

Taller 1 - Levantamiento de requerimientos

Presentado por:

Pablo Andres Nino Barreto - pninob@unal.edu.co

Jeisson Duvan Bareno Ruiz - jbareno@unal.edu.co

Victor Camilo Canon Castellanos - vcanonc@unal.edu.co

Jacobo Alzate Corredor - jalzateco@unal.edu.co

Profesor:

Oscar Eduardo Alvarez Rodriguez - oalvarezr@unal.edu.co



Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

2024

Índice

| | |
|---|-----------|
| Objetivo del documento..... | 3 |
| Alcance del proyecto..... | 4 |
| Funcionalidades principales del sistema..... | 4 |
| Características principales para el administrador..... | 4 |
| Características principales para el cliente..... | 4 |
| Requerimientos funcionales..... | 5 |
| Requerimientos no funcionales..... | 5 |
| Requerimientos de interfaz gráfica..... | 6 |
| Análisis de requerimientos..... | 8 |
| Priorización..... | 8 |
| Estimación de tiempo y complejidad..... | 9 |
| Casos de uso..... | 13 |
| 1. Añadir producto al carrito de compras (Cliente)..... | 14 |
| Historia de Usuario 1..... | 16 |
| Historia de Usuario 2..... | 19 |
| Historia de Usuario 3..... | 23 |
| Historia de Usuario 4..... | 27 |

Software de gestión de tienda on-line de manualidades y artículos de decoración

Objetivo del documento

Este documento tiene como objetivo el levantamiento de requerimientos de software de la entrevista generada por el **segundo grupo**, la cuál se encuentra en el siguiente [link](#).

Contexto

El cliente tiene un pequeño negocio de ventas en línea, el cuál se enfoca en manualidades y en la reventa de artículos para la decoración del hogar. Adicionalmente, el cliente planea ofrecer asesorías personalizadas en diseño de interiores en el futuro. El modelo de negocio puede describirse en los siguientes apartados:

- ❖ **Catálogo:** Se muestran todos los productos a través de una página de Instagram o por WhatsApp de manera individual.
- ❖ **Proceso de venta:** Se realiza manualmente a través de WhatsApp, desde el envío del inventario disponible al cliente con fotografías, el pago y el acuerdo de entrega.
- ❖ **Gestión de inventario:** Se utiliza una hoja de cálculo para el inventario de los productos.
- ❖ **Gestión de pagos:** Transferencias manuales entre el cliente y el comprador.

Se han identificado los siguientes problemas o necesidades del cliente:

- ❖ La actualización manual del estado de los pedidos tiende a generar confusión y errores en el proceso de ventas.
- ❖ Frecuentemente, los productos que no están disponibles no se reflejan correctamente en el inventario.
- ❖ El negocio carece de un control eficiente sobre la información de envío de los clientes.
- ❖ La gestión manual de ventas y pagos a través de transferencias individuales y la comunicación por WhatsApp genera un proceso desorganizado, aumentando la probabilidad de errores y retrasos en la confirmación de las transacciones.
- ❖ La exhibición de productos es ineficiente y desorganizada, ya que se realiza principalmente a través de canales limitados como WhatsApp e Instagram, lo que dificulta una presentación profesional y estructurada para los clientes.

Alcance del proyecto

Desarrollar e implementar un sistema integral de gestión en línea para la venta de productos, actualización de inventario y exhibición profesional que permita al cliente automatizar y optimizar los procesos de ventas, pagos, y gestión de pedidos, mejorando la eficiencia operativa y la experiencia del cliente.

El proyecto priorizará la automatización de la gestión de pedidos e inventario, como requerimientos más urgentes, seguido de funcionalidades complementarias como el calendario para asesorías.

Funcionalidades principales del sistema

Características principales para el administrador

1. **Gestión de clientes y pedidos:** Visualización, creación y actualización de información de los clientes, incluyendo la información de sus pedidos y su estado.
2. **Gestión de inventario:** Registrar y monitorear productos para la venta.
3. **Gestión de ventas y pagos:** Consulta, modificación y registro de las ventas consolidadas.
4. **Programación de asesorías:** Gestión completa del calendario para citas de asesoría personalizadas.
5. **Notificaciones sobre el estado del inventario:** Envío de recordatorios de bajo stock de productos.

Características principales para el cliente

1. **Catálogo en línea:** Visualización de los productos disponibles.
2. **Registro de usuario:** Creación de una cuenta para realizar compras.
3. **Gestión de pedidos y compras:** Consulta del estado de los pedidos realizados, y pasarela de pagos.
4. **Solicitud de agendamiento de citas:** Generación de solicitudes para programar asesorías personalizadas.
5. **Notificaciones sobre pedidos:** Avisos de cambios en el estado de los pedidos realizados.

Requerimientos funcionales

RF-01. Permitir que clientes potenciales se registren creando una cuenta con datos personales y de contacto.

RF-02. El administrador podrá ingresar información de nuevos clientes en el sistema.

RF-03. Gestión de los inventarios y actualización en tiempo real del mismo.

RF-04. Los usuarios podrán gestionar los pedidos realizados según sus permisos en el sistema, con notificaciones automatizadas sobre cambios en el estado de los mismos.

RF-05. El administrador podrá consultar, registrar y modificar las ventas de productos.

RF-06. El sistema contará con un sistema de pagos en línea para el cliente.

RF-07. Generación de facturación electrónica para cada venta concretada.

RF-08. Integración con pasarelas de pago locales para facilitar las transacciones.

RF-09. El administrador podrá organizar el calendario con las asesorías solicitadas por los clientes.

RF-10. Los clientes podrán pedir asesorías personalizadas.

RF-11. Catálogo en línea que permita la visualización de los productos disponibles.

RF-12. El sistema notificará a través de email y WhatsApp advertencias de bajo stock para el administrador.

RF-13. El sistema generará periódicamente un resumen de las finanzas del cliente.

RF-14. El sistema debe permitir a los clientes agregar productos a un carrito de compras virtual, modificar las cantidades, eliminar productos y proceder a la generación de un pedido desde el carrito.

Requerimientos no funcionales

RNF-01. Diseño responsive: El sistema tendrá soporte para pantallas de móviles y de escritorio.

RNF-02. Interfaz simple: La interfaz debe ser simple, con un menú claro y opciones fácilmente accesibles para los usuarios.

RNF-03. Seguridad: Los datos almacenados estarán protegidos y solo accesibles para el administrador del sistema. Esto incluye datos de clientes, pedidos e inventario.

RNF-04. Disponibilidad: El sistema debe ser accesible en todo momento a través de un navegador web, sin necesidad de instalación de software adicional.

RNF-05. Escalabilidad: El diseño debe permitir la adición de nuevas funcionalidades sin necesidad de un rediseño completo.

Requerimientos de interfaz gráfica

RIG-01. Paleta de colores: Utilizar tonos cálidos como beige, terracota y verde oliva para reflejar la personalidad del negocio.

RIG-02. Distribución visual: El diseño debe ser limpio y sencillo, destacando imágenes de productos con poco texto en las áreas principales.

RIG-03. Menú principal: Un menú fijo en la parte superior con opciones claras como "Inicio", "Productos", "Pedidos" y "Contacto."

RIG-04. Tipografía: Utilizar fuentes amigables y legibles, como Montserrat o Poppins.

RIG-05. Botones destacados: Los botones principales, como "Comprar ahora," deben resaltar visualmente y ser fáciles de identificar.

Análisis de requerimientos

Priorización

Se han listado las funcionalidades del sistema de la siguiente manera:

- | | |
|----------------------------------|-------------------------------------|
| 1. Registro de clientes. | 11. Pasarela de pagos. |
| 2. Gestión de clientes. | 12. Gestión de ventas. |
| 3. Catálogo en línea. | 13. Resumen de finanzas. |
| 4. Gestión de inventario. | 14. Programación de asesorías. |
| 5. Notificaciones de inventario. | 15. Solicitud de asesorías. |
| 6. Gestión de pedidos. | 16. Carrito de compras. |
| 7. Consulta de pedidos. | 17. Diseño responsive. |
| 8. Notificaciones de pedidos. | 18. Interfaz intuitiva y accesible. |
| 9. Pagos en línea. | 19. Seguridad. |
| 10. Facturación electrónica. | 20. Disponibilidad. |
| | 21. Escalabilidad inicial. |

Para los cuales se ha definido la siguiente priorización:

| Must Have <i>Esenciales para la viabilidad del producto</i> | Should Have <i>Importantes, pero con menor prioridad</i> | Could Have <i>Deseables, pero no esenciales</i> | Won't Have <i>Se pueden eliminar</i> |
|---|--|---|--|
| Gestión de clientes. | Registro de clientes. | Catálogo en línea. | Notificaciones de inventario. |
| Gestión de inventario. | Consulta de pedidos. | Notificaciones de pedidos. | Resumen de finanzas. |
| Gestión de pedidos. | Pagos en línea. | Facturación electrónica. | Solicitud de asesorías. |
| Gestión de ventas. | Disponibilidad. | Programación de asesorías. | |
| Seguridad. | Escalabilidad inicial. | Carrito de compras. | |
| | | Interfaz intuitiva y accesible. | |
| | | Diseño responsive | |

Estimación de tiempo y complejidad

Contando con un equipo de dos desarrolladores, se ha decidido las siguientes tecnologías:

- **Back-end:** Django REST Framework.
- **Front-end:** React.
- **Base de datos:** PostgreSQL.
- **Servidor para el despliegue:** Heroku.

Dado que el cliente requiere un sistema funcional entregable con unas funcionalidades mínimas en el menor tiempo posible, establecemos que el MVP (Mínimo Producto Viable) debe tener:

1. Gestión de pedidos y su actualización automática.
2. Gestión de inventario con actualización en tiempo real.
3. Gestión de clientes.
4. Gestión de ventas.

Utilizando el método Planning Poker con la escala (1, 2, 3, 5, 8, 13, 21). Se ha realizado la siguiente estimación para lograr el MVP:

| Funcionalidad | Story points | Justificación |
|---|---------------------|--|
| Gestión de pedidos y su actualización automática. | 8 | Lógica compleja de estados y dependencias. |
| Gestión de inventario con actualización en tiempo real. | 8 | Notificaciones y actualización en tiempo real. |
| Gestión de clientes. | 5 | CRUD básico con validaciones sencillas. |
| Gestión de ventas | 5 | Consolidación de información de inventarios y pedidos. |

Total de Story points para el MVP: 26.

Ahora para el resto de funcionalidades que serán adicionadas en consecuentes sprints a la entrega del MVP su estimación es la siguiente:

| Funcionalidad | Story points | Justificación |
|---------------------------------|--------------|--|
| Carrito de compras. | 8 | Verificaciones de stock en tiempo real, guardado de inventario |
| Catálogo en línea. | 5 | Creación de una interfaz para visualizar productos. Incluye filtros y búsqueda. |
| Notificaciones de inventario. | 3 | Consulta de inventario y comunicación con el dispositivo de preferencia para la notificación |
| Consulta de pedidos. | 3 | Acceso a los pedidos en la base de datos con una interfaz sencilla de entender |
| Notificaciones de pedidos. | 5 | Acceso a los pedidos, acceso a las notificaciones del dispositivo de preferencias y comunicación entre estos |
| Pagos en línea. | 8 | Comunicación con distintas plataformas para métodos de pago, conexión entre estas plataformas y las cuentas a las que llegarán e encriptación de datos |
| Facturación electrónica. | 5 | Recibimiento del pago en línea y plantilla en la cuál se rellenará, además de comunicación con el correo u otro medio de preferencia |
| Pasarela de pagos. | 8 | Apis de integración, encriptación de datos para brindar seguridad y comunicación entre servidores |
| Resumen de finanzas. | 5 | Historial de transacciones y balance usando modelos matemáticos y estadísticos |
| Programación de asesorías. | 5 | Visualización de formularios, y creación de una especie de calendario que el admin pueda ver para programar la asesoría |
| Solicitud de asesorías. | 3 | Envío de formulario a la base de datos |
| Diseño responsive. | 5 | Optimización, velocidad de carga, adaptación a varios dispositivos |
| Interfaz intuitiva y accesible. | 8 | Gestión de diseño excelente, tener en cuenta la experiencia de usuario y la accesibilidad |
| Seguridad. | 8 | Encriptamiento, verificación, comunicación con otras plataformas de seguridad |

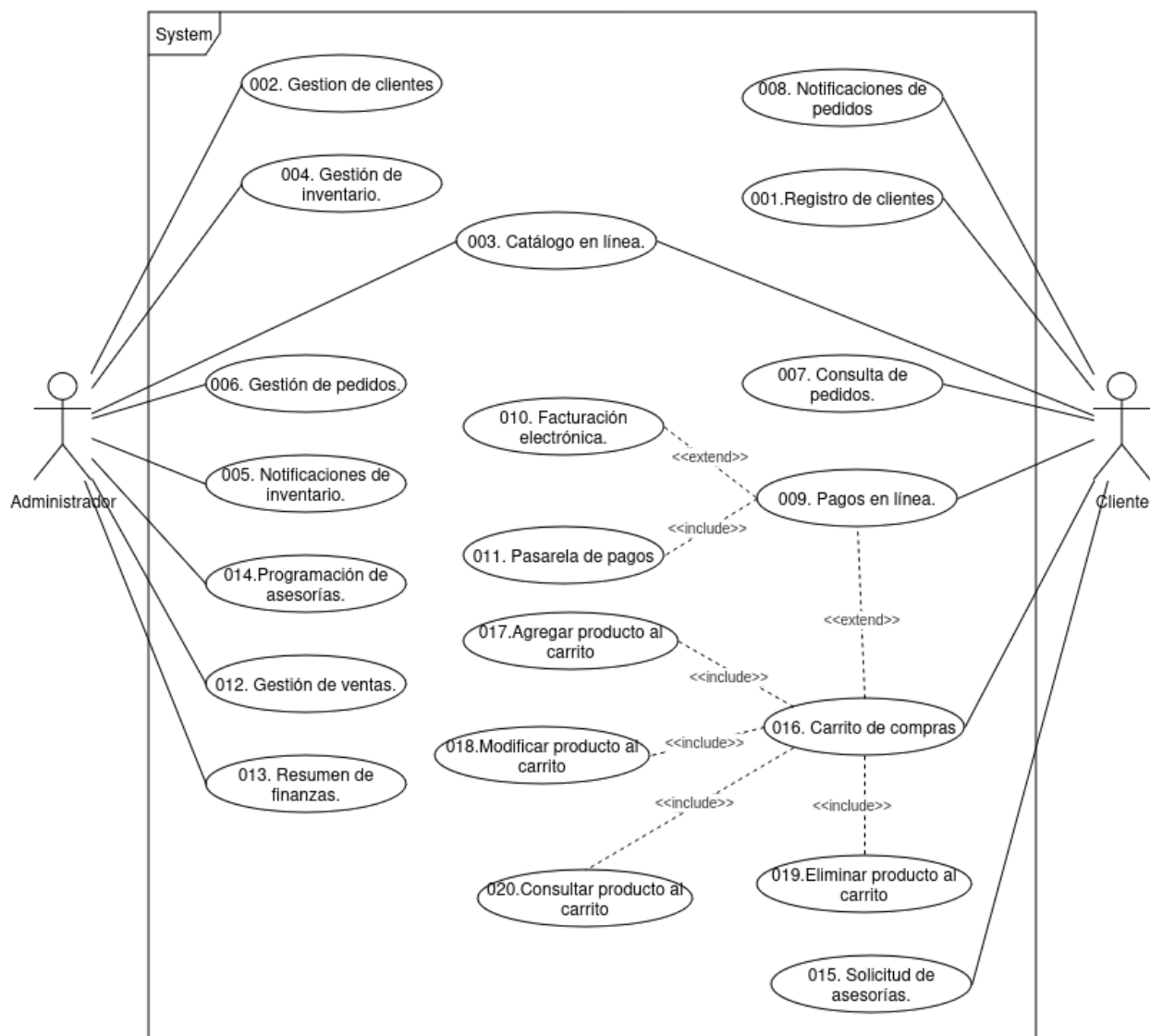
| | | |
|------------------------|---|---|
| Disponibilidad. | 3 | La página no debe ser muy pesada, tener unos buenos servidores y servicios de seguridad para prevenir ataques |
| Escalabilidad inicial. | 5 | Código eficiente y optimizado que no sea pesado |

Total de Story points del resto de funcionalidades: 87.

Total de Story points del proyecto: 113.

Casos de uso

El siguiente es el diagrama de caso de uso del sistema:



1. Añadir producto al carrito de compras (Cliente)

| | | |
|----------------------|---|---------------------------------|
| Caso de Uso | Añadir producto al carrito. | Identificador: CU-017 |
| Actores | Cliente. | |
| Tipo | Primario. | |
| Referencias | | |
| Precondición | El cliente debe estar logueado en el sistema. | |
| Postcondición | El sistema guarda un producto en el carrito de compras del cliente. | |
| Descripción | El cliente agrega productos, modifica las cantidades, elimina productos para la generación de un pedido desde el carrito. | |

a. Curso normal

| Nro. | Ejecutor | Paso o Actividad |
|------|----------|--|
| 1 | Cliente | Hace click en la sección “Productos”. |
| 2 | Sistema | El sistema despliega la información de todos los productos disponibles. |
| 3 | Cliente | Selecciona alguno de los productos mostrados. |
| 4 | Sistema | El sistema muestra una descripción completa del producto. Como también presenta un selector de cantidad y un botón llamado “Agregar al carrito”. |
| 5 | Cliente | Presiona el botón “Agregar al carrito”. |
| 6 | Sistema | Genera un mensaje de confirmación de adición del producto al carrito. |
| 7 | Cliente | El cliente confirma la adición del producto al carrito. |
| 8 | Sistema | Añade el producto al carrito. {Retorna al paso 4} |

b. Curso alternativo

| Nro. Paso | Condición | Descripción de acciones alternas |
|------------|-----------------------------|--|
| 2, 4, 6, 8 | Error interno del servidor. | <ol style="list-style-type: none"> 1. Un error interno ocurre en el sistema. 2. Se muestra el mensaje "Error del servidor. Intente nuevamente más tarde." {Retorna al paso 1} |

Historia de Usuario 1

Descripción conceptual

| | |
|---|--|
| Módulo | <i>Módulo de Gestión de Usuarios</i> |
| Descripción de la(s) funcionalidad(es) requerida(s): | <i>El usuario podrá Iniciar sesión a la página</i> |

Descripción técnica:

| URL | Método | Código html |
|--|--|---|
| localhost:8080/login | POST | 200 (Éxito) 401 (Credenciales inválidas) |
| Caso de uso técnico Cuando el usuario envía sus credenciales correctas (correo y contraseña), el sistema debe retornar un código 200 junto con un token de autenticación en la respuesta. Si las credenciales son incorrectas, se debe devolver un código 401 y un mensaje de error. | | |
| Datos de entrada <pre>{ "email": "usuario@ejemplo.com", "password": "123456" }</pre> | Datos de salida (Exitosa) <pre>{ "status": "success", "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...", "user": { "id": 1, "nombre": "Jacobito Alzate", "email": "usuario@ejemplo.com" } }</pre> | |
| | Datos de salida (Con error) <pre>{ "status": "error", "message": "Credenciales inválidas" }</pre> | |

Frontend

- El cliente podrá colocar su email registrado o su nombre de usuario para poder Iniciar sesión, además de poner la contraseña
- La página será oscura pero no negra, un color bastante oscuro con tonos un poco verdes y azules

Interacción esperada:

El usuario verá varios botones, uno para iniciar sesión cuando haya puesto el usuario o email y la contraseña, otro por si olvidó la contraseña que lo llevará a otro menú para recuperar esta contraseña, otro botón para iniciar sesión con una passkey y otro para registrarse por si el usuario no lo está aún

Mockups/Prototipos:

Sign in

Username or email adress

Password

Sign in

Or

Sign in with a passkey

Create an account

imagen hecha con: <https://wireframe.cc>

Flujo visual y eventos:

- Al presionar Sign in y no tener bien la contraseña saldrá un cuadro de texto diciendo “Usuario o contraseña no válidos”, cuando sean válidos aparecerá una pantalla de carga diciendo “bienvenido” y habrá un ruleta de carga
- Al presionar Sign in with a passkey aparecerá una interfaz en la cuál se podrá colocar un pin de seguridad de 10 dígitos que se le habrá dado con anterioridad
- Al presionar create an account, se abrirá una pantalla donde el nuevo usuario podrá registrarse con un correo electrónico y demás datos personales para poder acceder a la página de ese momento en adelante

Historia de Usuario 2

Descripción conceptual

| | |
|---|--|
| Módulo | <i>Módulo de Gestión de Usuarios</i> |
| Descripción de la(s) funcionalidad(es) requerida(s): | <i>El cliente podrá registrarse como un nuevo usuario en la página</i> |

Descripción técnica:

| URL | Método | Código html |
|--|---|--|
| localhost:8080/register | POST | 201 (Usuario creado exitosamente) 400 (Datos inválidos) |
| <p align="center">Caso de uso técnico</p> <p>Cuando un usuario desea registrarse, debe enviar un conjunto de datos (nombre, correo electrónico y contraseña). Si los datos son válidos, el sistema debe crear un nuevo usuario en la base de datos y retorna un código 201 con los detalles del usuario recién creado. Si los datos son inválidos o faltan campos, el servidor devolverá un código 400 y un mensaje de error.</p> | | |
| <p align="center">Datos de entrada</p> <pre>{ "nombre": "Juan Pérez", "username": JuaPe "email": "usuario@ejemplo.com", "password": "123456" "contact number":1234567890 "adress": }</pre> | <p align="center">Datos de salida (Exitosa)</p> <pre>{ "status": "success", "data": { "id": 1, "nombre": "Juan Pérez", "username": JuaPe "email": "usuario@ejemplo.com", "createdAt": "2024-12-20T10:00:00Z" } }</pre> | |
| | <p align="center">Datos de salida (Con error)</p> <pre>{ "status": "error", "message": "El correo electrónico ya está en uso" or "message": "La contraseña debe tener al menos 6 caracteres" or "message": "El nombre de usuario ya está en uso" }</pre> | |

Frontend

- El cliente deberá colocar su email registrado y un nuevo nombre de usuario no utilizado para poder registrarse en la página, además de establecer una contraseña que podrá guardar
- Esta sección será oscura pero no negra, un color bastante oscuro con tonos un poco verdes y azules

Interacción esperada:

El usuario verá los campos que deberá llenar con información válida, hay distintos errores que pueden saltar si: El nombre de usuario ya está registrado, el correo ya está registrado, la contraseña tiene menos de 6 caracteres, en el correo no hay ningún arroba ("@"), etc.

Mockups/Prototipos:

The wireframe shows a 'Sign up' form with the following elements:

- Sign up**: Main title of the form.
- Name and Last Name**: A single input field for the user's full name.
- Email**: An input field for the user's email address.
- Password**: An input field for the user's password, with a note below it stating 'Password should be at least 6 characters'.
- Contact Number**: An input field for the user's contact number.
- Adress**: An input field for the user's address (note the typo).
- Username**: An input field for the user's username, with a note below it stating 'Username may only contain alphanumeric characters'.
- By creating an account, you agree to the Terms of Service.**: A line of text indicating the user's agreement to the terms of service.
- Continue**: A dark button at the bottom of the form to proceed with the registration.

imagen hecha con: <https://wireframe.cc>

Flujo visual y eventos:

Cuando se presiona el botón “continúe”, se verifica que todas las credenciales sean válidas, si es así el usuario verá un “Felicitaciones te has registrado exitosamente, si las credenciales no son válidas el usuario verá un error dependiendo de cuál es la credencial que no es válida:

- Si es el correo puede ver un mensaje como: “Este correo electrónico ya está registrado” o “Correo electrónico invalido”
- Si es el nombre de usuario verá un error como: “Este nombre de usuario ya está registrado” o “Este nombre de usuario contiene un caracter invalido”
- Si es la contraseña: “Esta contraseña no tiene los caracteres suficientes o tiene caracteres no válidos”
- Si es el número de contacto: “En este campo no se pueden usar letras”

Historia de Usuario 3

Descripción conceptual

| | |
|---|---|
| Módulo | <i>Gestión de pedidos</i> |
| Descripción de la(s) funcionalidad(es) requerida(s): | <i>El cliente consulta un histórico de los pedidos realizados por este.</i> |

Descripción técnica

Backend

Aquí se debe explicar de manera técnica lo esperado,

| URL | Método | Código html |
|--|--|-------------|
| localhost:8080/pedidos | GET | 200 |
| Descripción | | |
| Al consultar, se debe retornar un código 200 y | | |
| Datos de entrada | Datos de salida | |
| | <pre>[{ "order_id": 1, "order_date": "2024-12-15T10:30:00Z", "products": [{ "product_id": 101, "name": "Collar", "quantity": 1 }, { "product_id": 202, "name": "Manilla", "quantity": 2 }], "order_status": "Pendiente", "delivery_deadline": "2024-12-22T23:59:59Z", "total_amount": 350.00, "payment_method": "Transferencia" }]</pre> | |

Frontend

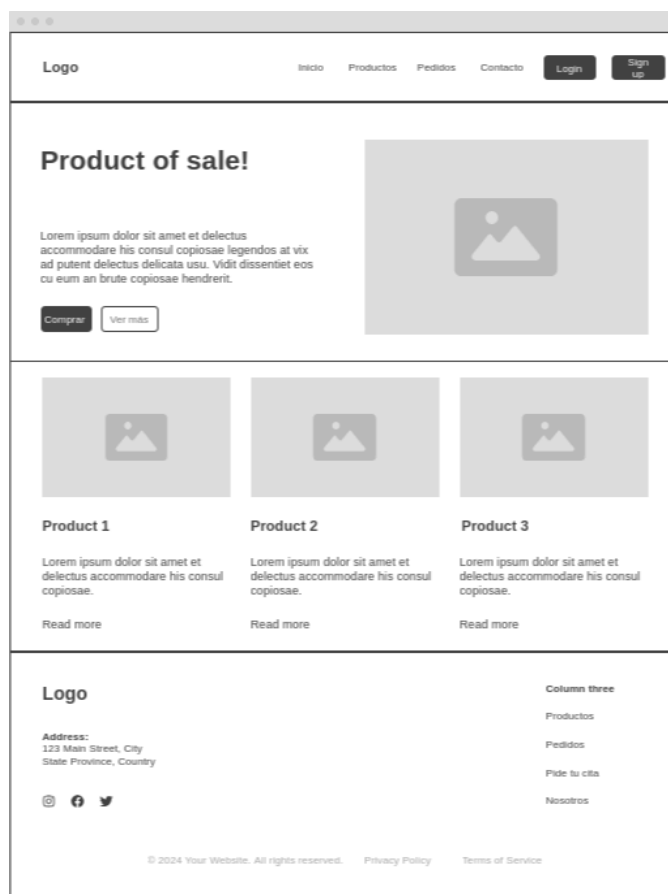
Interacción esperada:

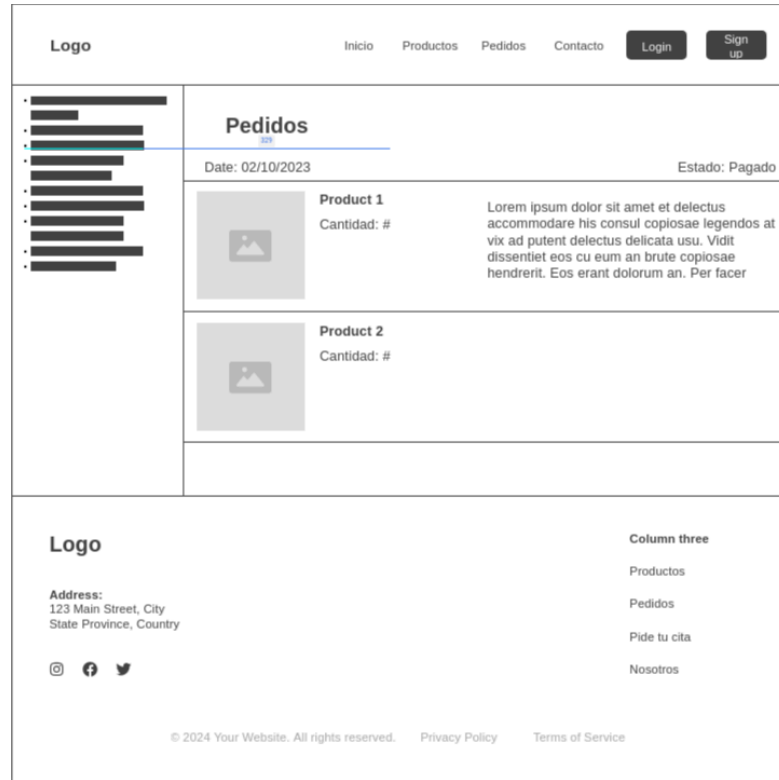
El cliente, una vez logueado en el sistema, encontrándose en la página principal del sistema se dirigirá a la sección de pedidos haciendo click en el botón correspondiente del menú.

Una vez en la página de pedidos tendrá a su disposición una lista de sus pedidos con la lista de productos, cantidades, fecha, estado de pedido, etc.

Además, también podrá visualizar un botón la realización de pagos pendientes en los pedidos con estado "Pendiente".

Mockups/Prototipos:





imagenes hechas con: <https://wireframe.cc>

Flujo visual y eventos:

Se mostrará una línea de carga mientras se consultan los pedidos del cliente.

Historia de Usuario 4

Descripción conceptual

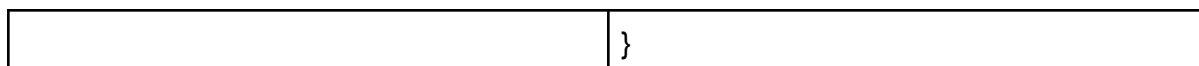
| | |
|---|---|
| Módulo | <i>Módulo de asesorías personalizadas.</i> |
| Descripción de la(s) funcionalidad(es) requerida(s): | <i>Solicitud de una cita personalizada por parte del cliente.</i> |

Descripción técnica

Backend

Aquí se debe explicar de manera técnica lo esperado,

| URL | Método | Código html |
|---|---|---|
| localhost:8080/contactame | POST | 201 (Solicitud creada) 400 (Datos inválidos) |
| Descripción El cliente encuentra un formulario para rellenar con sus datos para que el administrador del negocio programe una cita para la asesoría y se lo comunique. Deberá colocar su nombre, un mensaje al administrador y la fecha y hora en la que solicita una asesoría | | |
| Datos de entrada 200: <pre>{ "name": "Juan Pérez", "message": "Quisiera agendar una cita para una asesoría sobre transporte eléctrico.", "appointment_date": "2024-12-22T10:30:00" }</pre> | Datos de salida <pre>{ "status": "success", "message": "Tu solicitud de cita ha sido recibida. Nos comunicaremos contigo pronto." }</pre> | |
| | 400: <pre>{ "status": "error", "message": "Por favor, completa todos los campos del formulario." }</pre> | |
| | Redirección <pre>{ "status": "redirect", "redirect_url": "http://localhost:8080/thank-you" }</pre> | |



Frontend

Interacción esperada:

El cliente se dirige al formulario de contacto para pedir su cita personalizada haciendo click en “Contactame” en el menú de la aplicación web.

Se encontrará con un formulario de contacto con los siguientes campos:

- *Nombre.*
- *Fecha y hora de la cita.*
- *Mensaje.*

Adicionalmente encontrará dos botones, uno para enviar la información y otro para limpiar los campos del formulario.

Mockups/Prototipos:

Flujo visual y eventos:

- *Cuando no se rellene completamente los datos aparecerá un pop up avisando de la falta de datos obligatorios.*
- *Cuando ocurre un error interno el sistema mostrará un mensaje diciendo “Ha ocurrido un error. Intentalo más tarde. Lamentamos las molestias”.*
- *Mientras se cargan los datos al servidor, los campos de texto se deshabilitarán y se hará más opaco visualmente.*