venta de materiales para la construcción, con una amplia gama de productos, es esencial brindar a los clientes una forma simple y eficiente de cotizar los materiales y realizar pedidos.

La solución propuesta consiste en el desarrollo de una **aplicación móvil multiplataforma (iOS y Android)** que permita a los clientes generar cotizaciones y gestionar pedidos mediante una interfaz intuitiva y fácil de usar. Esta app ofrecerá cotizaciones personalizadas, donde los clientes podrán ingresar manualmente las dimensiones de la superficie a construir, seleccionar el tipo de material deseado o cargar un archivo Autocad, del cual la app calculará automáticamente los metros cuadrados de pared y techo necesarios mediante una API integrada. A partir de estos datos, el sistema calculará la cantidad de productos requeridos, aplicará descuentos (si los hubiere) y mostrará el precio total de la cotización.

Una vez generada la cotización, los clientes podrán convertirla en un pedido. Para completar el proceso, se les solicitará que ingresen los datos de envío y seleccionen una forma de pago, que podrá ser tarjeta de débito/crédito, transferencia bancaria o Mercado Pago. Además, por cada compra realizada, los clientes acumularán puntos en función de una regla ajustable por el administrador.

Toda la información relacionada con los clientes, cotizaciones, pedidos y puntos acumulados se almacenará en una base de datos, lo que permitirá a la app gestionar eficientemente el flujo de trabajo y el seguimiento de los pedidos.

REQUERIMIENTOS:

REQUERIMIENTOS FUNCIONALES:

RF1: El sistema debe permitir el registro e inicio de sesión para los clientes.

RF2: El sistema debe permitir a los clientes visualizar los productos disponibles.

RF3: El sistema debe permitir a los clientes realizar cotizaciones en base a un producto seleccionado y las dimensiones de construcción.

RF4: El sistema debe integrar una API que permita calcular los metros cuadrados de pared y techo a partir de un archivo Autocad cargado por el cliente.

RF5: El sistema debe mostrar el resultado de la cotización calculando la cantidad necesaria de materiales en función de las dimensiones ingresadas.

RF6: El sistema debe permitir a los clientes realizar un pedido basado en la cotización generada, requiriendo que ingresen la información de envío.

RF7: El sistema debe ofrecer a los clientes la opción de realizar el pago de sus pedidos mediante distintas plataformas como mercado pago, tarjeta de crédito/débito o transferencia.

RF8: El sistema debe permitir a los clientes acumular puntos de fidelización por cada compra realizada, basados en una regla configurable por el administrador.

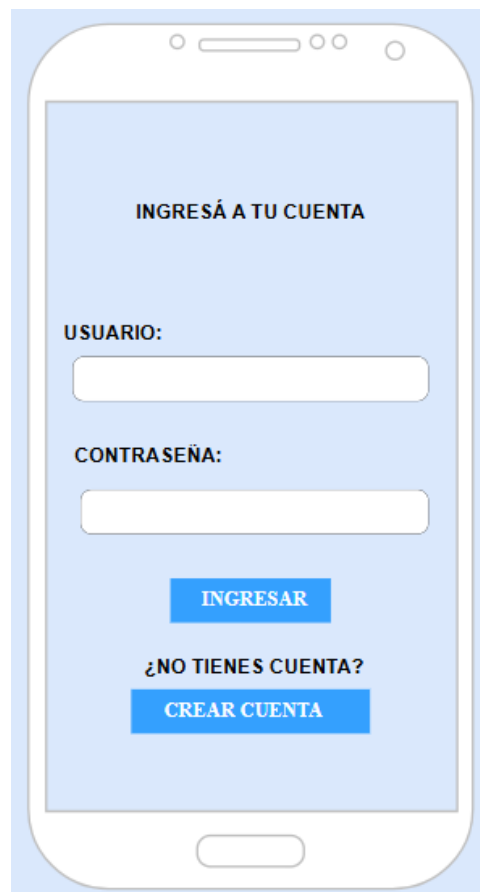
RF9: El sistema debe permitir el inicio de sesión para los administradores.

RF10: El sistema debe permitir a un administrador modificar la regla de acumulación de puntos, que definirá la cantidad de puntos (n) obtenidos por cada monto específico en pesos (x) gastado y el porcentaje de descuento en función de la cantidad de puntos acumulados.

PROTOTIPOS DE INTERFAZ DE USUARIO:

REGISTRO DE USUARIOS:

Los usuarios podrán iniciar sesión para ser identificados y en caso de no tener una cuenta podrán crearla.



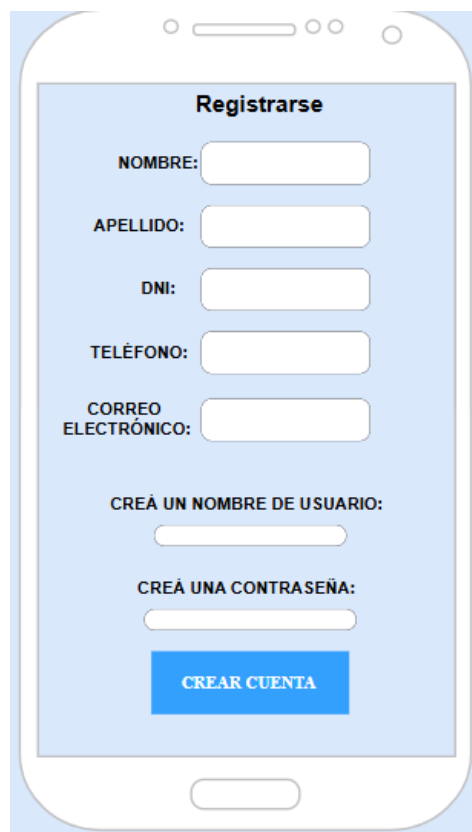
Cuando el usuario ingresa sus datos, es decir nombre de usuario y contraseña y presiona el botón "Ingresar", la app toma los valores ingresados en estos campos y se envían al backend:

```
{  
  "usuario": "nombreUsuario",  
  "contraseña": "12345"  
}
```

El Backend recibe estos datos y realiza una consulta a la base de datos en la tabla usuarios para verificar si el usuario existe y si la contraseña ingresada coincide.

CREACIÓN DE CUENTA:

En el caso de hacer clic en "Crear Cuenta" los usuarios serán dirigidos a la siguiente interfaz.

A mobile application interface for user registration. The screen is titled "Registrarse" in bold black text. Below the title, there are five input fields with labels: "NOMBRE:", "APELLIDO:", "DNI:", "TELÉFONO:", and "CORREO ELECTRÓNICO:". Below these fields, there are two more input fields with labels: "CREÁ UN NOMBRE DE USUARIO:" and "CREÁ UNA CONTRASEÑA:". At the bottom of the form is a blue button with the text "CREAR CUENTA" in white capital letters. The entire form is set against a light blue background within a white rounded rectangle that represents the phone screen.

Cuando el cliente completa el formulario y presiona "Crear Cuenta", los datos ingresados son capturados por la app y los envía al Backend.

```
[{
  "nombre": "Juan",
  "apellido": "Pérez",
  "usuario": "juan123",
  "contraseña": "miContraseñaSegura",
  "correo": "juan@correo.com",
  "telefono": "1234567890",
  "dni": "12345678"
}]
```

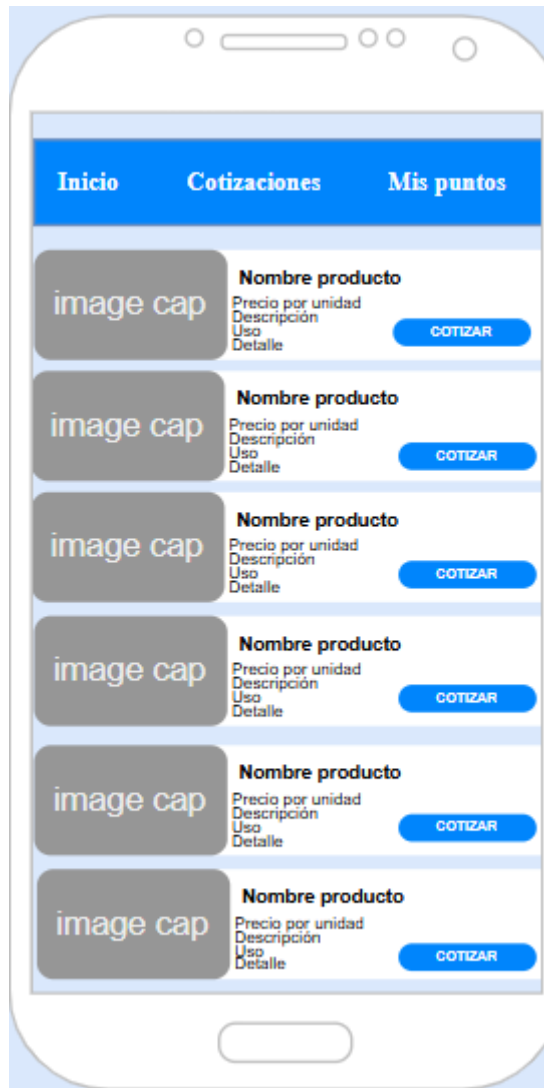
Una vez que el Backend recibe estos datos, establece una conexión con la tabla clientes en la base de datos insertándole los datos proporcionados por el cliente, donde el campo rol se establece automáticamente como cliente por defecto.

Tabla: usuarios

Campo	Tipo de dato
IdUsuario	INT (PK)
nombre	Varchar(50)
apellido	Varchar(50)
usuario	Varchar(30)
contraseña	Varchar(50)
correo	Varchar(50)
telefono	Numeric
dni	Numeric
fechaCreacion	datetime
rol (cliente o administrador)	Varchar(50)

BÚSQUEDA DE PRODUCTOS:

Los clientes pueden ver un listado de productos al ingresar a la pantalla de inicio.



Datos necesarios:

- Productos:
 - ID del Producto
 - Nombre del Producto
 - Descripción del Producto
 - Imagen del Producto
 - Precio por Unidad
 - Uso
 - Detalle

Cuando el cliente carga la pantalla de productos, la app envía una solicitud GET al backend para obtener los productos disponibles. El Backend accede a la base de datos y realiza una consulta para obtener los datos de la tabla de productos.

Tabla productos:

Campo	Tipo de dato
idProducto	Int (PK)
nombre	Varchar(50)
descripcion	Varchar(50)
precioUnidad	Decimal
uso	Varchar(100)
detalle	Varchar(50)

Una vez que el Backend obtiene los productos de la base de datos, empaqueta esta información y la envía a la app.

```
[{
  "IdProducto": 1,
  "Nombre": "Viga 4 mts",
  "Descripcion": "Ladrillo hueco cerámico 12x18x33 cm, 9 tubos, para tabiques y cerramientos.",
  "precioUnidad": 10619.00,
  "uso": "Utilizada para techos, colocada sobre paredes junto con ladrillos y malla sima.",
  "detalle": "Producto de alta resistencia para la construcción."
}]
```

La app recibe el JSON, lo interpreta y presenta cada producto.

Cuando el cliente presiona “Cotizar” en un producto específico la app redirige al cliente a la pantalla de cotización, pasando el idProducto como referencia para la siguiente solicitud.

REALIZAR COTIZACIÓN:

Una vez que los clientes seleccionaron un producto a cotizar, tendrán 2 opciones “ingresar las dimensiones de la construcción” o “subir plano de Autocad”

En el caso de que los clientes seleccionen la opción de ingresar las dimensiones, deberán ingresar largo, alto, ancho a construir. Al hacer clic en "Realizar Cotización", los datos ingresados se envían al Backend.

```
[{
  "idProducto": 1,
  "largo": 10,
  "ancho": 5,
  "alto": 4
}]
```

El backend recibe los datos, consulta la base de datos para obtener el precio unitario del producto y realiza los cálculos necesarios para determinar la cantidad de materiales y el precio total. En algunos casos se le podrá agregar un descuento que se lo calcula en función de las reglas creadas (por ejemplo, un 5% a partir de 10,000 unidades) o en función de los puntos acumulados.

En el caso de que el cliente haya seleccionado la opción de subir un plano de Autocad, en lugar de ingresar manualmente las dimensiones, el cliente cargará un archivo .dwg o .dxf (formato estándar de Autocad) desde su dispositivo.

Al hacer clic en “Realizar Cotización”, la app envía el archivo al backend.

```
{
  "idProducto": 2,
  "archivoPlano": "<archivo_autocad.dwg>"
}
```

El backend recibe el archivo de Autocad y hace una solicitud a la API externa implementada en él, que analiza el plano para calcular los metros cuadrados de pared y techo.

La API devuelve un JSON con la información procesada, incluyendo los metros cuadrados necesarios.

```
{
  "metrosCuadradosPared": 80,
  "metrosCuadradosTecho": 50
}
```

Luego, el backend utiliza el idProducto para consultar la base de datos y obtener el precio unitario del producto. Con esta información y los metros cuadrados proporcionados por la API, calcula la cantidad de materiales necesarios y el precio total de la cotización. Si corresponde, aplica descuentos según las reglas establecidas para generar el monto final.

Una vez que el backend tiene los datos procesados, empaqueta la información (metros cuadrados a construir, cantidad de materiales, precio unitario, precio total, descuentos, etc.) en un JSON y lo envía de vuelta a la app.

```
[{
  "nombreProducto": "..",
  "cantidadNecesaria": 100,
  "precioUnitario": 10619.00,
  "metrosCuadrados": 50,
  "descuentosAplicados": 5309.50,
  "precioTotal": 1061900.00
}]
```

COTIZACIONES - REALIZAR COTIZACIÓN

Producto seleccionado:

Metros cuadrados a construir:

Precio unitario:

Cantidad de productos:

Descuentos aplicados:

Precio total:

REALIZAR PEDIDO

CANCELAR


Si el cliente confirma la cotización haciendo clic en “realizar pedido”, el backend guarda los datos en la tabla cotizaciones en la base de datos.

Tabla Cotizaciones

Campo	Tipo de dato
IdCotizacion	Int (PK)
IdUsuario	Int (FK)
IdProducto	Int (FK)
cantProducto	Int
descuentos	Numeric
metrosCuadrados	Int
precioTotal	Numeric
fechaCotizacion	datetime

REALIZAR PEDIDO:

Cuando el cliente hace clic en "Realizar Pedido", es redirigido a una pantalla en la que debe ingresar los datos de envío.



REALIZAR PEDIDO

Ingrese los datos de envío

PROVINCIA:

CIUDAD:

CÓDIGO POSTAL:

CALLE:

NÚMERO:

CONFIRMAR PEDIDO

CANCELAR

Al presionar "Confirmar Pedido", la app envía estos datos al backend.

```
[{
  "idCotizacion": 1,
  "provincia": "Cordoba",
  "ciudad": "San Francisco",
  "codigoPostal": "2400",
  "calle": "ernesto vidal",
  "numero": 1990
}]
```

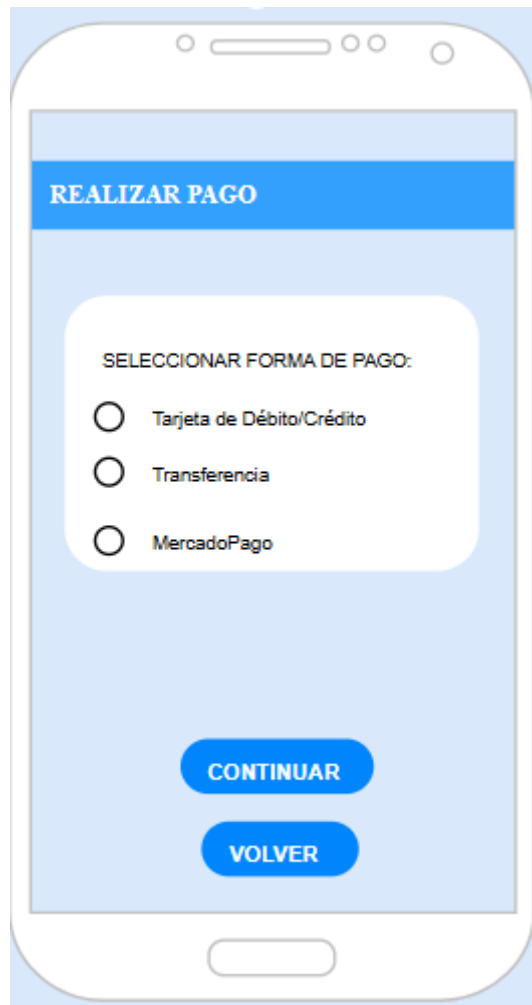
El backend recibe la solicitud y establece una conexión con la base de datos, almacenando los datos en la tabla Pedido con la siguiente estructura:

Tabla Pedido:

Campo	Tipo de dato
idPedido	Int (PK)
idCotizacion	Int (FK)
provincia	Varchar(50)
ciudad	Varchar(50)
codigoPostal	int
calle	Varchar(50)
numero	int

Una vez almacenado el pedido se captura el idPedido generado y se redirige al cliente a la siguiente pantalla

REALIZAR PAGO:



REALIZAR PAGO

SELECCIONAR FORMA DE PAGO:

☐ Tarjeta de Débito/Crédito

☐ Transferencia

☐ MercadoPago

CONTINUAR

VOLVER

El cliente en esta pantalla puede seleccionar entre las opciones de pago disponibles: tarjeta de débito/crédito, MercadoPago o transferencia bancaria.

Tras elegir una opción, la app redirige a la pantalla correspondiente según el método seleccionado.

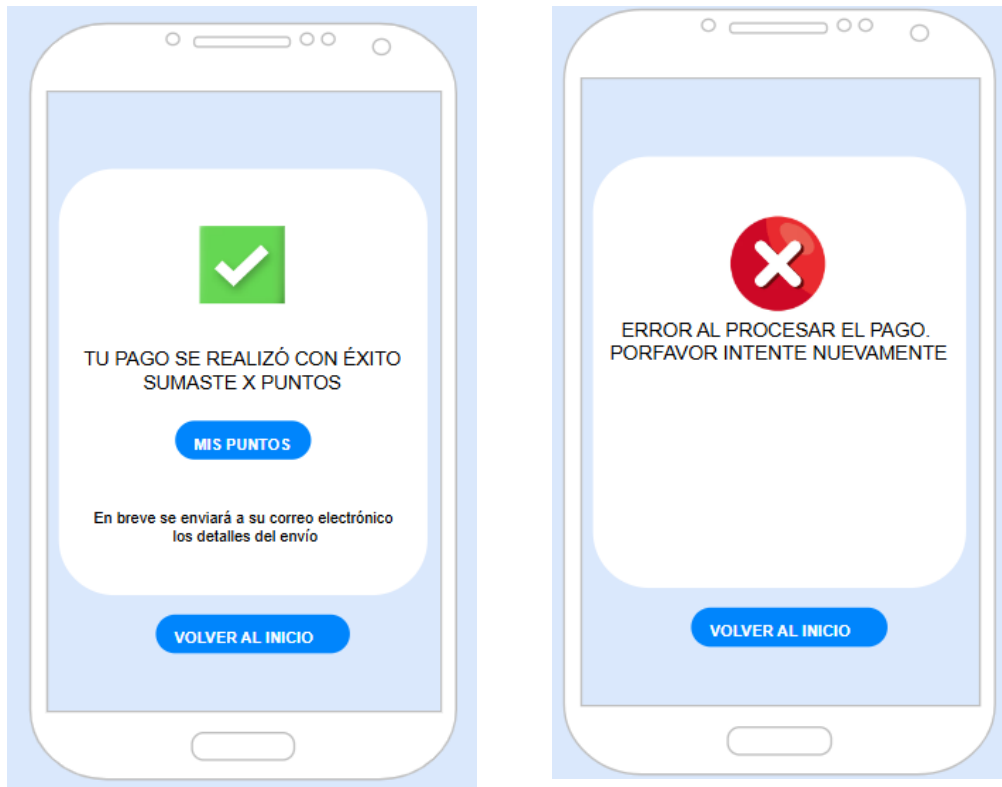
PAGO CON TARJETA:



The screenshot shows a mobile application interface for making a payment with a card. The screen is titled "Realizar pago con tarjeta:" in a blue header. Below the header, there are several input fields: "NÚMERO DE TARJETA" with the value "1234 1234 1234 1234", "VENCIMIENTO" with the value "MM/AA", "CÓDIGO DE SEGURIDAD" with the value "123", "NOMBRE DEL TITULAR COMO APARECE EN LA TARJETA" with the value "María López", and "DOCUMENTO DEL TITULAR" with a dropdown menu set to "DNI" and the value "99999999". At the bottom of the form, there are two buttons: "REALIZAR PAGO" and "CANCELAR".

En esta pantalla el cliente introduce los datos solicitados, una vez completados, la app envía estos datos de manera segura al backend, este lo procesa y envía una respuesta indicando el estado del pago (éxito o error).

Si el pago se realiza con éxito, la app muestra un mensaje de confirmación y si ocurre un error, la app muestra el mensaje: "Error al procesar el pago. Por favor, intente nuevamente."



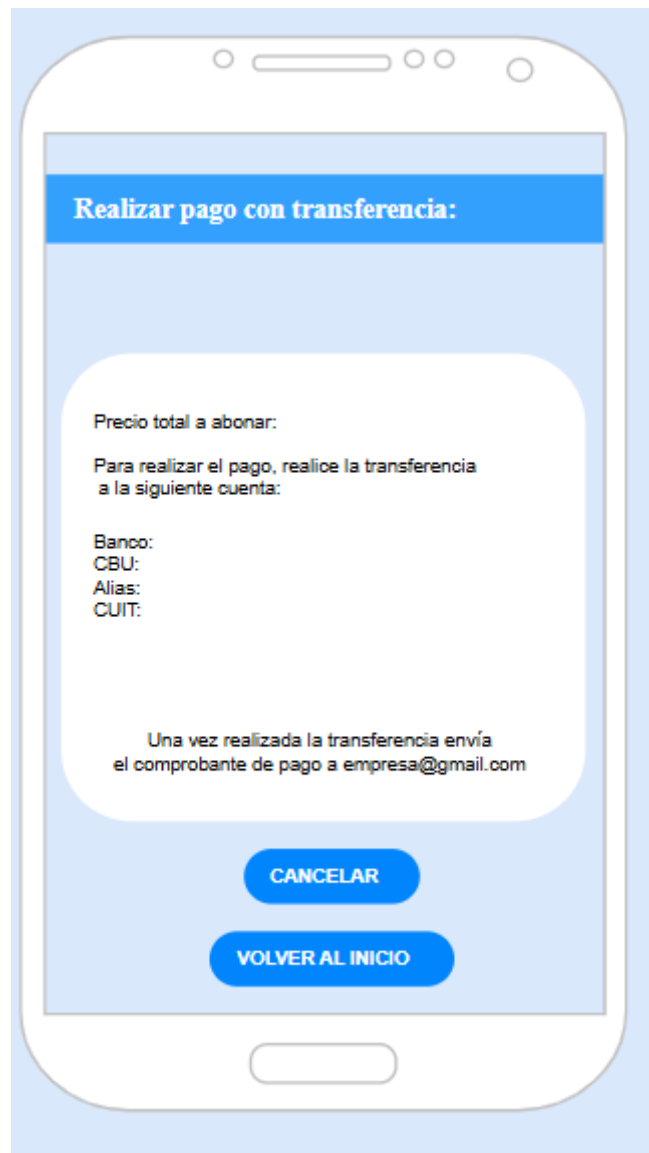
Luego de que el cliente completa la transacción, se insertará un nuevo registro en la tabla Pagos de la base de datos.

Tabla pagos

Campo	Tipo de dato
idPago	Int (PK)
idPedido	Int (FK)
monto	Decimal
metodoPago	Varchar(50)
fechaPago	datetime
referenciaPago	Varchar(50)

PAGO CON TRANSFERENCIA:

El cliente al seleccionar la opción de transferencia en la pantalla de selección de forma de pago, se lo redirige a esta pantalla donde aparecerán los datos para realizarla.

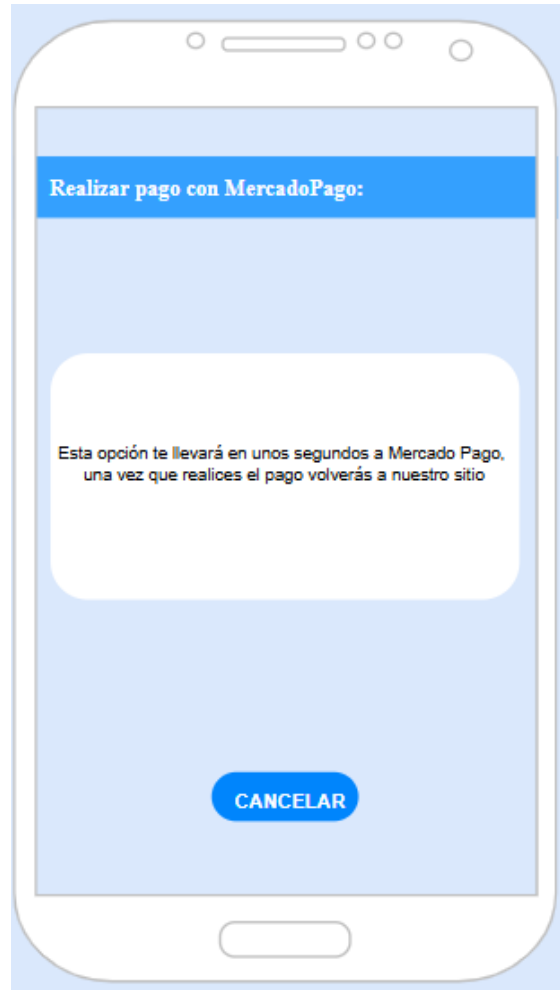


Luego de que el cliente completa la transacción, se insertará un nuevo registro en la tabla Pagos de la base de datos.

Campo	Tipo de dato
idPago	Int (PK)
idPedido	Int (FK)
monto	Decimal
metodoPago	Varchar(50)
fechaPago	datetime
referenciaPago	Varchar(50)

PAGO CON MERCADO PAGO:

En el caso de seleccionar MercadoPago en la pantalla de seleccionar método de pago se lo redirige a esta pantalla.



Cuando el cliente selecciona esta opción, la app inicia una redirección automática a la página de pago de MercadoPago mediante una integración con su API. Una vez completado el pago, esta plataforma envía una notificación automática (webhook) al backend de la app, confirmando el estado de la transacción, esta notificación incluye detalles como el idPedido, monto pagado, y el estado de la transacción (éxito o error).

Al recibir la notificación, el backend actualiza la información del pago en la tabla Pagos de la base de datos.

Tabla pagos

Campo	Tipo de dato
idPago	Int (PK)
idPedido	Int (FK)

monto	Decimal
metodoPago	Varchar(50)
fechaPago	datetime
referenciaPago	Varchar(50)

Si el pago se realiza con éxito, la app muestra un mensaje de confirmación y si ocurre un error, la app muestra el mensaje: “Error al procesar el pago. Por favor, intente nuevamente.”



MIS PUNTOS:

Cuando el cliente finaliza un pago exitoso, el backend calcula los puntos a acumular según la regla definida: por ejemplo, “cada \$50,000 gastados, ganas 10 puntos”.

El backend registra estos puntos en la base de datos en una tabla de historial de puntos.

Tabla HistorialPuntos:

Campo	Tipo de dato
-------	--------------

idHistorial	Int (PK)
idCliente	Int (FK)
fecha	Datetime
montoGastado	decimal
puntosGanados	Int

Cuando el cliente presiona el botón "Mis Puntos", la app envía una solicitud GET al backend para obtener:

- El total de puntos acumulados.
- La regla actual de acumulación de puntos.
- El historial de puntos ganados en transacciones anteriores.

El backend consulta la tabla HistorialPuntos para sumar todos los puntos ganados por el cliente y obtener el total actual.

También consulta la tabla de configuración de puntos (reglaPuntos) para obtener la regla actual (por ejemplo, "Cada \$50,000 gastados ganas 10 puntos; con 20 puntos accedes a un 15% de descuento en tu próxima compra").

El backend empaqueta los datos en un JSON que contiene el total de puntos acumulados, la regla de puntos, y el historial de transacciones y la app muestra estos datos en la pantalla de "Mis Puntos".



INICIO DE SESIÓN PARA ADMINISTRADORES:

Los administradores al ingresar a la app web podrán iniciar sesión de la misma forma que los clientes, pero con un usuario y contraseña proporcionados por la empresa que en la base de datos su rol es "administrador".

El backend verifica las credenciales y rol de "administrador" en la base de datos. Si son correctas la app redirige al administrador a la pantalla de gestión de puntos.

GESTIÓN DE PUNTOS:

Al ingresar a la pantalla, la app envía una solicitud GET al backend para obtener la configuración actual de la regla de puntos.

El backend consulta la base de datos y devuelve los valores actuales.

The screenshot shows a mobile app interface for 'Gestión de puntos'. At the top is a blue header with the title 'Gestión de puntos'. Below the header, the current rule is displayed: 'Regla de puntos actual: 10 puntos por cada 50.000 pesos gastados. Cada 20 puntos se accede a un descuento del 15%.' Below this, there is a section for a 'Nueva regla de puntos' with four input fields: 'Monto para puntos:', 'Puntos por monto:', 'Puntos para descuento:', and 'Porcentaje de descuento:'. At the bottom of the form area is a blue button labeled 'GUARDAR CAMBIOS'. Below the entire form area is a blue button labeled 'CANCELAR'.

El administrador podrá modificar las reglas actuales de puntos completando los campos "monto para puntos", "puntos por monto", "puntos para descuento", "porcentaje de descuento".

Al presionar el botón “guardar cambios”, la app envía los datos al backend en una solicitud POST.

El backend valida los datos y actualiza la tabla de regla de puntos en la base de datos.

Tabla reglaPuntos

campo	Tipo de dato
idRegla	Int
montoParaPuntos	Int
puntosPorMonto	Int
puntosParaDescuento	Int
porcentajeDescuento	decimal