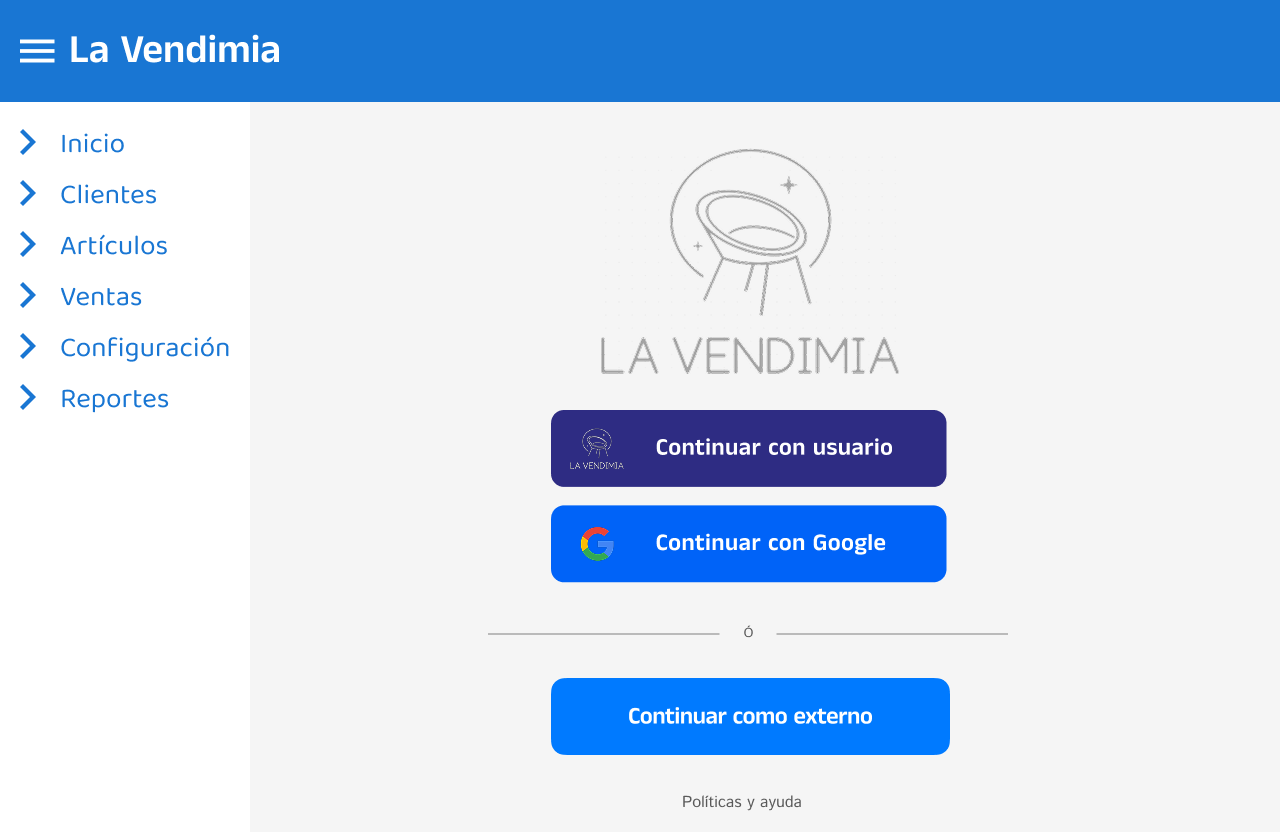
## Diseño gráfico de pantalla sugerido para la funcionalidad

Para visualizar la presentación consulte el siguiente enlace de Figma:

<https://www.figma.com/proto/weJFKAB8pcpp8wrIkKfe5x/La-Vendimia?node-id=12-3359&p=f&t=a114tVPFbfmPDilq-1&scaling=min-zoom&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=1%3A2>

A continuación se presentan las imágenes del prototipo:

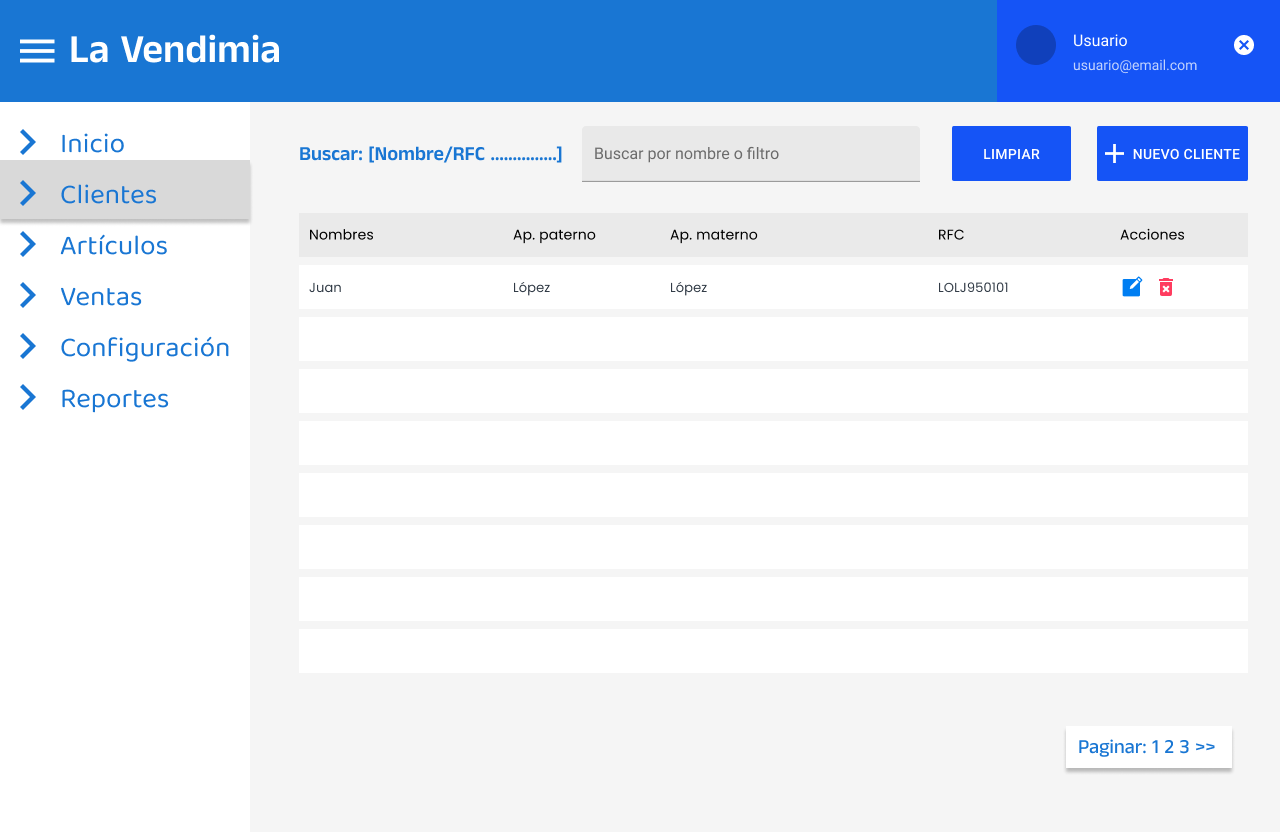
*Inicio sesión*



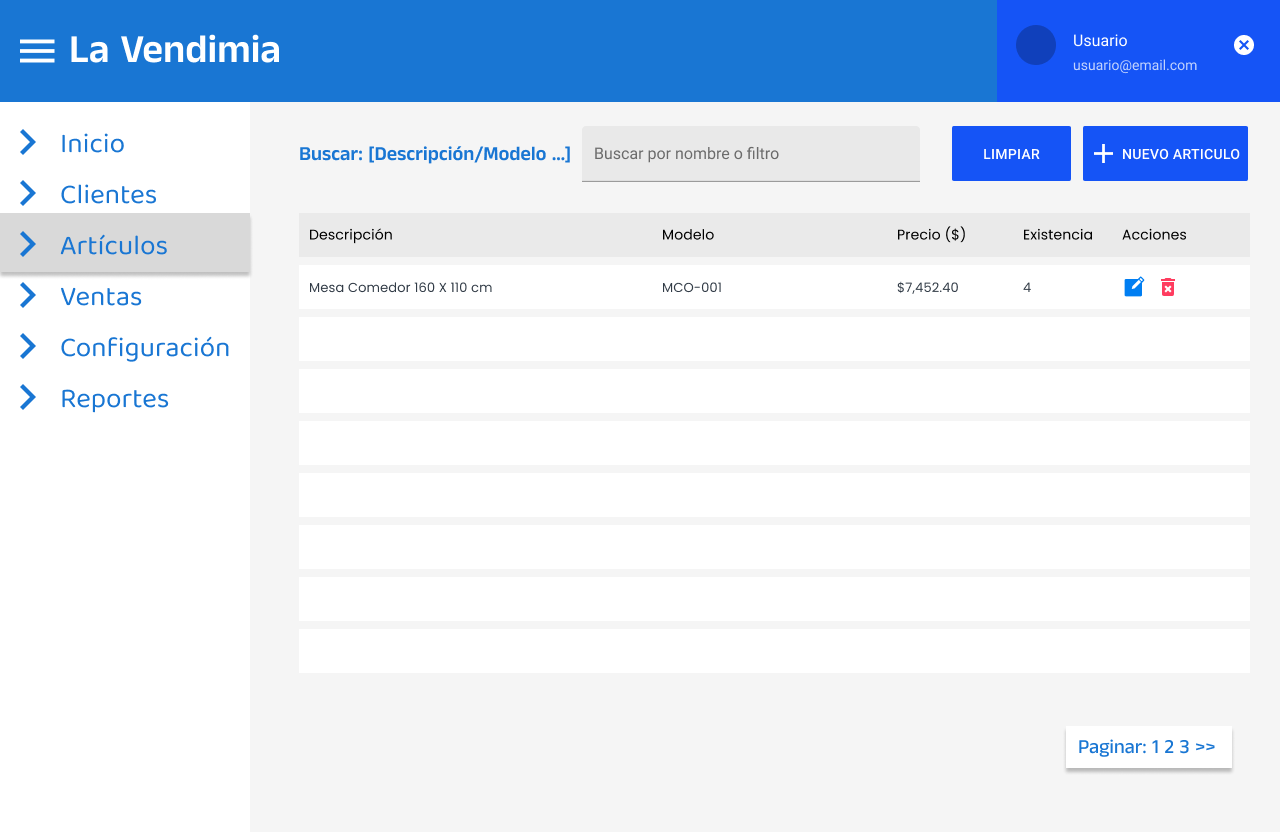
*Inicio - Dashboard*



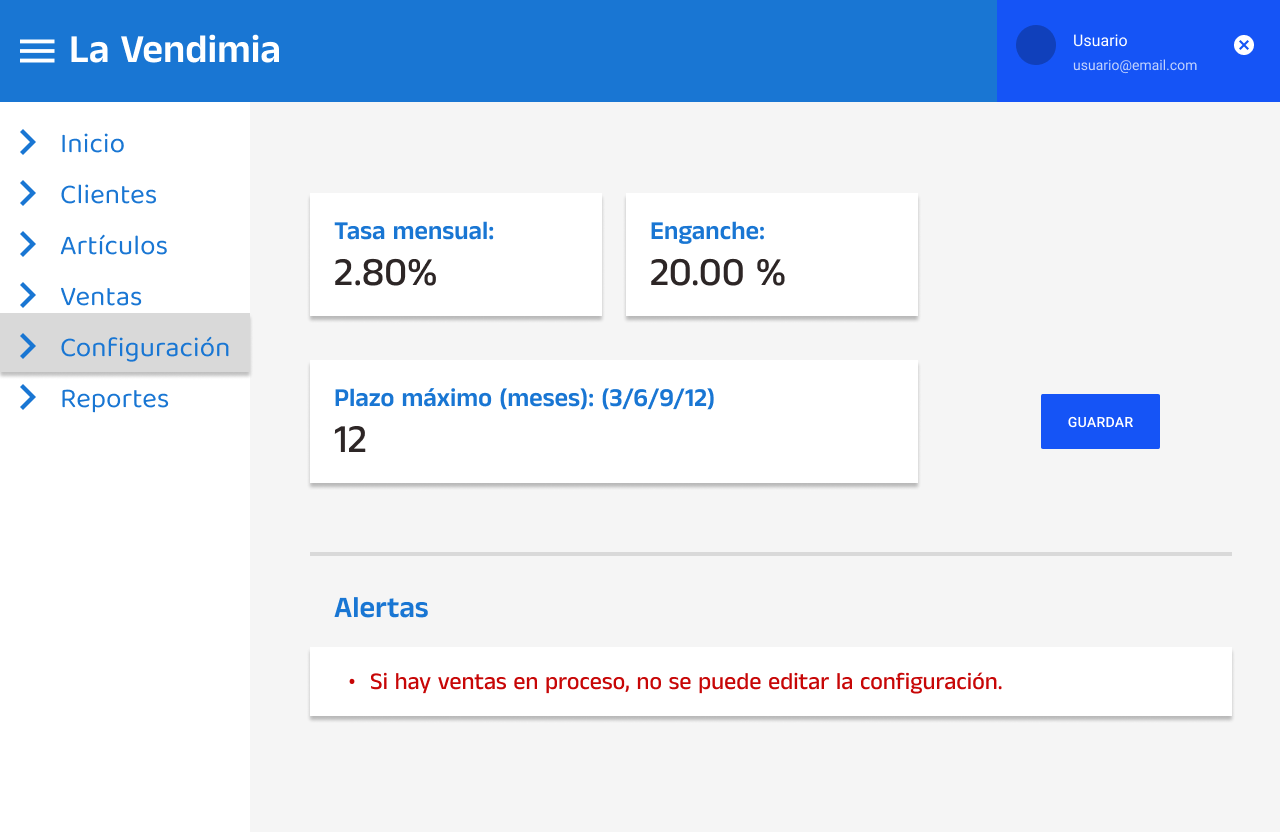
*Clientes*



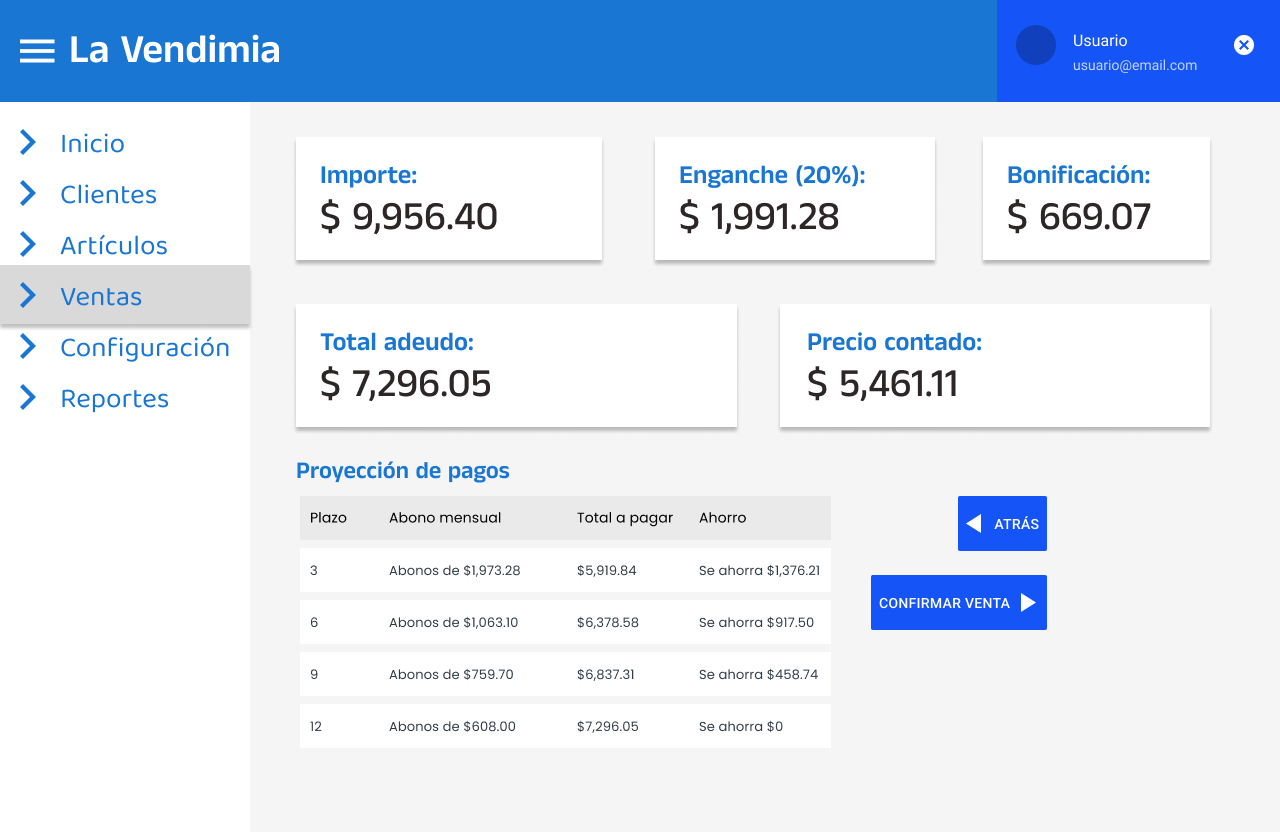
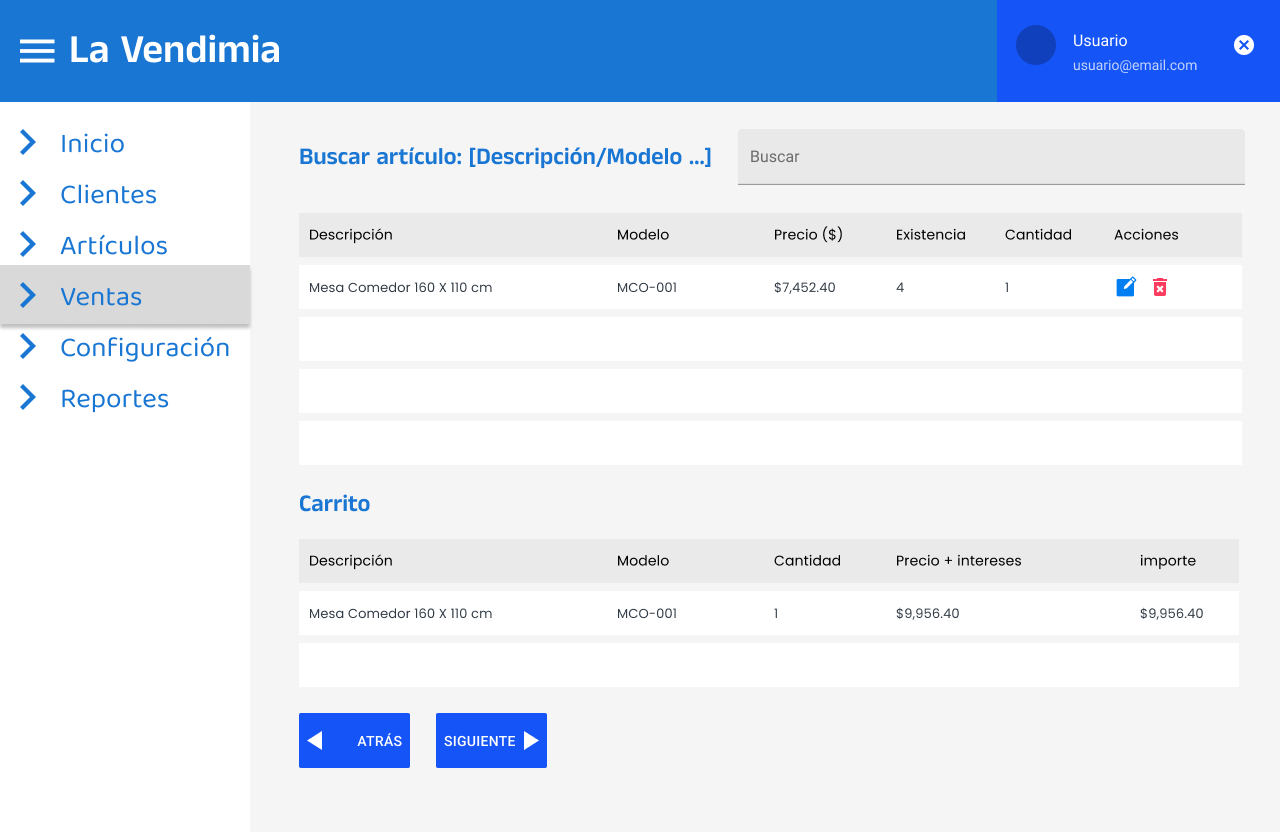
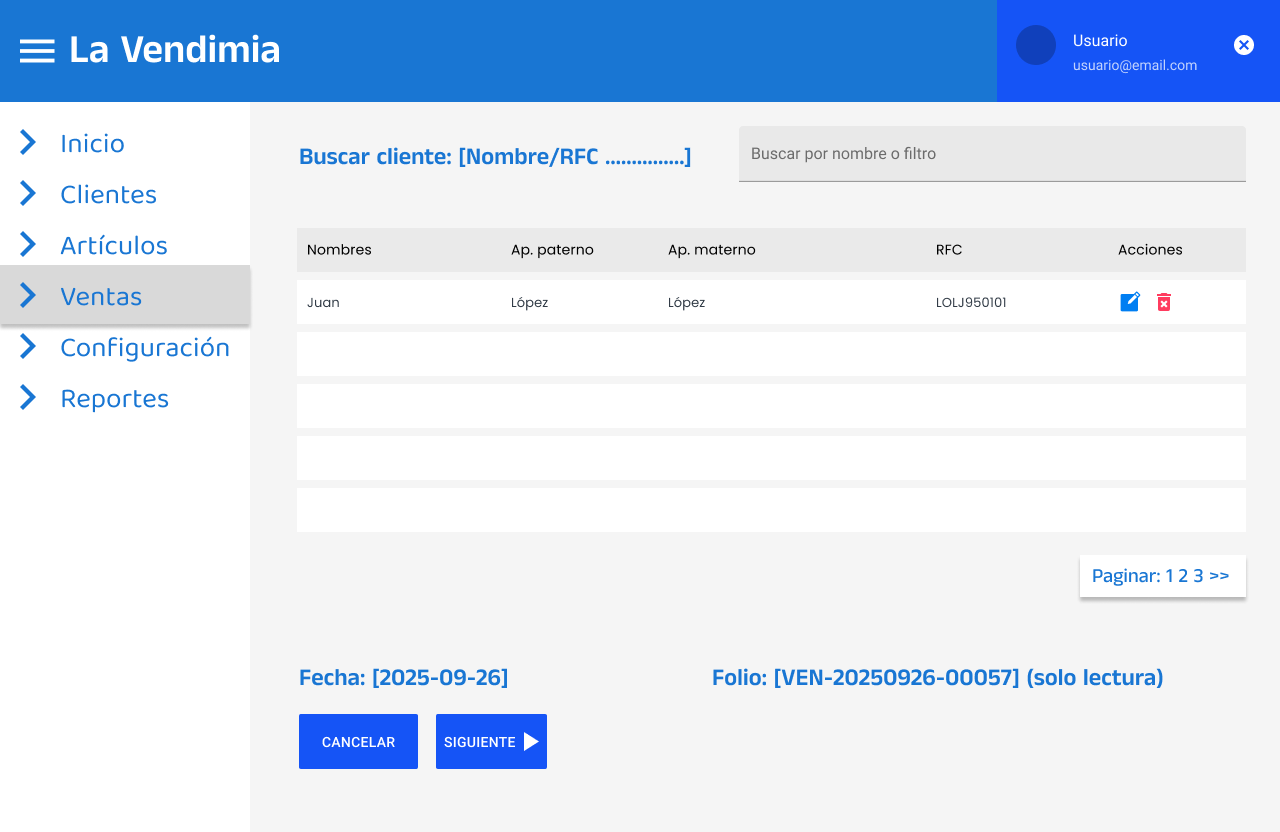
*Artículos*



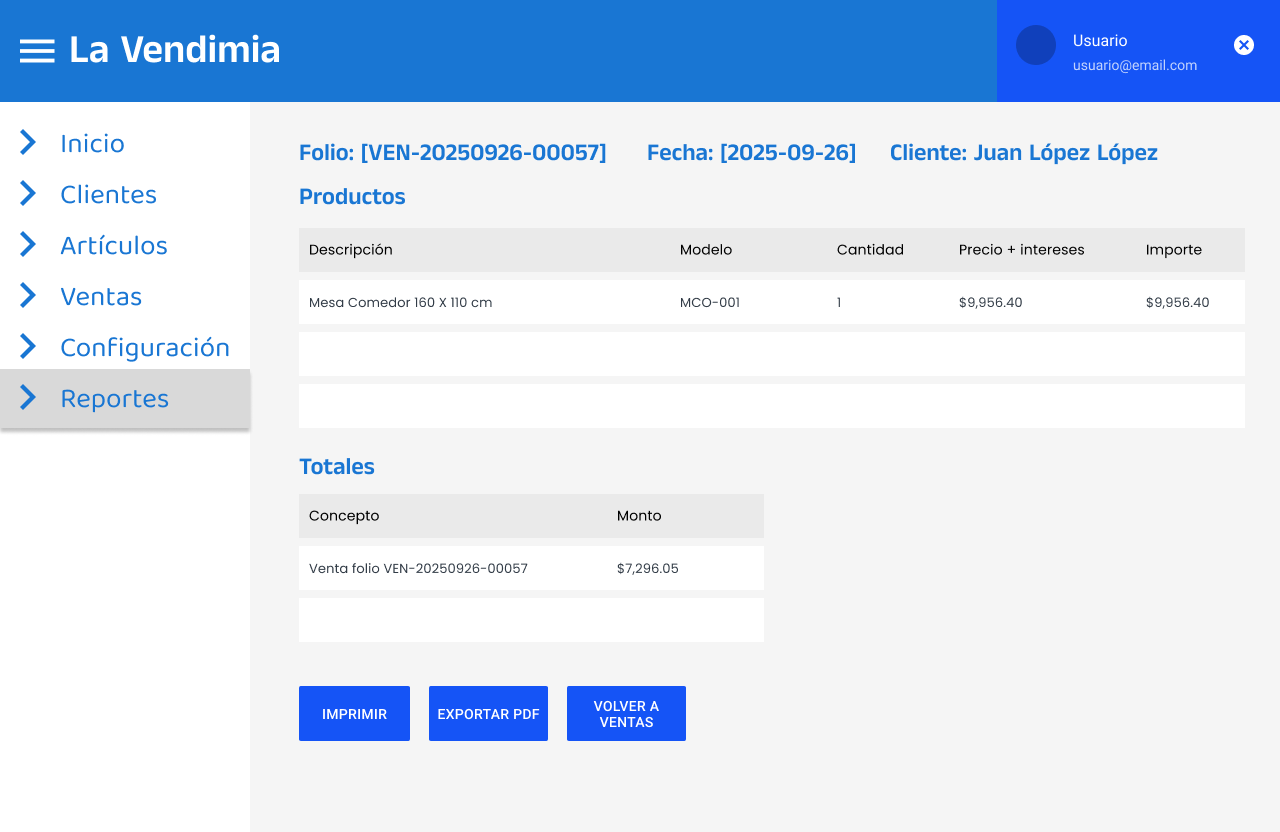
*Configuración*



*Ventas*



*Reportes*



## Requerimientos funcionales y no funcionales detectados en el documento

Requerimientos funcionales:

* Gestión de clientes. (Registrar, consultar, editar y eliminar clientes):
* Datos: Nombre, apellido paterno, apellido materno, RFC.
* Búsqueda de clientes por filtro.
* Gestión de artículos. (Registrar, consultar, editar y eliminar artículos):
* Datos: Descripción, modelo, precio, existencia.
* Búsqueda de artículos por filtro.
* Gestión de configuración. (Parametizable):
* Tasa de financiamiento.
* Porcentaje de enganche.
* Plazo máximo.
* Restricción: no se puede editar la configuración si hay ventas en proceso.
* Gestión de ventas.
* Selección de cliente y artículos.
* Registrar fecha y folio único de la venta.
* Validar existencia de artículos
* Cálculo automático de: precio con intereses, importe total, enganche sugerido, bonificación de enganche, total adeudo.
* Proyección de pagos a 3, 6, 9 y 12 meses: abono mensual, total a pagar, ahorro por elegir ese plazo.
* Cálculos financieros automáticos.

Requerimientos no funcionales:

* Usabilidad. (Interfaz amigable y fácil de usar, con pantallas intuitivas).
* Mantenibilidad. (Parámetros, es decir: interés, enganche, plazos).
* Confiabilidad. (Los cálculos deben ser exactos y consistentes).
* Seguridad:
* Folios de venta únicos e irrepetibles.
* Validación de existencia de artículos antes de confirmar la venta.
* Rendimiento. (Respuestas rápidas al realizar cálculos).
* Escalabilidad. (Posibilidad de migrar a sistema web o móvil sin cambios en la lógica de negocio).
* Disponibilidad. (Sistema accesible en horarios de operación de la empresa).
* Integridad de datos. (No se permiten registros incompletos ni ventas con datos inválidos).

## Historias de usuario

* Gestión de clientes.
* Como vendedor, quiero registrar nuevos clientes con sus datos completos para poder identificarlos en futuras ventas.
* Consulta de clientes.
* Como vendedor, quiero buscar un cliente por su nombre o RFC para localizarlo rápidamente si revisar toda la lista.
* Gestión de artículos.
* Como administrador, quiero registrar artículos con descripción, modelo precio y existencia para mantener actualizado el cátalogo de productos.
* Consulta de artículos.
* Como vendedor, quiero buscar un artículo en el catálogo para ofrecer al cliente la información precisa de su precio y existencia.
* Configuración del sistema.
* Como administrador, quiero configurar la tasa de financiamiento, el porcentaje de enganche y el plazo máximo para adaptar el sistema a cambios del mercado.
* Validación de existencia.
* Como vendedor, quiero que el sistema valide la existencia de los artículos seleccionados para evitar registrar ventas de productos agotados.
* Cálculo automático de financiamiento.
* Como vendedor, quiero que el sistema calcule automáticamente precio con intereses, importe, enganche, bonificación y total adeudo para agilizar el proceso de venta y evitar errores manuales.
* Proyección de pagos.
* Como cliente, quiero que el sistema me muestre proyecciones de abonos en plazos de 3, 6, 9 y 12 meses para decidir la opción de pago más conveniente.
* Generación de folio único.
* Como vendedor, quiero que cada venta tenga un folio único e irrepetible para llevar un control ordenado y rastreable de las operaciones.
* Reporte de ahorro.
* Como cliente, quiero ver cuánto dinero ahorro si elijo plazos más cortos para tomar una mejor decisión financiera.

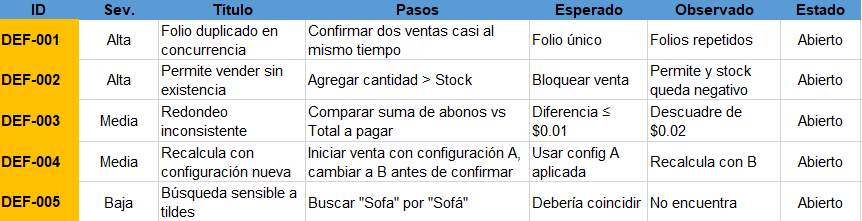
## Insumos de pruebas

* Configuración base.
* Tasa = 2.8 mensual
* Enganche = 20%
* Plazo máximo = 12
* Clientes.
* Si un RFC es el mismo, se puede consultar los de otros clientes teniendo otros nombres.
* Nombres con acentos y mayúsculas / minúsculas.
* Artículos.
* A1: “Mesa Comedor 160 X 110 cm”, MCO-001, 7,452.40, 1
* Ventas – escenarios.
* Venta con 1 artículo y cantidad válida.
* Venta con cantidad > existencia (debe de fallar).
* Venta multiartículo.
* Venta iniciada con una configuración y confirmada tras cambiarla (debe usar la configuración inicial).

## Plan de pruebas de ejecución

* Alcance.
* CRUD clientes, CRUD artículos, configuración (con bloqueo), ventas (folio único, validación de existencia), cálculos y proyección (3/6/9/12).
* Tipos.
* Unitarias (cálculo y validación), integración (ventas + stock), sistema (end-to-end), UAT (con usuario de negocio).
* Criterios de entrada.
* Historias listas, endpoints estables, datos cargados.
* Criterios de salida.
* Ningún defecto crítico, 100% de pruebas pasadas, cálculos verificados con ejemplos del documento.
* Métricas.
* Porcentaje de casos pasados, defectos por severidad, coherencia entre totales y abonos.

## Simulación para levantar defectos con su decripción



## Simulación reporte de resultados de las pruebas

* Alcance: CRUD clientes, CRUD artículos, configuración, flujo de venta end-to-end, proyección 3/6/9/12, cálculos financieros.
* Ejecutados: 84.
* Pasados: 73 (86.9%).
* Fallados: 11 (13.1%).
* Defectos: 2 altos, 3 medios, 6 bajos.
* Riesgos: folio único y validación de stock.
* Recomendación: transacciones y claves únicas a nivel BD; revalidación de stock al confirmar.

## Lista de errores en la revisión del documento

* Error de montos en ejemplo de Bonificación/Enganche:
* Por ejemplo hay una línea que usa “Enganche = 5,678” pero el cálculo correcto es 1,135.60 y esa cifra es la que usa después el documento.
* Sin reglas para cancelaciones o devoluciones:
* Reverso de existencia o bonificaciones.
* Formato de folio no definifdo:
* Por ejemplo no hay ningún formato como VEN-YYYYMMDD-#### o algo que se asemeje.
* Impuestos (IVA), sin ser mencionados.
* Autenticación y roles no definidos.

## Documentación necesaria para que el equipo de desarrollo de sistemas pueda realizar pruebas unitarias e integrales del sistema de La Vendimia

* Alcance y objetivos de pruebas.
* Probandose servicios de cálculo financiero, validadores (RFC, existencia), generación de folio, CRUDs, endpoints de ventas y proyección.
* Arquitectura y módulos a probar.
* Diagrama breve (API <-> Servicio de cálculo <-> Base de datos).
* Límites por módulo (puntos de entrada y dependencias) definiendo que va a Unit Test y que va a pruebas de integración (API + BD).
* Contratos de API (para integración).
* OpenAPI/Swagger del backend (endpoints, request/response, status codes, errores).
* Códigos de error y mensajes esperados.
* Reglas de negocio e invariantes testables.
* Unicidad: RFC de cliente, folio de venta.
* Integridad: existencia ≥ 0; cantidad > 0; precio.
* Estados: “venta en proceso” bloquea edición de configuración.
* Búsquedas: tolerancia a mayúsculas/acentos.
* Transacciones: confirmar venta = revalidar stock + garantizar folio único.
* Especificación de cálculos.
* Orden: Precio con intereses -> Importe -> Enganche -> Bonificación -> Total adeudo -> Precio contado -> total(p) -> Abono -> Ahorro.
* Redondeo: interno a 2 decimales.
* Parámetros: tasa mensual, % enganche, plazo máx (3,6,9,12).
* Esquema de Bases de datos.
* DDL resumido (tablas, PK, índices únicos, CHECKs).
* Plan de pruebas unitarias (por módulo).
* Cálculo financiero: precio con intereses, importe, enganche, bonificación, adeudo, proyección (3/6/9/12) con asserts exactos.
* Validadores: RFC único, campos obligatorios, precio/existencia)
* Generador de folio: formato y colisión (simular concurrencia con mocks).
* Busqueda/filtros: coincidencias por nombre/RFC; acentos y mayúsculas (si aplica).
* Plan de pruebas de integración.
* Flujos API + BD (con Postman o bruno y asserts sobre BD).
* Criterios de funcionamiento: status codes, payloads, efectos en bases de datos.
* Herramientas y entregables propuestos.
* Colección Postman.
* Un OpenApi.yaml
* JSON y CSVs, asi como las consultas.
* Gestión de defectos.
* Plantilla de bug: (ID, severidad, prioridad, módulo, pasos, esperado, observado, evidencia, estado, responsable).
* Flujo de estados: Abierto -> En progreso -> Resuelto -> Verificado -> Cerrado.