

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
INGENIERÍA EN CIENCIAS Y SISTEMAS**

ANALISIS Y DISEÑO DE SISTEMAS 2



MANUAL TÉCNICO FRONTEND

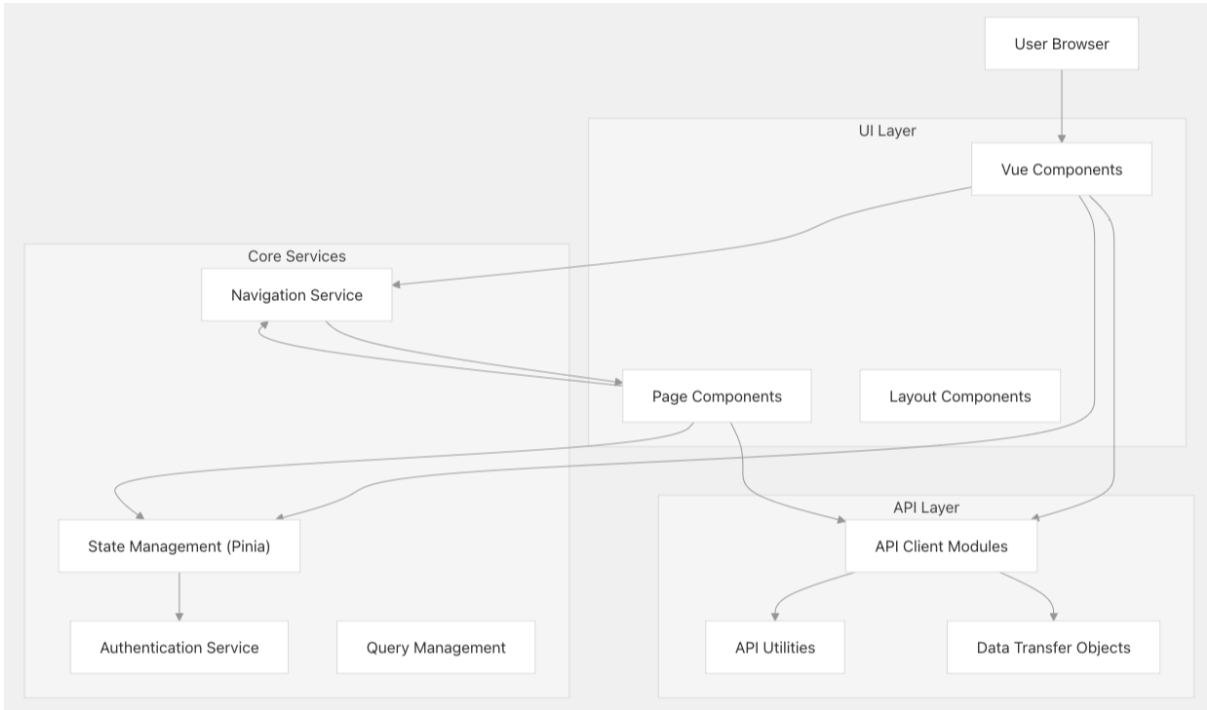
Quetzaltenango 15 de mayo de 2025

202030542 - Fernando José Rodríguez Ramírez
202031794 - Luis Antonio Monterroso Guzman
201931012 - Carlos Benjamín Pac Flores

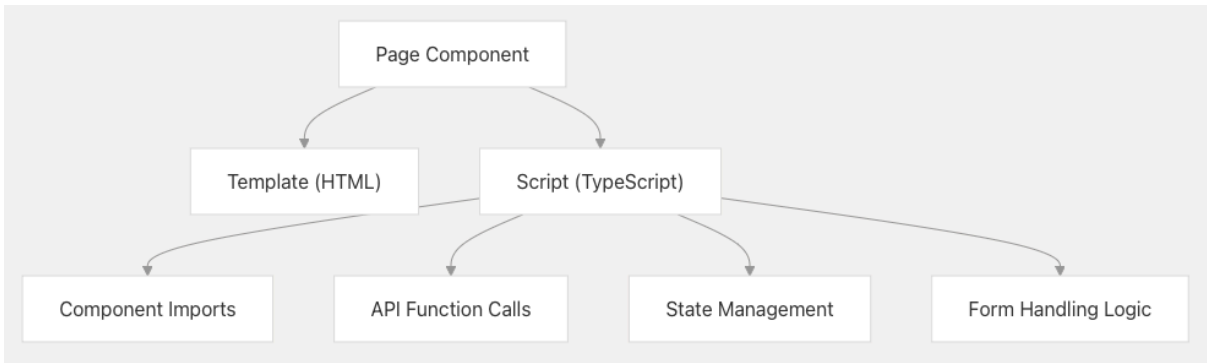
Descripción General

Arquitectura

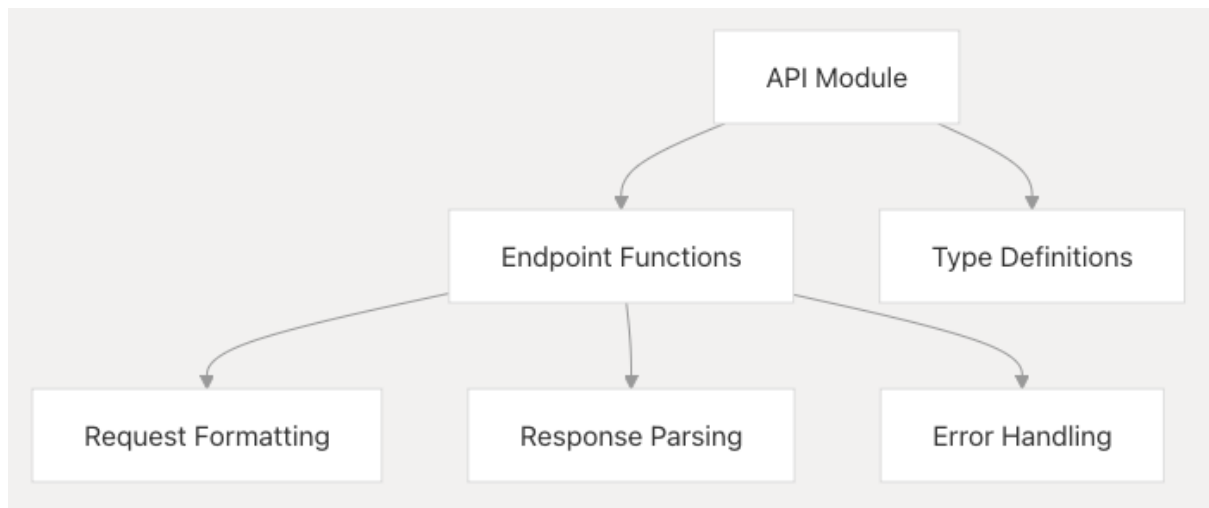
Capas de la arquitectura



Capa de Interfaz de Usuario



Capa de API



Despliegue

Una aplicación frontend basada en Vue.js que interactúa con diversos servicios API del backend. Para configurar el entorno de desarrollo, se requieren las siguientes herramientas y tecnologías:

- Node.js v22.14.0
- npm (Node Package Manager)
- Git para control de versiones
- Framework Vue.js

Configuración Local de Desarrollo

Para clonar y ejecutar el proyecto localmente:

1. Clonar el repositorio
2. Instalar dependencias mediante `npm install`
3. Ejecutar el servidor de desarrollo con `npm run dev`
4. Construir la versión de producción con `npm run build`

Flujo de Desarrollo

El flujo de desarrollo sigue un enfoque estructurado, con una estrategia de ramas centrada en la rama dev como rama principal de integración.

Pasos del flujo de desarrollo:

1. El desarrollador crea una rama de funcionalidad
2. Implementa los cambios
3. Ejecuta pruebas locales
4. Realiza commit y push a la rama de funcionalidad
5. Crea una Pull Request hacia la rama dev

6. Revisión y aprobación de código
7. Fusión a la rama dev
8. Jenkins ejecuta la pipeline automáticamente
9. Despliegue automatizado

Integración y Despliegue Continuo (CI/CD)

El proyecto utiliza **Jenkins** como herramienta de integración y despliegue continuo. La pipeline está definida en un archivo Jenkinsfile y automatiza el proceso de construcción y despliegue cada vez que se realizan cambios en la rama dev.

Descripción general de la CI/CD

Infraestructura de Despliegue:

- Jenkins Pipeline
- Checkout del código
- Instalación de dependencias
- Compilación del frontend
- Despliegue automático del frontend
- Gestión del proceso mediante PM2
- Servidor web para archivos compilados

Etapas de la Pipeline

La pipeline de Jenkins se compone de cuatro etapas principales:

1. **Checkout:** Recupera el código más reciente de la rama dev
2. **Instalación de dependencias:** Instala o actualiza los paquetes npm
3. **Construcción del frontend:** Compila y genera la versión de producción
4. **Despliegue del frontend:** Copia la aplicación al entorno de producción y reinicia el proceso

Instalación de Dependencias

La pipeline optimiza la instalación de dependencias al verificar si hubo cambios en package.json. Si no se detectan cambios, se omite la reinstalación para acelerar el proceso.

Cambio detectado en package.json	Acción
Sí	Elimina node_modules y ejecuta npm install

No	Se omite la instalación y se continúa con la construcción
----	---

Proceso de Construcción

El proceso de construcción utiliza Node.js mediante nvm (Node Version Manager) para garantizar consistencia en la versión de Node.js utilizada durante los despliegues.

- **Versión de Node.js:** 22.14.0
- **Comando de construcción:** npm run build

Proceso de Despliegue

Durante la etapa de despliegue, se realizan las siguientes acciones:

- Detener el proceso actual de PM2 si existe
- Limpiar el directorio de despliegue
- Copiar los archivos generados al destino
- Reiniciar el proceso usando PM2

Componente	Valor
Nombre de la App en PM2	front-p2ayd2
Directorio de despliegue	/var/www/project2AyD2
Directorio de build	.output/

Configuración del Entorno

La pipeline de Jenkins utiliza las siguientes variables de entorno:

Variable	Descripción	Valor
NVM_DIR	Directorio de instalación de NVM	/var/lib/jenkins/.nvm

DEPLOY_FRONTEND_DIR	Directorio de despliegue del frontend	/var/www/project2AyD2
NODE_VERSION	Versión de Node.js	22.14.0
PM2_APP_NAME	Nombre de la aplicación en PM2	front-p2ayd2

Arquitectura de Despliegue

La arquitectura de despliegue emplea **PM2 (Process Manager 2)** para gestionar la aplicación Node.js en producción. PM2 se encarga del manejo de procesos, reinicios automáticos y registro de logs.

Infraestructura:

- Servidor de Producción
- Servidor Jenkins
- Jenkins ejecuta la pipeline
- Los archivos construidos se copian al servidor de producción
- PM2 ejecuta y supervisa la aplicación
- Los archivos se sirven desde el directorio web correspondiente

Verificación Post-Despliegue

Después del despliegue, la pipeline realiza una verificación básica consultando los logs de PM2 para asegurar que la aplicación haya iniciado correctamente. Esto permite detectar rápidamente posibles errores de despliegue.

Sistemas de Lógica del Negocio

Dashboard

El sistema de Dashboard proporciona una interfaz unificada que conecta a los usuarios con todas las áreas funcionales principales de la aplicación. Tras la autenticación, el usuario visualiza una cuadrícula de tarjetas de acceso rápido que permiten ingresar con un solo clic a funciones administrativas y operativas.

Áreas accesibles desde el Dashboard:

- Autenticación
- Dashboard

- Administración
- Reportes
- Bodegas
- Proveedores
- Cajas Registradoras
- Productos
- Gestión de Stock
- Entradas
- Salidas
- Entradas de Productos
- Reservaciones
- Juegos
- Horarios
- Paquetes
- Facturación

Estructura del Dashboard

El dashboard está implementado como un componente Vue que muestra:

- Un mensaje de bienvenida personalizado con el nombre de usuario autenticado
- Una cuadrícula con tarjetas de acceso rápido, cada una enlazando a una sección distinta del sistema

Jerarquía de Componentes del Dashboard

- index.vue – Página principal del Dashboard
- **Almacenamiento de autenticación** – Maneja la información del usuario
- **Componente MenuShortcutCard** – Representa cada tarjeta de acceso
- **Datos de configuración de menús**

Estructura del Menú de Navegación

El dashboard brinda acceso a **15 áreas funcionales** mediante una cuadrícula de tarjetas. Cada ítem del menú incluye:

- **Título:** Nombre del área funcional
- **Descripción:** Breve explicación de su propósito
- **Ruta:** Dirección dentro de la aplicación

Título	Descripción	Ruta
Administración	Administración general del hospital	/admin

Reportes	Generación de reportes	/reportes
Bodegas	Administración de bodegas	/bodegas
Proveedores	Administración de proveedores	/proveedores
Cajas	Administración de cajas	/cajas
Productos	Administración de productos	/productos
Stock	Administración de inventario	/stocks
Entradas	Registro de entradas	/entradas
Salidas	Registro de salidas	/salidas
Entradas Productos	Entradas detalladas de productos	/entradas_producto s
Reservaciones	Gestión de reservaciones	/reservaciones
Juegos	Gestión de juegos	/juegos
Horarios	Gestión de horarios	/horarios
Paquetes	Gestión de paquetes	/paquetes

Facturación	Gestión de facturas	/facturacion
-------------	---------------------	--------------

Integración con Sistemas Clave del Negocio

El dashboard integra múltiples sistemas del negocio proporcionando navegación directa a sus interfaces correspondientes.

Ejemplo de Integración: Sistema de Stock

Relación entre componentes:

- index.vue: Dashboard principal
- MenuShortcutCard.vue: Tarjetas de navegación
- stocks/index.vue: Interfaz del sistema de stock
- stock.ts: Funciones API para stock

Funciones específicas del sistema de stock:

- Monitoreo de productos con bajo stock
- Vista por bodega del stock disponible
- Edición de niveles mínimos de inventario

Contexto de Autenticación del Usuario

El dashboard personaliza la experiencia del usuario autenticado mostrando un mensaje de bienvenida con su nombre, recuperado desde el store de autenticación.

Proceso:

1. Usuario inicia sesión
2. Se recupera información del usuario
3. Se muestra mensaje de bienvenida con su nombre
4. Se renderizan las opciones de menú
5. Usuario selecciona opción
6. Navegación hacia el sistema correspondiente

Detalles de Implementación

Componente Principal del Dashboard

- Implementado en index.vue como la página raíz de la aplicación
- Usa Composition API (<script setup>)
- Importa el componente MenuShortcutCard
- Accede al usuario autenticado desde el store con storeToRefs

- Define un arreglo menus con las opciones de navegación
- Renderiza una cuadrícula de tarjetas pasando cada menú como prop

Ejemplo de Integración con Backend

El dashboard facilita el acceso a distintos endpoints del backend. Por ejemplo, al seleccionar la opción de “Stock”, se navega a una interfaz que se comunica con los siguientes endpoints de API:

Endpoint	Función
GET /v1/stocks/warehouse/{warehouseId}	Obtener stock por bodega
GET /v1/stocks/low-stock	Obtener productos con bajo stock
PATCH /v1/stocks/minimum-stock	Actualizar stock mínimo por producto

Sistema de Gestión de Stock

El **Sistema de Gestión de Stock** proporciona funcionalidades para monitorear y gestionar los niveles de inventario en distintas bodegas. Permite a los usuarios rastrear las cantidades actuales de productos, definir umbrales mínimos de stock, recibir alertas cuando los productos alcanzan niveles críticos, y mantener un inventario óptimo.

Sistemas externos relacionados:

- Sistema de Gestión de Stock
 - Interfaz de usuario de stock (StockUI)
 - API de stock (StockAPI)
 - Integración con bodegas y productos
 - Alertas por bajo stock
- Sistema de Gestión de Bodegas
- Sistema de Gestión de Productos
- Sistema de Entradas de Productos

Modelo de Datos

El sistema opera con las siguientes estructuras clave:

Entidades:

- **Stock**
 - id: string
 - productId: string
 - quantity: number
 - minimumStock: number
 - warehouse: Warehouse
- **UpdateMinStock**
 - productId: string
 - warehouseId: string
 - minimumStock: string
- **Warehouse**
 - id: string
 - name: string
 - ublication: string
 - active: boolean
- **Product**
 - id: string
 - name: string
 - price: number

La entidad principal es **Stock**, que vincula un producto con su cantidad actual en una bodega específica y define un umbral mínimo para generar alertas de bajo stock.

Funcionalidades Principales

Monitoreo de Stock por Bodega

El sistema permite visualizar los niveles actuales de inventario filtrando por bodega. El usuario selecciona una bodega desde un menú desplegable y se muestra una tabla con todos los productos, sus cantidades actuales y umbrales mínimos.

Funcionalidad	Descripción
Selección de Bodega	Menú desplegable para filtrar stock por bodega
Tabla de Stock	Muestra nombre del producto, bodega, cantidad actual y mínimo

Detalles de Stock	Información detallada del inventario en la bodega seleccionada
-------------------	--

Gestión de Niveles Mínimos

Los usuarios pueden definir o actualizar el stock mínimo por producto y bodega. Si el stock cae por debajo de este umbral, el producto aparece en la tabla de alertas.

Flujo:

1. Usuario hace clic en “Editar mínimo”
2. Se abre un diálogo de edición
3. Ingresa el nuevo valor de stock mínimo
4. Guarda los cambios
5. Se actualiza en la base de datos y se refrescan los datos visuales

Alertas por Bajo Stock

El sistema identifica automáticamente los productos con cantidades por debajo del mínimo establecido y los muestra en una tabla separada para facilitar su reposición.

Funcionalidad	Descripción
Tabla de Bajo Stock	Lista productos con cantidad menor al mínimo
Actualizaciones	Se refresca automáticamente cuando se editan los mínimos
Contexto de Bodega	Muestra la bodega para facilitar la reposición contextual

Implementación Técnica

Capa API

El archivo stock.ts expone los siguientes endpoints:

- getStockByWarehouse: Obtener todos los productos por bodega
- getAllProductsLowStock: Obtener productos con bajo stock en todas las bodegas
- getAllProductsLowStockByWarehouse: Obtener productos con bajo stock por bodega

- **updateMinStock:** Actualizar el stock mínimo por producto y bodega

Interfaz de Usuario

La UI se divide en dos secciones:

1. **Stock por Bodega**
 - Permite filtrar por bodega y visualizar el stock
 - Posibilidad de editar los mínimos
2. **Productos con Bajo Stock**
 - Lista productos por debajo del mínimo definido

Se utilizan componentes de **PrimeVue** como DataTable, Dropdown, Dialog e InputNumber para una experiencia interactiva.

Flujo técnico:

- **onMounted():** carga datos iniciales
- **refetchStock():** actualiza el stock
- **saveMinimumStock():** guarda cambios
- **getStockByWarehouse(), fetchLowStock() y updateMinStock()** conectan con la API

Integración con Otros Sistemas

El sistema se integra con:

- **Gestión de Bodegas:** El stock se agrupa y filtra por ID de bodega
- **Gestión de Productos:** Cada ítem de stock hace referencia a un producto
- **Entradas de Productos:** Al recibir productos, se actualiza automáticamente el stock

Flujo de Trabajo del Usuario

1. Acceder al sistema desde el Dashboard
2. Seleccionar una bodega
3. Revisar los niveles de stock
4. Identificar productos con bajo stock
5. Ajustar umbrales mínimos si es necesario
6. Guardar los cambios

Sistema de Reportes

El **Sistema de Reportes** proporciona capacidades de inteligencia de negocios para la aplicación, permitiendo a los usuarios generar, visualizar y exportar diversos reportes analíticos. Este sistema se enfoca principalmente en la analítica relacionada con las reservaciones, incluyendo métricas de desempeño financiero, comportamiento del cliente y eficiencia operativa. Su objetivo es apoyar la toma de decisiones estratégicas por parte de la administración.

Tipos de Reportes

El sistema soporta **seis tipos de reportes** distintos, cada uno ofreciendo una perspectiva analítica diferente de los datos del negocio:

Tipo de Reporte	Identificador Interno	Propósito
Reporte de Reservasiones	RESERVATIONS	Lista detallada de todas las reservaciones
Reporte de Horas Populares	POPULAR_HOURS	Análisis de los horarios con mayor demanda
Reporte de Ganancias	RESERVATIONS_PROFIT	Análisis financiero de ingresos por reservaciones
Reporte de No Shows	NOT_SHOW	Análisis de clientes que no asistieron a su reservación
Reporte de Tiempo Promedio	AVERAGE_TIME	Promedio de tiempo de permanencia por grupo
Reporte de Clientes Frecuentes	FREQUENT	Análisis de clientes recurrentes y su frecuencia de visita

Arquitectura del Sistema

El sistema de reportes sigue un patrón de arquitectura en capas, separando claramente los componentes de interfaz de usuario y los servicios API.

Componentes principales:

- Servicios backend
- Capa API (reporte.ts)
- Capa de interfaz de usuario (reportes/index.vue)

- Filtros de fecha y selector de tipo de reporte
- Tabla de datos con columnas dinámicas
- Estadísticas de resumen
- Botón de exportación a PDF

Flujo del Proceso de Reportes

1. El usuario selecciona el tipo de reporte
2. Define un rango de fechas (opcional)
3. Da clic en el botón **Filtrar**
4. Se llama a la función API correspondiente
5. Los datos son procesados y mostrados en tabla
6. Se visualizan estadísticas adicionales
7. El usuario puede exportar el resultado en PDF

Estructura de Datos

Cada tipo de reporte cuenta con su propia configuración de tabla y estructura de datos:

Tipo de Reporte	Columna Clave	Campos Principales
Reporte de Reservas	reservationId	id, fecha, hora inicio, hora fin, nombre cliente, NIT, noShow
Horas Populares	startTime	hora inicio, hora fin, total
Reporte de Ganancias	reservationId	id, fecha, hora, cliente, método de pago, total
Reporte de No Show	reservationId	id, fecha, hora, cliente, NIT, noShow
Tiempo Promedio	date	fecha, horas promedio
Cientes Frecuentes	customerId	id cliente, nombre cliente, visitas totales

Flujo de Datos y Estado

El sistema mantiene su estado a través de variables reactivas que controlan la visualización de datos y componentes UI:

Variables clave:

- reportType
- startDate y endDate
- reportData
- tableConfig
- globalFinancialSummary, totalReservations, totalAverage

Funciones clave:

- cargarReporteActual()
- filtrar()
- exportReports()
- recargarDatos()

Funciones API

Funciones de obtención de datos

Función	Parámetros	Descripción
getReservationReport	startDate?, endDate?	Obtiene el reporte de reservaciones
getPopularHoursBetweenDates	startDate?, endDate?	Horarios de alta demanda
createReservationProfitReport	startDate?, endDate?	Reporte de ingresos por reservaciones
createNotShowReport	startDate?, endDate?	Clientes que no asistieron
getAverageTimeReport	startDate?, endDate?	Tiempo promedio de permanencia

createFrequentCustomersReport	startDate?, endDate?	Clientes frecuentes
-------------------------------	-------------------------	---------------------

Funciones de exportación PDF

Función	Parámetros	Descripción
exportReservationReport	startDate?, endDate?	Exporta el reporte de reservaciones en PDF
exportPopularHoursBetweenDates	startDate?, endDate?	Exporta reporte de horas populares
exportReservationProfitReport	startDate?, endDate?	Exporta reporte de ganancias
exportNotShowReport	startDate?, endDate?	Exporta reporte de no shows
exportAveregangeTimeReport	startDate?, endDate?	Exporta tiempo promedio
exportFrequentCustomersReport	startDate?, endDate?	Exporta clientes frecuentes

Componentes de la Interfaz de Usuario

El sistema cuenta con una sola vista que integra múltiples elementos interactivos:

- **Selector de tipo de reporte:** permite elegir el análisis deseado
- **Filtros por fecha:** seleccionan el rango de análisis
- **Botones de acción:**
 - *Exportar PDF:* descarga el reporte actual
 - *Filtrar:* aplica filtros para generar el reporte
 - *Restablecer:* limpia filtros y reinicia los datos

- **Tabla de datos:** genera columnas dinámicas según el reporte
- **Estadísticas resumen:** muestra totales y promedios

Integración con Otros Sistemas

El sistema se conecta con distintos módulos para obtener y analizar datos:

- **Sistema de Reservaciones:** fuente principal de datos sobre fechas, horas y clientes
- **Sistema de Juegos:** datos sobre asistencia y desempeño en juegos
- **Sistema de Facturación:** datos financieros para reportes económicos
- **Base de Datos de Clientes:** análisis de comportamiento y frecuencia de visitas

Sistema de Reservaciones

El **Sistema de Reservaciones** gestiona todo el ciclo de vida de una reservación para sesiones de juego, desde su creación hasta la finalización del pago. Administra tanto reservaciones presenciales como en línea, ofreciendo interfaces tanto para el personal administrativo como para usuarios públicos. Este sistema se integra estrechamente con el sistema de gestión de horarios, el sistema de facturación y el sistema de juegos.

Arquitectura del Sistema

El sistema de reservaciones está compuesto por múltiples componentes interconectados que gestionan distintas fases del proceso:

- **API de Reservaciones**
- **Componentes de interfaz de usuario (UI)**
- **Modelo de datos de reservaciones**
- **Sistema de Horarios**
- **Sistema de Juegos**
- **Sistema de Facturación**
- **Sistema de Paquetes**
- **Sistema de Productos**
- **Usuarios Administrativos y Públicos**

Modelo de Datos

El sistema utiliza varias estructuras de datos para representar una reservación, incluyendo campos como:

- ID de reservación
- Información del cliente
- Horario seleccionado
- Productos o paquetes asociados
- Estado de pago
- Asistencia

Endpoints API de Reservasiones

El sistema expone los siguientes endpoints para gestionar las reservasiones:

Función	Endpoint	Método	Descripción
createReservation	/v1/reservations/presenti al	POST	Crear una reservación presencial
createReservationOnli ne	/v1/reservations/online	POST	Crear reservación en línea con jugadores
cancelReservation	/v1/reservations/cancel/: id	PATCH	Cancelar una reservación existente
payReservation	/v1/reservations/pay	PATCH	Procesar el pago de una reservación
getReservationById	/v1/reservations/:id	GET	Obtener detalles de una reservación específica
getAllReservations	/v1/reservations/all	GET	Obtener todas las reservasiones (filtrables)
deleteReservation	/v1/reservations/:id	DELET E	Eliminar una reservación

Proceso de Creación de Reservasiones

Reservación Presencial

Estas son creadas por el personal administrativo e implican tres pasos en la interfaz:

1. **Ingreso de Datos:** Fecha, cliente y horario disponible

2. **Selección de Elementos:** Agregar productos y/o paquetes
3. **Facturación:** Elegir método de pago y completar la reservación

Flujo resumido:

- Se ingresan los datos básicos
- Se consultan horarios disponibles
- Se selecciona un horario
- Se agregan paquetes/productos
- Se selecciona el método de pago
- Se envía la reservación
- Se genera y descarga la factura en PDF

Reservación en Línea

Realizadas por usuarios públicos, estas reservaciones se centran en registrar jugadores y utilizan un proceso de dos pasos:

1. **Ingreso de Datos:** Fecha, cliente, horario
2. **Registro de Jugadores:** Hasta 4 jugadores por sesión

Flujo resumido:

- El usuario público completa el formulario
- Se consultan y seleccionan horarios
- Se ingresan datos del cliente y jugadores
- Se envía la solicitud
- Se crea la sesión de juego con jugadores
- Se muestra confirmación de reservación

Gestión de Reservaciones

Visualización

Los administradores pueden ver:

- Detalles de la reservación (ID, fecha, hora)
- Información del cliente
- Estado de pago
- Estado de asistencia

Completar el Pago

Para reservaciones sin pagar, el administrador puede:

1. Ver los detalles
2. Agregar productos/paquetes si es necesario
3. Elegir método de pago

4. Confirmar el pago
5. Generar factura y mostrar PDF

Control de Acceso

El sistema implementa un middleware de autenticación para:

- Permitir acceso a la UI pública sin token
- Requerir autenticación para la UI administrativa
- Verificar token válido para rutas protegidas
- Redirigir al login si el acceso no está autorizado

Puntos de Integración

El sistema de reservaciones se integra con los siguientes módulos:

Sistema	Propósito de Integración
Gestión de Horarios	Selección de horarios para las reservaciones
Gestión de Paquetes	Agregar paquetes de servicios a la reservación
Gestión de Productos	Agregar productos individuales a la reservación
Sistema de Facturación	Generar facturas tras completar el pago
Sistema de Juegos	Crear sesiones de juego con jugadores desde reservaciones en línea

Sistema de Reservaciones

El **Sistema de Reservaciones** gestiona todo el ciclo de vida de las reservaciones para sesiones de juego, desde su creación hasta la finalización del pago. Administra tanto reservaciones presenciales como en línea, y ofrece interfaces tanto para el personal administrativo como para los usuarios públicos. Este sistema se integra estrechamente con

el sistema de gestión de horarios, el sistema de facturación y el sistema de gestión de juegos.

Arquitectura del Sistema

El sistema está compuesto por múltiples componentes interconectados que manejan distintos aspectos del proceso de reservación:

- API de Reservaciones
- Componentes de Interfaz de Usuario
- Modelo de Datos de Reservación
- Sistema de Horarios
- Sistema de Juegos
- Sistema de Facturación
- Sistema de Paquetes
- Sistema de Productos
- Usuario Administrativo
- Usuario Público
- Interfaz Pública de Reservaciones

Modelo de Datos

El sistema utiliza diversas estructuras de datos para gestionar la información de las reservaciones, que incluyen:

- Identificador de reservación
- Fecha y hora
- Cliente
- Estado de pago
- Asistencia
- Productos y paquetes asociados
- Información de jugadores (en reservaciones en línea)

Endpoints de la API de Reservaciones

El sistema expone los siguientes endpoints para interactuar con las reservaciones:

Función	Endpoint	Método	Descripción
createReservation	/v1/reservations/presentia	POST	Crea una reservación presencial

createReservationOnline	/v1/reservations/online	POST	Crea una reservación en línea con jugadores
cancelReservation	/v1/reservations/cancel/:id	PATCH	Cancela una reservación existente
payReservation	/v1/reservations/pay	PATCH	Procesa el pago de una reservación
getReservationById	/v1/reservations/:id	GET	Obtiene una reservación específica
getAllReservations	/v1/reservations/all	GET	Lista todas las reservaciones con filtros
deleteReservation	/v1/reservations/:id	DELETE	Elimina una reservación

Proceso de Creación de Reservaciones

Reservación Presencial

Las reservaciones presenciales son creadas por personal administrativo y siguen un flujo de tres pasos en la interfaz:

1. **Recolección de Datos:** Fecha, cliente, horario disponible
2. **Selección de Elementos:** Paquetes y productos
3. **Facturación:** Método de pago y finalización

Flujo general:

- Ingreso de datos
- Consulta de horarios disponibles
- Selección de horario
- Ingreso de información del cliente
- Agregado de productos o paquetes
- Selección del método de pago
- Envío de reservación

- Generación y visualización de la factura en PDF

Reservación en Línea

Las reservaciones en línea pueden ser creadas por el público general. Este proceso está enfocado en el registro de jugadores y consta de dos pasos:

1. **Recolección de Datos:** Similar a la reservación presencial
2. **Registro de Jugadores:** Hasta 4 jugadores por sesión

Flujo general:

- Ingreso de datos de la reservación
- Consulta y selección de horarios
- Ingreso de datos del cliente
- Registro de jugadores
- Envío de la reservación
- Creación de la sesión de juego
- Confirmación exitosa

Gestión de Reservaciones

Visualización

Los administradores pueden acceder a los detalles completos de cada reservación:

- ID, fecha y hora
- Cliente
- Estado de pago
- Estado de asistencia

Completar el Pago

Para reservaciones no pagadas, el administrador puede realizar el proceso de cobro:

- Visualización de la reservación
- Adición de productos o paquetes (si aplica)
- Selección del método de pago
- Envío del pago
- Generación de factura
- Visualización del PDF y confirmación

Control de Acceso

El sistema de reservaciones incluye control de acceso mediante middleware:

- Las rutas públicas permiten el acceso sin autenticación

- Las funciones administrativas requieren token válido
- Si el usuario no está autenticado, se redirige al inicio de sesión
- Se verifican permisos adecuados para operaciones sensibles

Puntos de Integración

El sistema se conecta con varios módulos clave para su funcionamiento:

Sistema	Finalidad de la Integración
Sistema de Horarios	Selección de horarios disponibles
Sistema de Paquetes	Asociación de paquetes a la reservación
Sistema de Productos	Adición de productos individuales
Sistema de Facturación	Generación automática de factura tras el pago
Sistema de Juegos	Creación de partidas con jugadores desde reservaciones

Sistema de Facturación

El **Sistema de Facturación** es responsable de la creación, gestión y procesamiento de pagos de facturas dentro de la aplicación. Permite registrar ventas tanto de bienes físicos como de servicios, admite múltiples métodos de pago y lleva el seguimiento de transacciones financieras. Este documento cubre los componentes principales, modelos de datos y flujos de trabajo clave del sistema.

Arquitectura del Sistema

El sistema está compuesto por varios componentes interconectados que permiten la funcionalidad completa de gestión de facturas:

- **Capa API (invoice.ts)**

- **Capa de interfaz de usuario**
 - Creación de facturas (facturacion/crear.vue)
 - Visualización y gestión de facturas (facturacion/index.vue)
- **Sistemas externos integrados:**
 - Gestión de productos
 - Gestión de paquetes
 - Gestión de bodegas
 - Sistema de cajas registradoras

Modelos de Datos

El sistema utiliza los siguientes modelos principales:

Invoice (Factura)

- id: string
- paymentMethod: InvoicePaymentMethod
- subtotal: number
- tax: number
- total: number
- clientDocument: string
- details: InvoiceDetail[]

InvoiceDetail (Detalle de Factura)

- id: string
- itemId: string
- itemName: string
- itemType: string
- quantity: number
- unitPrice: number
- total: number

CreateInvoice (Crear Factura)

- paymentMethod: InvoicePaymentMethod
- clientDocument: string
- details: CreateInvoiceDetail[]

CreateInvoiceDetail (Detalle al Crear Factura)

- itemId, itemName, itemType, quantity, unitPrice

Enumeraciones:

- **InvoiceItemType:** GOOD (producto físico), SERVICE (paquete)
- **InvoicePaymentMethod:** CASH, CARD, ONLINE

Proceso de Creación de Factura

El proceso incluye los siguientes pasos:

1. Acceder al formulario de creación
2. Seleccionar el método de pago
3. Seleccionar la bodega
4. Ingresar documento del cliente
5. Buscar productos o paquetes disponibles
6. Definir cantidad (productos) o usar cantidad fija (paquetes)
7. Calcular totales parciales y generales
8. Enviar la factura
9. Mostrar notificación y redirigir tras éxito

Funcionalidades Principales

Creación de Factura

El sistema permite crear nuevas facturas incluyendo:

- Documento del cliente (identificación)
- Método de pago (Efectivo, Tarjeta, En línea)
- Selección de bodega para gestión de inventario
- Ítems de la factura (productos o servicios)
 - **Productos:** cantidad ajustable
 - **Paquetes:** cantidad fija en 1
- Cálculo automático de totales por ítem y del total general

Gestión de Facturas

El sistema ofrece una vista en tabla con funcionalidades como:

- Visualización de todas las facturas
- Filtros por:
 - Documento del cliente
 - Método de pago
- Despliegue de información financiera:
 - Subtotal
 - Impuestos
 - Total
- Acceso a los detalles de cada factura individual

Procesamiento de Pagos

El sistema admite los siguientes métodos de pago:

- Efectivo

- Tarjeta
- En línea

Cada factura almacena el método de pago utilizado, el cual debe ser definido en el momento de su creación.

Referencia de API

Funciones principales disponibles:

Función	Descripción
createInvoice	Crea una factura usando la bodega del usuario actual
createInvoiceByWarehouseId	Crea una factura para una bodega específica
getInvoice	Recupera una factura por su ID
getInvoiceByClientDocument	Lista facturas de un cliente específico
getAllInvoices	Lista todas las facturas con filtros opcionales
getItemTypes	Obtiene los tipos de ítems disponibles (producto o servicio)
getPaymentMethods	Obtiene los métodos de pago disponibles

Integración con Otros Sistemas

El sistema se integra con varios módulos de la aplicación:

Sistema Integrado	Propósito de Integración
-------------------	--------------------------

Gestión de Productos	Facturación de productos físicos
Gestión de Paquetes	Facturación de servicios o combos
Gestión de Bodegas	Asocia facturas a bodegas específicas para control de inventario
Gestión de Stock	Actualización del inventario al facturar productos
Sistema de Cajas	Vinculación del pago con el sistema de cajas para trazabilidad

Componentes de la Interfaz

El sistema de facturación incluye dos interfaces principales:

- **Listado de Facturas:**
 - Tabla con todas las facturas
 - Filtros dinámicos
 - Botones para ver detalles
- **Creación de Factura:**
 - Formulario con campos para:
 - Documento del cliente
 - Método de pago
 - Selección de bodega
 - Selección y cantidad de ítems
 - Cálculo automático de totales

Sistema de Gestión de Paquetes

Propósito y Alcance

El **Sistema de Gestión de Paquetes** proporciona funcionalidades para crear, editar, visualizar y administrar paquetes de productos. Los paquetes son agrupaciones de productos con cantidades específicas que se venden a un precio unificado. Este sistema se integra con el sistema de gestión de productos y es utilizado por los módulos de *Reservaciones* y *Facturación* durante las operaciones de venta.

Modelo de Datos

Interfaces principales:

Package

- id: string
- name: string
- description: string
- price: number
- active: boolean
- packageDetail: PackageDetail[]

PackageDetail

- id: string
- product: Product
- quantity: number

CreatePackage

- name: string
- description: string
- price: number
- packageDetail: CreatePackageDetail[]
- active: boolean

Arquitectura del Sistema

Componentes principales:

- Funciones de API: createPackage, updatePackage, getPackageById, getAllPackages
- Interfaz de usuario:
 - Lista de paquetes
 - Creación de paquetes
 - Edición de paquetes
 - Vista de detalle

Se apoya en el sistema de productos para obtener los productos disponibles que se agregan a los paquetes.

Operaciones Clave

Listado y Filtrado de Paquetes

El sistema muestra todos los paquetes disponibles en una tabla con columnas para:

- Nombre
- Descripción
- Precio
- Estado (activo/inactivo)
- Acciones (Editar, Ver, Cambiar estado)

Desde esta vista, el usuario puede:

- Crear nuevos paquetes
- Editar paquetes existentes
- Consultar detalles
- Activar o desactivar paquetes

Creación de Paquetes

Flujo:

1. Navegar al formulario de creación
2. Consultar productos activos disponibles
3. Ingresar nombre, descripción y precio del paquete
4. Agregar productos con sus cantidades
5. Enviar formulario
6. Confirmar éxito y redirigir al listado

Edición de Paquetes

La edición carga los datos del paquete existente con `getPackageById`, permitiendo:

- Modificar nombre, descripción o precio
- Agregar, eliminar o modificar productos
- Guardar cambios mediante `updatePackage`

Gestión del Estado de Paquetes

Se puede activar o desactivar un paquete desde el listado. Esto determina su disponibilidad en otros módulos del sistema como reservaciones y facturación.

Flujo:

1. Clic en botón “Cambiar Estado”
2. Confirmación del usuario
3. Se actualiza el estado mediante `updatePackage`
4. Se refresca el listado

Componentes Clave

Vista de Lista

Muestra todos los paquetes en formato de tabla con resumen de cantidad total al final.
Incluye botones de acción para editar, ver y cambiar estado.

Formulario de Creación

Incluye:

- Información básica: nombre, descripción, precio
- Selector de productos con cantidad
- Tabla de productos agregados
- Validación del formulario

Vista de Detalle

Vista de solo lectura que muestra:

- Nombre, descripción, precio y estado
- Tabla con productos y sus cantidades
- Botones para editar o regresar al listado

Validación de Formularios

Las validaciones aseguran que:

- El nombre y la descripción sean obligatorios
- El precio sea mayor que 0
- Se agregue al menos un producto al paquete

Estas validaciones están implementadas con la librería **Zod**.

Referencia de API

Función	Propósito	Parámetros	Retorno
createPackage	Crea un nuevo paquete	data: CreatePackage	Package
updatePackage	Actualiza un paquete existente	id: string, data: CreatePackage	Package
getPackageById	Obtiene un paquete por su ID	id: string	Package

getAllPackages	Lista todos los paquetes	—	Package[]
----------------	--------------------------	---	------------

Todos los endpoints utilizan la ruta base /v1/packages.

Integración con Otros Sistemas

El sistema de paquetes se comunica con:

Sistema	Finalidad de la Integración
Gestión de Productos	Obtener productos disponibles para los paquetes
Sistema de Reservaciones	Incluir paquetes en el proceso de reservación
Sistema de Facturación	Agregar paquetes como ítems en facturas

Sistema de Juegos

El **Sistema de Juegos** gestiona el ciclo completo de las partidas de mini-golf, desde el registro de jugadores hasta la jugabilidad hoyo por hoyo y la presentación de resultados finales. Este documento describe los componentes principales, modelos de datos, flujo de juego, interfaces de usuario y puntos de integración que conforman este sistema.

Este módulo está enfocado exclusivamente en la gestión de la partida como tal. Para el proceso de creación de juegos mediante reservaciones, ver el *Sistema de Reservaciones*.

Componentes Clave y Modelos de Datos

El sistema está compuesto por múltiples entidades y estructuras que trabajan de manera conjunta para gestionar la experiencia de juego.

Entidades principales:

Entidad	Descripción	Propiedades clave
Game	Representa una sesión de juego de mini-golf	id, reservationId, players, hasFinished, currentHole
Player	Participante dentro del juego	id, name, playerNumber
ScoreHolePlayer	Puntuación por hoyo de un jugador	id, name, totalShots

Interfaces auxiliares:

- UpdateScoreGamePayload: Datos para actualizar puntuaciones en un hoyo específico
- ScorePlayerPayload: Información de puntuación de un jugador
- UpdatePlayersGame: Registro de jugadores en una partida
- PlayersGamePayload: Detalle de los jugadores a registrar

Ciclo de Vida del Juego

El flujo de un juego en el sistema sigue estas etapas:

1. Creación de juego desde una reservación
2. Registro de jugadores (hasta 4)
3. Inicio del juego (hoyo 1)
4. Registro de puntuaciones hoyo por hoyo (1 al 18)
5. Finalización del juego y presentación de resultados

Las puntuaciones se registran en tiempo real a través de la API updateScoreGame.

Componentes del Sistema

El sistema incluye componentes de interfaz y servicios API:

- **API Services:** getGameById, getScoreHole, getScoreGame, updateScoreGame, updatePlayerGame
- **Componentes de Interfaz:**
 - Página de listado de juegos

- Página de inicio del juego
- Página de juego por hoyo
- Página de resultados finales

Funcionalidades Principales

1. Listado de Juegos

Permite visualizar todas las partidas con indicadores de estado. Incluye:

- ID del juego
- Estado de finalización
- Hoyo actual (si está en curso)
- Acciones para jugar o visualizar

2. Registro de Jugadores

Antes de comenzar el juego, se deben registrar los participantes:

- Permite de 1 a 4 jugadores
- Se asignan nombres y número de jugador
- Se valida que todos los campos estén completos
- Los datos se envían a través de `updatePlayerGame`

3. Juego Hoyo por Hoyo

La partida se desarrolla de forma secuencial en 18 hoyos:

- Muestra puntuaciones actuales por jugador
- Permite ingresar tiros por hoyo
- Valida que se ingresen todas las puntuaciones
- Soporta navegación entre hoyos
- Acumula la puntuación total del jugador

4. Resultados Finales

Una vez finalizada la partida, se muestra un resumen:

- Ranking de jugadores por cantidad total de tiros (menor = mejor)
- Visualización destacada de los tres primeros lugares
- Íconos o insignias: oro, plata y bronce

Integración con la API

Endpoint	Método	Función	Propósito
/v1/games	GET	getAllGames	Obtener todas las partidas
/v1/games/:game_id	GET	getGameById	Obtener partida por ID
/v1/games/score/:game_id/:hole	GET	getScoreHole	Obtener puntuaciones de un hoyo
/v1/games/score/:game_id	GET	getScoreGame	Obtener puntuación total
/v1/games/score/:game_id	PATCH	updateScoreGame	Actualizar puntuación por hoyo
/v1/games/players/:game_id	PATCH	updatePlayerGame	Registrar jugadores en el juego

Flujos de Usuario Clave

Flujo de Creación y Registro de Jugadores

1. Selección del juego desde el listado
2. Registro de jugadores si no existen
3. Envío de datos
4. Confirmación

Flujo de Juego y Puntuación

1. Acceso al hoyo correspondiente
2. Ingreso de tiros por jugador
3. Validación
4. Paso al siguiente hoyo
5. Finalización automática al completar el hoyo 18

Detalles Técnicos de Implementación

Gestión del Estado

Se utiliza Vue con Composition API y **Pinia Query Cache** para manejar el estado:

- useCustomQuery: para obtener y cachear datos del juego
- useMutation: para mutaciones con manejo de éxito/error
- Invalidación de queries tras actualización para asegurar datos actualizados

Validaciones de Usuario

- Validación de nombre de jugadores
- Ingreso obligatorio de puntuaciones
- Mínimo de 1 tiro por jugador
- Verificación de puntuaciones completas antes de avanzar

Sistemas de Administración

Sistema de Gestión de Empleados

El **Sistema de Gestión de Empleados** proporciona funcionalidades para visualizar y administrar información del personal, asignar cajas registradoras y consultar el historial de cada empleado. Esta documentación cubre la interfaz para visualizar detalles individuales, gestionar asignaciones de cajas y revisar el historial laboral. Para información sobre la configuración de cajas registradoras, ver el *Sistema de Gestión de Cajas*.

Visión General

Este sistema forma parte del módulo administrativo y permite la gestión integral del registro de empleados, incluyendo:

- Información personal
- Detalles salariales
- Deducciones por beneficios
- Asignaciones operativas actuales
- Historial de cambios

Además, mantiene un registro cronológico de eventos significativos relacionados con el empleado, lo cual garantiza trazabilidad y soporte para auditorías.

Componentes del Sistema

Vistas y funcionalidades principales:

- Visualización de información del empleado
- Asignación y cambio de caja registradora
- Historial de cambios del empleado

Visualización de Información del Empleado

La interfaz muestra los siguientes datos:

- Nombre completo (nombres y apellidos)
- Tipo de empleado (con etiqueta visual)
- Salario actual
- Deducciones de beneficios:
 - IGSS (Instituto Guatemalteco de Seguridad Social)
 - IRTRA (Instituto de Recreación de los Trabajadores)
- Indicadores visuales muestran si cada deducción aplica al empleado

Asignación de Caja Registradora

Este módulo permite gestionar qué caja está asignada a cada empleado, función crítica para las operaciones del sistema.

Visualización de asignación actual:

- Caja asignada actualmente (si existe)
- Advertencia si no hay caja asignada

Gestión de asignación:

- Visualización de cajas disponibles (no asignadas)
- Selección de nueva caja y reasignación automática
- Proceso completo de:
 - Desasignar caja anterior
 - Asignar nueva caja
 - Actualizar datos y mostrar confirmación

Seguimiento del Historial del Empleado

Se muestra un historial en formato de tabla que incluye:

- Tipo de evento (con etiquetas visuales)
- Comentario o descripción del evento
- Fecha de ocurrencia

Manejo especial para cambios salariales:

- Aumentos y reducciones claramente diferenciados
- Nuevo monto salarial destacado en los comentarios

Implementación Técnica

Estructura del Componente

La funcionalidad está implementada como un componente Vue que interactúa con múltiples endpoints de API para obtener y actualizar información.

Flujos de Datos

- Uso de **Composition API** y **Pinia Query Cache**
- Mutaciones con useMutation para cambios como la asignación de cajas
- Invalida caché tras cada operación exitosa para mantener la información actualizada

Tipos de Datos Principales

Tipo de Dato	Descripción	Campos clave
Employee	Registro base del empleado	firstName, lastName, employeeType, salary, igssPercentage, irtraPercentage
CashRegister	Información de caja registradora	id, code, active, warehouseId, employeeId
EmployeeHistory	Evento registrado del historial laboral	historyType, commentary, historyDate
UpdateCashRegister	Estructura para actualizar caja	code, warehouseId, active, employeeId

Principales Interacciones API

Función	Descripción
---------	-------------

getEmployeeById	Obtiene los detalles completos del empleado, incluyendo historial
getCashRegisterByEmployeeId	Consulta la caja asignada al empleado
getAllCashRegisters	Lista todas las cajas disponibles para asignar
updateCashRegister	Actualiza los datos de una caja registradora
changeCashRegisterToEmployee	Cambia la caja asignada a un empleado

Integración con Otros Sistemas

Este sistema se comunica e integra con:

Sistema Relacionado	Funcionalidad Asociada
Gestión de Cajas	Asignación directa y validación de disponibilidad
Operaciones de Caja	Seguimiento de actividad según empleado asignado
Facturación	Relación entre caja, empleado y transacciones procesadas
Gestión de Bodegas	Asociación entre caja y bodega correspondiente

Sistema de Gestión de Cajas Registradoras

Propósito y Alcance

El **Sistema de Gestión de Cajas Registradoras** proporciona una interfaz técnica y funcional para crear, editar, visualizar y administrar cajas registradoras dentro de la aplicación. Incluye el modelo de datos, las interfaces de usuario para su gestión y las operaciones de API necesarias para mantenerlas.

Modelo de Datos

El sistema maneja los siguientes atributos clave:

- Cada caja está identificada por un **código único**
- Está **asociada a una bodega específica**
- Puede estar **activa o inactiva**
- Puede estar **asignada opcionalmente a un empleado**, mediante su ID

Interfaz de Gestión

El sistema incluye tres pantallas principales:

1. **Listado de Cajas**
Muestra todas las cajas en formato de tabla con filtros y acciones para editar o cambiar su estado.
2. **Crear Caja Registradora**
Formulario para agregar nuevas cajas con selección de bodega activa (mediante un diálogo emergente). Las cajas se crean activas por defecto.
3. **Editar Caja Registradora**
Permite modificar código y bodega asociada de una caja existente. La asignación de empleados se gestiona desde el módulo correspondiente.

Vista de Listado

La tabla de listado incluye:

- Código de la caja
- Bodega asociada
- Estado (activo/inactivo)
- Acciones (editar, cambiar estado)
- Filtros por código, nombre de bodega y estado
- Contador de total de cajas al pie de la tabla

Opciones de Filtro:

Campo	Descripción
-------	-------------

Código	Texto para filtrar por código de caja
Bodega	Texto para filtrar por nombre de bodega
Estado	Desplegable: Todas, Activas, Inactivas

Creación de Cajas

El formulario de creación incluye:

- **Código** (requerido): Identificador único
- **Bodega** (requerido): Seleccionada desde un diálogo de bodegas activas
- **Empleado (opcional)**: Puede dejarse vacío para cajas no asignadas

Edición de Cajas

El formulario de edición:

- Carga datos existentes de la caja seleccionada
- Permite modificar el código y la bodega asociada
- No incluye la asignación directa de empleado (esto se hace desde el sistema de empleados)

Operaciones de API

Función	Descripción	Método o HTTP	Endpoint
getAllCashRegisters	Obtiene todas las cajas con filtros opcionales	GET	/v1/cash-registers/all

getCashRegister	Obtiene una caja específica por ID	GET	/v1/cash-registers/{id}
createCashRegister	Crea una nueva caja registradora	POST	/v1/cash-registers
updateCashRegister	Actualiza una caja existente	PUT	/v1/cash-registers/{id}
toggleCashRegister	Cambia el estado activo/inactivo de una caja	PATCH	/v1/cash-registers/{id}/toggle
getCashRegisterByEmployeeId	Consulta la caja asignada a un empleado	GET	/v1/cash-registers/employee/{employeeId}
changeCashRegisterToEmployee	Asigna una caja a un empleado	PATCH	/v1/cash-registers/{id}/employee/{employeeId}

Integración con Otros Sistemas

Integración con Bodegas

Cada caja debe estar vinculada a una bodega activa. Durante la creación o edición, el usuario selecciona una bodega disponible mediante un diálogo emergente. Esto garantiza una correcta organización física de las cajas.

Integración con Empleados

Las cajas pueden ser asignadas a empleados específicos. Esta relación se gestiona principalmente desde el módulo de empleados, aunque puede definirse inicialmente al momento de crear la caja.

- `getCashRegisterByEmployeeId`: Consulta de caja asignada a un empleado
- `changeCashRegisterToEmployee`: Asignación o reasignación de caja a empleado

Manejo de Errores

El sistema incluye manejo de errores en varias operaciones:

- **Validación de campos requeridos:** código y bodega son obligatorios
- **Errores de API:** se notifican con mensajes claros para el usuario
- **Diálogos de confirmación:** al cambiar el estado de una caja, se solicita confirmación previa

Los errores se presentan mediante notificaciones tipo *toast* para mantener informada a la persona usuaria.

Patrones de Uso

El uso típico del sistema incluye:

- Crear cajas para cada punto de venta físico
- Asignar empleados a cajas específicas
- Activar o desactivar cajas según necesidad operativa
- Actualizar datos cuando cambien la bodega o el código

Este sistema permite un control eficiente y trazable de las operaciones de punto de venta en múltiples ubicaciones.

Sistema de Gestión de Horarios

El **Sistema de Gestión de Horarios** permite crear, visualizar, actualizar y eliminar bloques de tiempo que se utilizan en el sistema de reservaciones del centro de entretenimiento. Su función principal es definir franjas horarias disponibles que luego pueden ser asignadas a reservaciones. Este módulo está enfocado exclusivamente en la administración de horarios. Para la asignación de estos a reservaciones específicas, consultar el *Sistema de Reservaciones*.

Visión General del Sistema

Este sistema, aunque ligero en complejidad, es esencial para el control de disponibilidad en las operaciones del negocio. Ofrece una interfaz sencilla para definir horarios de inicio y fin que luego serán seleccionados en el proceso de reservación.

Modelo de Datos

El modelo de datos del sistema se basa en la entidad Schedule, la cual representa una franja horaria. Sus atributos principales son:

- id: Identificador único
- startTime: Hora de inicio (formato HH:MM)
- endTime: Hora de finalización (formato HH:MM)

Interfaces adicionales:

- CreateSchedule: utilizada para la creación de nuevos horarios
- UpdateSchedule: utilizada para la edición de horarios existentes

Ambas contienen los mismos campos (startTime y endTime), pero se definen por separado para mantener la seguridad de tipos y la separación de responsabilidades.

Interfaz de API

El sistema expone una API REST a través del endpoint /v1/schedules con las siguientes operaciones:

Operación	Descripción	Método HTTP	Endpoint
Obtener por ID	Recupera un horario específico	GET	/v1/schedules/{id}
Obtener todos	Lista todos los horarios existentes	GET	/v1/schedules
Crear	Crea un nuevo horario	POST	/v1/schedules
Actualizar	Modifica un horario existente	PUT	/v1/schedules/{id}
Eliminar	Elimina un horario existente	DELETE	/v1/schedules/{id}

Interfaz de Usuario

El sistema ofrece tres páginas principales para la gestión de horarios:

Listado de Horarios

La vista principal muestra todos los horarios disponibles en formato de tabla. Permite:

- Visualizar horarios existentes con horas de inicio y fin
- Navegar a la creación de nuevos horarios
- Editar o eliminar horarios existentes
- Confirmar la eliminación mediante un cuadro de diálogo

Creación de Horarios

El formulario de creación permite:

- Ingresar la hora de inicio (obligatorio)
- Ingresar la hora de finalización (obligatorio)
- Validación del formulario para asegurar campos completos
- Redirección al listado tras la creación exitosa

Edición de Horarios

El formulario de edición:

- Muestra los datos actuales del horario seleccionado
- Permite modificar hora de inicio y fin
- Aplica las mismas reglas de validación que en la creación
- Muestra notificación de éxito al finalizar

Validación de Formularios

Tanto el formulario de creación como el de edición utilizan la librería **Zod** para validación:

- La hora de inicio es obligatoria y no puede estar vacía
- La hora de finalización también es obligatoria
- Se utiliza zodResolver de PrimeVue para aplicar la validación en los formularios

Componentes de Interfaz

Página	Componentes	Funcionalidad principal
/horarios/	DataTable, Button, ConfirmDialog	Lista de horarios con opciones de editar y eliminar

/horarios/crear Form, InputTime

Formulario de creación de nuevos horarios

/horarios/[id] Form, InputTime

Edición de horarios existentes

Integración con Otros Sistemas

Este sistema se integra principalmente con el *Sistema de Reservaciones*. Los horarios definidos aquí se utilizan como opciones al momento de crear nuevas reservaciones, garantizando que solo se asignen franjas horarias válidas.

Aunque la integración no se muestra directamente en el código fuente de este módulo, está diseñada para exponer horarios disponibles para su uso en componentes de reservación y otras operaciones dependientes del tiempo.

Implementación Técnica

Tecnologías y herramientas utilizadas:

- **Vue.js** para la construcción de componentes frontend
- **PrimeVue** para elementos de interfaz como tablas, formularios y botones
- **Zod** para validación de formularios
- API personalizada para la comunicación con el backend
- Patrón estándar **CRUD** con separación clara entre modelos, lógica API e interfaz de usuario

Sistema de Gestión de Bodegas

El **Sistema de Gestión de Bodegas** permite a los administradores crear, visualizar, editar y gestionar el estado de las bodegas dentro de la aplicación. Estas bodegas representan ubicaciones físicas de almacenamiento y son componentes esenciales en el proceso de gestión de inventario. Para información relacionada con el inventario de productos, consultar el *Sistema de Gestión de Stock*, y para la gestión de relaciones con proveedores, ver el *Sistema de Gestión de Proveedores*.

Visión General del Sistema

Este sistema proporciona una interfaz completa para la administración de entidades de tipo bodega. Cada bodega cuenta con atributos como nombre, ubicación y estado (activo o inactivo). Entre las funcionalidades disponibles se incluyen:

- Visualización de todas las bodegas con capacidad de filtrado

- Creación de nuevas bodegas
- Edición de información existente
- Activación o desactivación de bodegas

Modelo de Datos

Entidad Principal:

Warehouse

Campo	Tipo	Descripción
id	string	Identificador único de la bodega
name	string	Nombre de la bodega
ubication	string	Ubicación física de la bodega
active	boolean	Indica si la bodega está activa o inactiva

Interfaces relacionadas:

- **CreateWarehouse:** Utilizada al crear una nueva bodega
- **UpdateWarehouse:** Utilizada al actualizar una bodega existente
- **SpecWarehouse:** Utilizada para filtrar bodegas en la vista de listado

Flujos Principales

Listado y Filtrado de Bodegas

La página principal muestra todas las bodegas registradas en una tabla, con opciones de filtro por:

- **Nombre**
- **Ubicación**
- **Estado activo/inactivo**

Desde esta vista también se pueden ejecutar acciones como editar o cambiar el estado de una bodega.

Creación de Bodegas

El flujo de creación permite a los administradores registrar nuevas bodegas:

1. Acceder a la página de creación
2. Llenar el formulario con nombre y ubicación
3. Validar campos mediante Zod
4. Enviar el formulario
5. Recibir confirmación de éxito y redirigir al listado

Las nuevas bodegas se crean activas por defecto. Las validaciones son:

- Nombre: requerido, máximo 100 caracteres
- Ubicación: requerida, máximo 255 caracteres

Edición de Bodegas

El formulario de edición permite modificar una bodega existente. El flujo es similar al de creación:

1. Acceso mediante ID
2. Carga de datos actuales
3. Edición de nombre y ubicación
4. Validación del formulario
5. Envío y confirmación

Cambio de Estado (Activar/Inactivar)

Desde el listado, se puede cambiar el estado activo de una bodega:

1. El usuario hace clic en el botón “Cambiar Estado”
2. Se muestra un diálogo de confirmación
3. Al confirmar, se ejecuta la acción mediante API
4. Se actualiza la vista y se muestra notificación

Referencia de API

Función	Método HTTP	Endpoint	Descripción

getAllWarehouses	GET	/v1/warehouses/all	Lista todas las bodegas con opción de filtro
getWarehouseById	GET	/v1/warehouses/:id	Obtiene una bodega por su ID
createWarehouse	POST	/v1/warehouses	Crea una nueva bodega
updateWarehouse	PUT	/v1/warehouses/:id	Actualiza los datos de una bodega existente
toggleWarehouse	PATCH	/v1/warehouses/:id/toggle	Cambia el estado activo/inactivo de una bodega

Componentes de Interfaz

El sistema incluye tres páginas principales:

- Listado de Bodegas (bodegas/index.vue)**
 - Tabla de bodegas con columnas: nombre, ubicación, estado
 - Filtros para cada campo
 - Acciones para editar y cambiar estado
 - Diálogo de confirmación para cambios de estado
- Creación de Bodega (bodegas/crear.vue)**
 - Formulario con validación para nombre y ubicación
 - Notificaciones de éxito o error
- Edición de Bodega (bodegas/[id].vue)**
 - Formulario precargado con datos actuales
 - Validación idéntica al formulario de creación
 - Notificaciones y redirección tras actualización

Los componentes utilizan **PrimeVue** para elementos como DataTable, Button, InputText, Dropdown, Tag y ConfirmDialog.

Integración con Otros Sistemas

Las entidades de tipo bodega creadas en este sistema se utilizan en varios otros módulos clave:

Sistema Relacionado	Funcionalidad Asociada
Gestión de Stock	Asociación de productos con su ubicación física
Gestión de Productos	Referencia cruzada entre productos y bodegas
Entradas de Productos	Registro de productos recibidos en una bodega específica
Gestión de Proveedores	Asociación de entregas con bodegas receptoras

Sistema de Gestión de Proveedores

El **Sistema de Gestión de Proveedores** permite a los administradores crear, visualizar, editar y gestionar la información de los proveedores dentro de la aplicación. Este sistema es esencial para el seguimiento y mantenimiento de las relaciones comerciales que respaldan las operaciones de inventario y bodega. Incluye funcionalidades para registrar nuevos proveedores, modificar sus datos y activar o desactivar su uso dentro del sistema según sea necesario.

Modelo de Datos

Entidad Principal:

Supplier

Campo	Descripción	Tipo	Restricciones
id	Identificador único	string	Autogenerado

nit	Número de identificación tributaria	string	Requerido, máximo 50 caracteres
name	Nombre del proveedor	string	Requerido, máximo 100 caracteres
address	Dirección física	string	Requerido, máximo 255 caracteres
taxRegime	Régimen tributario	number	0.05 (Pequeño contribuyente), 0.12 (General)
active	Estado del proveedor	boolean	true = activo, false = inactivo

Interfaz de Gestión

El sistema se compone de tres páginas principales:

- Listado de Proveedores**
 - Tabla con columnas: nombre, dirección y estado
 - Filtros por nombre, dirección y estado (activo/inactivo)
 - Acciones disponibles: editar proveedor, cambiar estado
 - Contador total de proveedores en el pie de la tabla
- Creación de Proveedor**
 - Formulario con campos: NIT, nombre, dirección, régimen tributario
 - Validaciones: todos los campos son obligatorios
 - Al enviar con éxito, se registra como proveedor activo y se muestra notificación
- Edición de Proveedor**
 - Mismo formulario que el de creación, pero precargado con datos actuales
 - Permite modificar toda la información excepto el ID
 - Al guardar, se actualizan los datos y se redirige al listado con notificación

Flujos de Usuario

Listado y Filtrado

La página principal permite:

- Buscar por nombre o dirección con entrada de texto
- Filtrar por estado (todos, activos, inactivos)
- Aplicar filtros en tiempo real
- Acceder a formularios de edición o alternar el estado del proveedor

Creación de Proveedores

1. Acceder al formulario de creación
2. Ingresar datos válidos en todos los campos
3. Seleccionar régimen tributario (5% o 12%)
4. Enviar formulario
5. Recibir notificación y redirección al listado

Edición de Proveedores

1. Ingresar a la página de edición desde el listado
2. Ver formulario con datos actuales
3. Modificar campos deseados
4. Enviar cambios
5. Recibir confirmación y volver al listado

Activación/Desactivación

- En el listado, cada proveedor tiene un botón “Cambiar Estado”
- Al hacer clic, se solicita confirmación
- Si se confirma, se invierte el estado activo/inactivo
- Se conserva el historial del proveedor, pero no se permite su uso en nuevas operaciones si está inactivo

Validaciones de Formulario

- **NIT:** obligatorio, máximo 50 caracteres
- **Nombre:** obligatorio, máximo 100 caracteres
- **Dirección:** obligatoria, máximo 255 caracteres
- **Régimen tributario:** debe ser exactamente 0.05 o 0.12

API del Sistema

Función	Propósito	Parámetros	Retorno
getAllSuppliers	Obtener proveedores con filtros opcionales	specSupplier: SpecSupplier	Promise<Supplier[]>

getSupplierById	Obtener un proveedor por ID	id: string	Promise<Supplier>
createSupplier	Crear un nuevo proveedor	payload: CreateSupplier	Promise<Supplier>
updateSupplier	Editar un proveedor existente	id: string, payload: UpdateSupplier	Promise<Supplier>
toggleSupplier	Alternar estado activo/inactivo	id: string	Promise<void>

Componentes de Interfaz

- **Listado (/proveedores/index.vue):**
 - Tabla con filtros
 - Botones de acción para editar o cambiar estado
 - Diálogo de confirmación para cambios de estado
- **Formulario de creación (/proveedores/crear.vue):**
 - Campos para NIT, nombre, dirección y régimen
 - Validación con Zod
 - Notificaciones visuales
- **Formulario de edición (/proveedores/[id].vue):**
 - Precarga de datos del proveedor seleccionado
 - Mismas validaciones que el formulario de creación

Integración con Otros Sistemas

El sistema de proveedores se relaciona con:

Sistema Relacionado	Integración
Gestión de Bodegas	Asociación de proveedores con bodegas para recepción de productos

Entradas de Productos	Registro del proveedor que entrega los productos
------------------------------	--

Cuando un proveedor es desactivado, no se permite su uso en nuevas operaciones, pero se conserva su historial para trazabilidad.

Sistema de Gestión de Productos

El **Sistema de Gestión de Productos** permite a los usuarios crear, visualizar, editar y eliminar registros de productos dentro de la aplicación. Este módulo es una pieza fundamental para el control de inventario, ventas y operaciones de bodega, ya que centraliza toda la información relacionada con productos físicos y digitales, incluyendo atributos, clasificaciones y precios.

Modelo de Datos

Entidad Principal:

Product

Campo	Descripción
id	Identificador único del producto
name	Nombre del producto
code	Código interno
barCode	Código de barras legible por máquina
price	Precio del producto
type	Referencia al tipo de producto (clave foránea)

state	Referencia al estado del producto (clave foránea)
createdAt	Fecha de creación

Los **tipos** y **estados** de producto se administran como entidades independientes para clasificar y organizar el catálogo de productos.

Flujos de Gestión

Las operaciones principales admitidas por este sistema son:

- Visualización del listado de productos
- Filtrado de productos por múltiples criterios
- Creación de nuevos productos
- Edición de productos existentes
- Eliminación de productos

Cada operación se realiza mediante formularios e interfaces específicas conectadas a la API correspondiente.

Componentes de Interfaz

Página de Listado

La interfaz principal muestra una tabla con todos los productos registrados. Ofrece:

- Paginación y carga dinámica
- Filtros por nombre, código, código de barras, tipo y estado
- Acceso directo a formularios de creación y edición
- Visualización del número total de productos registrados

Formularios de Creación y Edición

Ambos formularios comparten la misma estructura de campos:

Campo	Reglas de Validación	Tipo de Componente
-------	----------------------	--------------------

Nombre	Obligatorio, máximo 100 caracteres	Campo de texto
Código	Obligatorio, máximo 50 caracteres	Campo de texto
Código de barras	Obligatorio, máximo 50 caracteres	Campo de texto
Precio	Obligatorio, mínimo 1	Entrada numérica
Tipo	Selección obligatoria	Dropdown
Estado	Selección obligatoria	Dropdown

Todos los campos son validados antes del envío para asegurar la integridad de los datos.

Integración con la API

La gestión de productos se realiza mediante un cliente API REST, el cual encapsula las operaciones de comunicación con el backend.

Endpoints Disponibles:

Función	Método HTTP	Endpoint	Propósito
createProduct	POST	/v1/products	Crear un nuevo producto
getAllProducts	GET	/v1/products/all	Obtener productos con filtros
updateProduct	PUT	/v1/products/{id}	Actualizar información de un producto

deleteProduct	DELETE	/v1/products/{id}	Eliminar un producto
getProduct	GET	/v1/products/{id}	Obtener producto por ID
getStates	GET	/v1/products/statuses	Obtener lista de estados disponibles
getTypes	GET	/v1/products/types	Obtener lista de tipos disponibles
getProductsByIds	POST	/v1/products/ids	Obtener productos por múltiples IDs

Integración con Otros Sistemas

Este sistema representa el **catálogo central de productos**, utilizado por los siguientes módulos:

Módulo Relacionado	Integración
Sistema de Stock	Seguimiento de existencias por producto y bodega
Sistema de Entradas	Registro de ingresos de productos en bodegas
Sistema de Facturación	Selección de productos al generar facturas
Sistema de Paquetes	Inclusión de productos dentro de paquetes de servicios

Gracias a esta integración, se garantiza la coherencia de la información de productos en toda la aplicación.

Sistema de Gestión de Entradas de Productos

El **Sistema de Gestión de Entradas de Productos** se encarga del registro y documentación de productos recibidos en las bodegas. Este módulo permite a los usuarios registrar las entregas provenientes de proveedores, mantener un historial de inventario entrante y detallar cada producto recibido con su cantidad, precio unitario y proveedor correspondiente.

Visión General del Sistema

El sistema incluye tres interfaces principales:

1. **Listado de Entradas**
Muestra todas las entradas registradas con filtros por número de factura, proveedor, bodega y fecha.
2. **Creación de Entrada**
Permite ingresar nueva información de productos recibidos, incluyendo productos múltiples.
3. **Visualización de Entrada**
Muestra los detalles completos de una entrada específica, incluyendo proveedor, bodega y productos recibidos.

Componentes y Funcionalidades Clave

Listado de Entradas

- Tabla filtrable por número de factura, bodega, proveedor y fecha
- Visualización del total de entradas registradas
- Acceso a detalles y opción para crear nueva entrada
- Actualización dinámica al aplicar filtros

Creación de Entrada de Productos

Flujo de creación:

1. Ingresar información general:
 - Número de factura
 - Fecha
 - Bodega receptora
 - Proveedor
2. Agregar productos recibidos:
 - Selección de producto
 - Cantidad

- Precio unitario
- 3. Visualizar cálculo automático del subtotal y total
- 4. Confirmar la operación mediante un diálogo
- 5. Guardar la entrada y redirigir al listado

Visualización de Entrada

Muestra los datos registrados de una entrada específica:

- Número de factura y fecha
- Proveedor (nombre, NIT, dirección)
- Bodega (nombre, ubicación)
- Tabla de productos con cantidad, precio unitario y subtotal
- Total general de la entrada

Modelo de Datos

Entidades Principales

Entidad	Campos Principales
ProductEntry	id, invoiceNumber, date, warehouse, supplier, details[]
ProductEntryDetail	id, productId, quantity, unitPrice
CreateProductEntry	invoiceNumber, date, warehouseId, supplierId, details[]
CreateProductEntryDetail	productId, quantity, unitPrice
SpecProductEntry	Filtros opcionales: id, invoiceNumber, warehouseId, supplierId, date

Entidades Relacionadas

Entidad	Campos Clave	Uso
---------	--------------	-----

Bodega	id, name, ublication	Determina la bodega receptora
Proveed r	id, nit, name, address	Identifica al proveedor de los productos
Producto	id, name, code, price	Identifica los productos recibidos

Integración con la API

Endpoints

Función	Métod o HTTP	Endpoint	Descripción
getAllProductEntries	GET	/v1/product-entries/all	Obtiene todas las entradas con filtros opcionales
getProductEntry	GET	/v1/product-entries/{id}	Obtiene una entrada por ID
getProductEntryByInvoiceNumber	GET	/v1/product-entries/invoice-number/{invoice}	Obtiene una entrada por número de factura
createProductEntry	POST	/v1/product-entries	Crea una nueva entrada de productos

Flujo de Trabajo: Crear una Entrada

1. Iniciar formulario
2. Cargar listas de opciones para:
 - Bodegas disponibles

- Proveedores activos
- Productos en inventario
- 3. Ingresar:
 - Número de factura
 - Fecha
 - Proveedor
 - Bodega
- 4. Añadir detalles de productos (cantidad, precio)
- 5. Confirmar y guardar
- 6. Redirigir al listado con mensaje de éxito

Caso de Uso Típico

Cuando una bodega recibe un lote de productos de un proveedor:

- El usuario accede al listado de entradas
- Inicia una nueva entrada
- Registra la factura, la fecha, el proveedor y la bodega
- Agrega productos con sus respectivas cantidades y precios
- Guarda y confirma la operación
- El sistema actualiza el inventario en base a los productos registrados

Este proceso garantiza trazabilidad de los ingresos y control del inventario.

Sistema de Configuración del Sistema

Propósito y Alcance

El **Sistema de Configuración del Sistema** permite a los administradores gestionar información empresarial esencial utilizada en toda la aplicación. Este módulo proporciona la funcionalidad para visualizar y actualizar datos clave como el nombre de la empresa, el número de identificación tributaria (NIT) y el régimen fiscal. Estos parámetros son fundamentales para el funcionamiento correcto de los procesos del negocio y aparecen en diversos documentos oficiales del sistema, como facturas, reportes y encabezados.

Interfaz de Configuración

La interfaz de configuración se accede desde el panel de administración a través del menú "Configuración". Está organizada en secciones, cada una correspondiente a un parámetro configurable de la empresa:

- Nombre de la empresa
- Número de identificación tributaria (NIT)
- Régimen fiscal

Cada sección incluye un campo de entrada, un botón para guardar cambios y una notificación de confirmación en caso de éxito.

Configuración del Nombre de la Empresa

Esta sección permite:

- Ver el nombre actual de la empresa
- Editar el nombre mediante un campo de texto
- Guardar los cambios con el botón **“Cambiar nombre”**
- Recibir una notificación de éxito al actualizar el valor

Configuración del NIT

Esta sección permite:

- Ver el NIT actual de la empresa
- Editar el NIT mediante un campo de texto
- Guardar los cambios con el botón **“Cambiar NIT”**
- Este valor es crítico y se utiliza en facturas y reportes legales

Configuración del Régimen Fiscal

Esta sección permite:

- Ver el régimen fiscal actual
- Seleccionar un nuevo régimen desde un menú desplegable
- Guardar los cambios con el botón **“Cambiar Régimen”**

Opciones disponibles:

- **General** – 12%
- **Pequeño contribuyente** – 5%

Este valor afecta directamente los cálculos de impuestos en el sistema de facturación.

Modelo de Datos y API

Estructura del Parámetro de Configuración

Los valores de configuración se almacenan como pares clave-valor con nombres descriptivos. Cada parámetro tiene los siguientes campos:

Campo	Descripción
-------	-------------

parameterKey	Identificador único del parámetro
value	Valor actual de configuración
name	Nombre legible para el usuario

Funciones API de Consulta

Función	Propósito
getNit()	Obtiene el número de identificación tributaria
getName()	Obtiene el nombre actual de la empresa
getRegime()	Obtiene el régimen fiscal configurado

Cada función retorna un objeto ParameterResponse con la información solicitada.

Funciones API de Actualización

Función	Propósito	Parámetro esperado
updateNIT(data)	Actualiza el número de identificación tributaria	{ nit: string }
updateName(data)	Actualiza el nombre de la empresa	{ newName: string }

updateRegime(data)	Actualiza el régimen fiscal	{ newRegime: string }
--------------------	-----------------------------	-----------------------

Flujo de Actualización de Configuraciones

1. El usuario visualiza el valor actual mediante una solicitud GET /v1/config/{parameter}
2. Edita el valor deseado en la interfaz
3. Hace clic en el botón correspondiente
4. Se envía una solicitud PATCH a la API
5. Se muestra una notificación de éxito o error

Este flujo se aplica de manera uniforme para todos los parámetros configurables.

Implementación Técnica

- Uso de Vue con API Composition
- Manejo de estado reactivo mediante watch
- Los cambios se gestionan mediante useMutation
- Las actualizaciones se comunican con el backend mediante solicitudes PATCH
- Se muestran notificaciones tipo “toast” para informar del estado de la operación

Integración con Otros Sistemas

Los valores configurados se utilizan en diversos módulos del sistema:

Parámetro	Módulos que lo utilizan	Uso principal
Nombre de empresa	Facturación, reportes, dashboard	Encabezados, pies de página y documentos oficiales
NIT	Facturación y reportes fiscales	Identificación tributaria en documentos legales
Régimen fiscal	Sistema de Facturación	Cálculo de impuestos según la normativa vigente

Estos valores son esenciales para el cumplimiento legal y la operación correcta del sistema, por lo tanto, su correcta configuración es crítica para la empresa.

