

MAYO 2025

Manual de TÉCNICO

REALIZADO POR:

Luisa Fernanda Hernandez Hernandez 368068

Ingrid Yuliana Perez Rodriguez 367760

ÍNDICE

01

Sistema/Aplicación

02

Especificación de requerimientos

03

Arquitectura del Sistema/Aplicación

04

Base de datos

05

Lógica/Reglas del Negocio

06

Descripción Interfaz de la aplicación

07

Descripción de reglas de seguridad

08

Conclusión

09

Anexos

10

Bibliografía/Referencias

SISTEMA/APLICACIÓN

Nombre del sistema

WormBook

Descripción y delimitación del sistema

El sistema WormBook esta diseñado para que los clientes de una cadena de librerías puedan comprar libros en línea y revisar la cantidad en existencia en diversas sucursales, también permite a los administradores del sistema ver los datos de los pedidos, envíos y ventas.

Objetivo general

El objetivo general del sistema es mejorar la experiencia de los clientes al permitirles conocer los productos disponibles en inventario y decidir si adquirirlos en la tienda física o mediante compra en línea.

Descripción de tipos de usuarios

El sistema está dirigido a dos tipos de usuarios:

1. Los clientes, quienes pueden realizar compras y consultar disponibilidad
2. Los empleados de la librería, encargados de gestionar envíos y registrar la información de ventas

Entorno operativo del sistema

Hosting/Servidores

- Opciones:
 - Cloud: AWS (EC2, S3), Google Cloud, Azure
 - Hosting compartido: SiteGround, Bluehost (para proyectos pequeños)
 - Serverless: Vercel, Netlify (para frontends estáticos)
- Especificaciones mínimas:
 - CPU: 2 núcleos (mínimo)
 - RAM: 4 GB
 - Almacenamiento: 50 GB SSD (más espacio si hay muchos libros con imágenes)

b) Base de Datos

- Oracle

c) Servicios Externos

- Pagos: Stripe, PayPal
- Envíos: FedEx

3. Entorno de Seguridad

- Certificados: HTTPS (SSL/TLS con Let's Encrypt)
- WAF (Web Application Firewall)
- Autenticación: OAuth 2.0 (para login con redes sociales)

4. Entorno del Usuario Final

- Navegadores compatibles:
 - Chrome, Firefox, Safari, Edge (últimas 2 versiones)
- Dispositivos:
 - PC
- Requisitos mínimos:
 - Conexión a Internet: 5 Mbps
 - JavaScript habilitado

ESPECIFICACIÓN DE REQUERIMIENTOS

Requerimientos funcionales

RF-01: El administrador debe tener acceso a un registro con los datos de los envíos.

RF-02: El administrador puede ver un registro de las ventas.

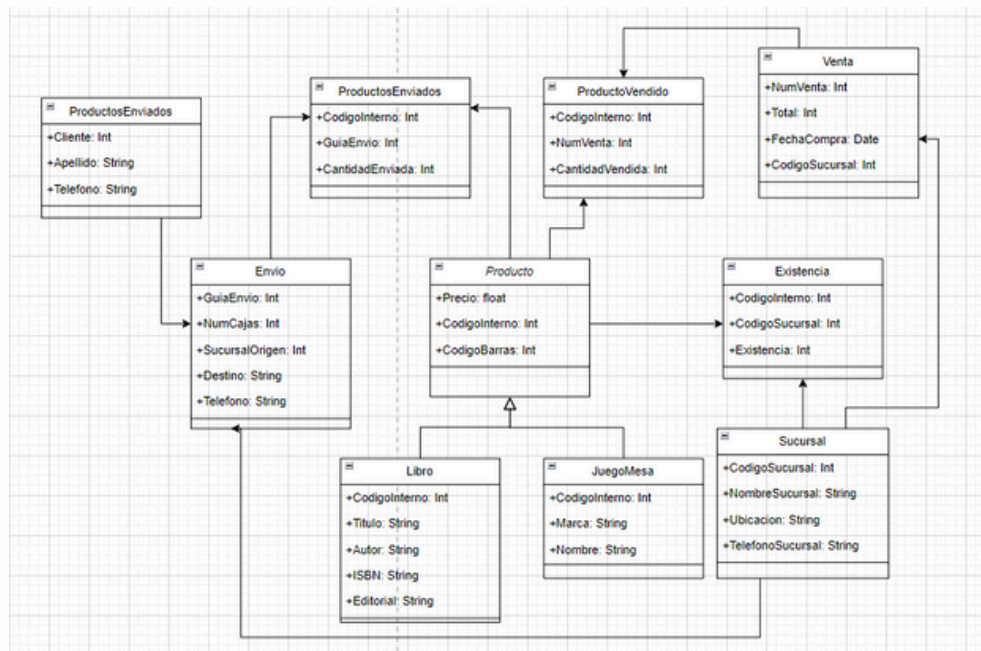
RF-03: El administrador puede editar la existencia de productos.

RF-04: El cliente puede realizar compras mediante la aplicación.

RF-05: El cliente debe poder ver en qué sucursales se encuentran ciertos productos y cuántos hay en existencia.

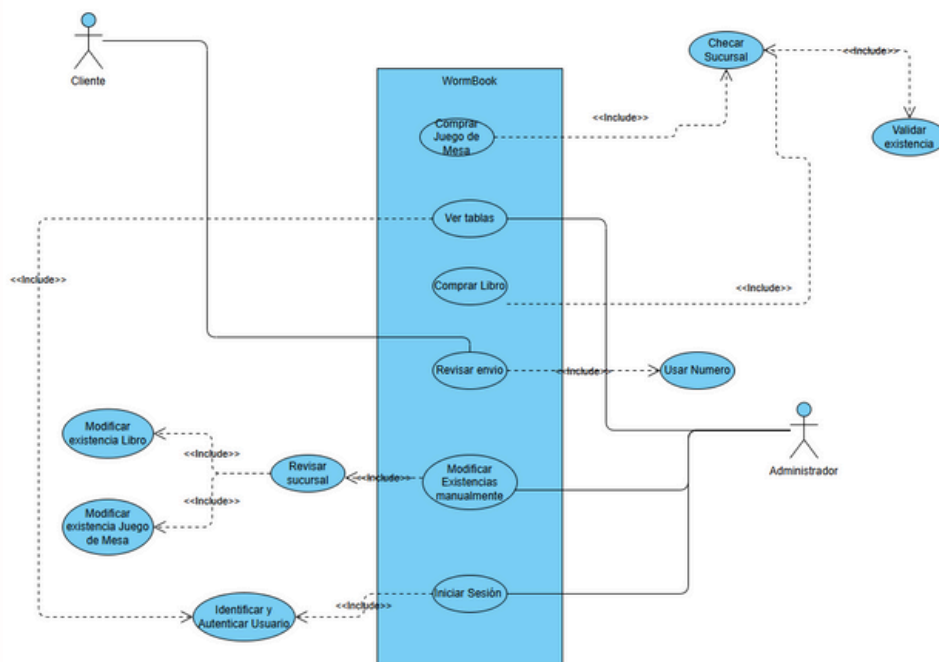
Modelado del sistema

a) Diagrama de clases



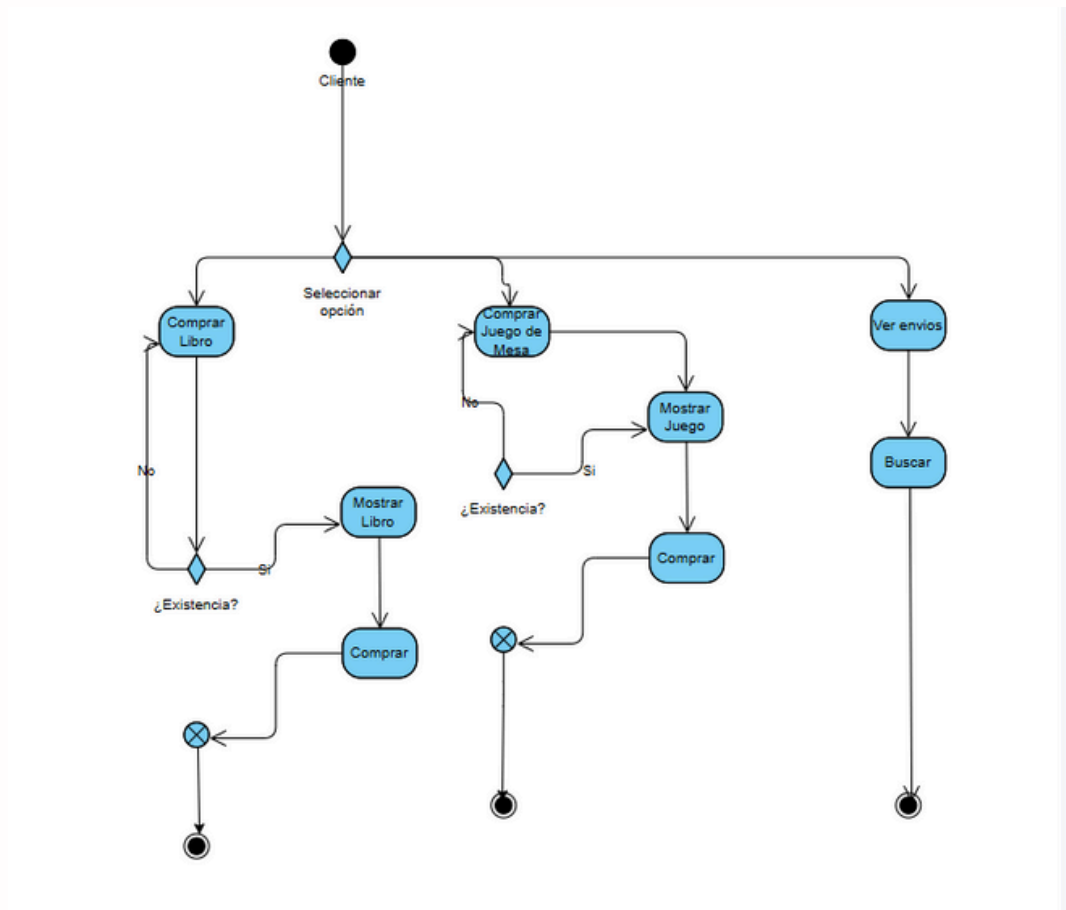
Modelado del sistema

b) Diagrama de casos de uso



Modelado del sistema

b) Diagrama de actividades



Requerimientos no funcionales

RNF-01: Base de datos Oracle.

RNF-02: Base de datos normalizada hasta la forma BCNF.

RNF-03: Compatibilidad con el navegador Chrome.

ARQUITECTURA DEL SISTEMA/APLICACIÓN

Layer/Tier's

- El sistema implementa una arquitectura en capas basada en el patrón MVC (Modelo-Vista-Controlador), utilizando ASP.NET MVC y Entity Framework como tecnología ORM. Esta separación permite una estructura clara, mantenible y escalable.
- Las capas del sistema son:
- Capa de Presentación (View):
- Esta capa contiene las vistas construidas con Razor y HTML, las cuales se encargan de mostrar información al usuario y recibir su interacción. Es el punto de contacto visual entre el sistema y el usuario final, se incluye js y solicitudes ajax para llegar a los controladores.
- Capa de Controladores (Controller):
- Gestiona la lógica de flujo de la aplicación. Recibe solicitudes del usuario, procesa la información a través de los modelos y determina la vista que se debe retornar. Actúa como intermediario entre la vista y los modelos.
- Capa de Modelo (Model):
- En esta capa se encuentra la lógica de negocio y las clases que representan las entidades del sistema. Entity Framework permite mapear estas clases a la base de datos.

Frontback/Backend

- Frontend:
- Corresponde a la interfaz gráfica con la que el usuario interactúa. Se desarrolla utilizando Razor Pages, HTML5, CSS3 y JavaScript. Esta capa se encarga de presentar los datos y recoger la entrada del usuario.
- Backend:
- Comprende la lógica de negocio, los controladores, los modelos y el acceso a base de datos. Está construido en ASP.NET MVC con Entity Framework, funcionando en el servidor. Se encarga de procesar solicitudes, validar datos, aplicar reglas del negocio y gestionar la persistencia en la base de datos.

Estructura modular del sistema

El sistema está dividido en los siguientes módulos funcionales, organizados bajo el patrón MVC. Cada módulo contiene sus propias vistas, controladores y modelos, lo que facilita su mantenimiento y escalabilidad.

1. Módulo de Autenticación (Login):

Permite el acceso al sistema mediante un formulario de inicio de sesión.

- Los usuarios con rol de administrador (empleados de la librería) pueden autenticarse.
- Una vez autenticados, los administradores tienen la posibilidad de registrar nuevos usuarios administradores.

2. Módulo de Compras:

Disponible para los clientes del sistema.

- Permite la compra en línea de libros y juegos de mesa.
- Las compras generan automáticamente un identificador de envío (idEnvío) asociado.

3. Módulo de Catálogos:

Accesible solo para administradores.

- Permite dar de baja existencias de libros o juegos de mesa cuando la venta se realiza de forma presencial.
- Permite consultar en qué sucursales se encuentran disponibles los productos del catálogo.

Estructura modular del sistema

4. Módulo de Tablas Administrativas:

Módulo exclusivo para administradores.

- Muestra el contenido de distintas tablas de la base de datos, como:
 - Catálogo de libros
 - Catálogo de juegos de mesa
 - Existencias por sucursal
 - Ventas realizadas
 - Envíos generados
- Permite la consulta detallada de información relacionada con ventas y envíos.

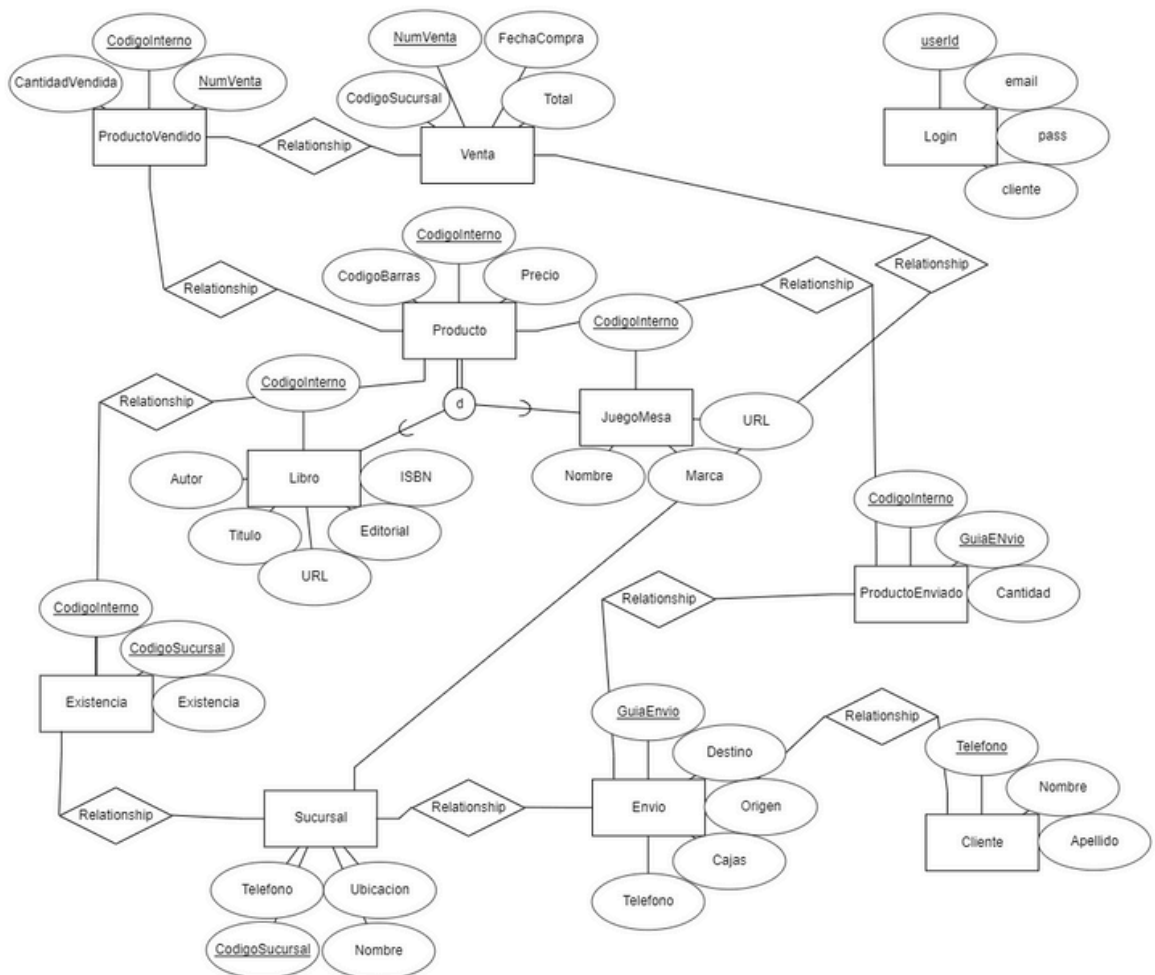
5. Módulo de Envíos:

Dirigido a los clientes.

- Permite consultar el estado de su envío mediante el ingreso del idEnvío generado al momento de la compra.
- Muestra detalles como el estado actual del envío y la sucursal de origen.

BASE DE DATOS

Diagrama Conceptual



Esquema Lógico de la Base de Datos

Especificación Tablas

1.login

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 USERID	NUMBER	No	"WORMBOOK"."ISEQ\$\$_80400".nextval	1	(null)
2 EMAIL	VARCHAR2(255 BYTE)	No	(null)	2	(null)
3 PASSWORD	VARCHAR2(255 BYTE)	No	(null)	3	(null)
4 CLIENTE	NUMBER(1,0)	No	0	4	(null)

2.Envio

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 GUIAENVIO	NUMBER(38,0)	No	(null)	1	(null)
2 NUMCAJAS	NUMBER(38,0)	No	(null)	2	(null)
3 DESTINO	VARCHAR2(100 BYTE)	No	(null)	3	(null)
4 SUCURSALORIGEN	NUMBER(38,0)	No	(null)	4	(null)
5 TELEFONO	NUMBER(38,0)	No	(null)	5	(null)

3.Existencia

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 CODIGOINTERNO	NUMBER(38,0)	No	(null)	1	(null)
2 EXISTENCIA	NUMBER(38,0)	No	(null)	2	(null)
3 CODIGOSUCURSAL	NUMBER(38,0)	No	(null)	3	(null)

4.JuegoMesa

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 CODIGOINTERNO	NUMBER(38,0)	No	(null)	1	(null)
2 NOMBRE	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3 MARCA	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4 URL	VARCHAR2(255 BYTE)	Yes	(null)	4	(null)

5.Libro

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 CODIGOINTERNO	NUMBER(38,0)	No	(null)	1	(null)
2 TITULO	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3 AUTOR	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4 ISBN	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5 EDITORIAL	VARCHAR2(20 BYTE)	Yes	(null)	5	(null)
6 URL	VARCHAR2(255 BYTE)	Yes	(null)	6	(null)

6.Producto

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODIGOINTERNO		NUMBER(38,0)		No	(null)		1		(null)
2	PRECIO		NUMBER(38,0)		Yes	(null)		2		(null)
3	CODIGOBARRAS		NUMBER(38,0)		Yes	(null)		3		(null)

7.ProdcutosEnviados

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODIGOINTERNO		NUMBER(38,0)		No	(null)		1		(null)
2	GUIAENVIO		NUMBER(38,0)		No	(null)		2		(null)
3	CANTIDADENVIADA		NUMBER(38,0)		No	(null)		3		(null)

8.ProductoVendido

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODIGOINTERNO		NUMBER(38,0)		No	(null)		1		(null)
2	NUMVENTA		NUMBER(38,0)		No	(null)		2		(null)
3	CANTIDADVENDIDA		NUMBER(38,0)		No	(null)		3		(null)

9.Sucursal

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODIGOSUCURSAL		NUMBER(38,0)		No	(null)		1		(null)
2	NOMBRESUCURSAL		VARCHAR2(50 BYTE)		No	(null)		2		(null)
3	UBICACION		VARCHAR2(20 BYTE)		No	(null)		3		(null)
4	TELEFONOSUCURSAL		VARCHAR2(20 BYTE)		No	(null)		4		(null)

10.Venta

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	NUMVENTA		NUMBER(38,0)		No	(null)		1		(null)
2	TOTAL		NUMBER(38,0)		Yes	(null)		2		(null)
3	FECHACOMPRA		DATE		Yes	(null)		3		(null)
4	CODIGOSUCURSAL		NUMBER(38,0)		No	(null)		4		(null)

11. Cliente

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	TELEFONO		NUMBER(38,0)		No	(null)		1		(null)
2	NOMBRE		VARCHAR2(20 BYTE)		Yes	(null)		2		(null)
3	APELLIDO		VARCHAR2(20 BYTE)		Yes	(null)		3		(null)

Integridad de Datos:

```
ALTER TABLE Cliente  
ADD CONSTRAINT XPKCliente PRIMARY KEY (Telefono);
```

```
ALTER TABLE Envio  
ADD CONSTRAINT PK_Envio PRIMARY KEY (GuiaEnvio);
```

```
ALTER TABLE Existencia  
ADD CONSTRAINT PK_Existencia PRIMARY KEY  
(CodigoInterno,CodigoSucursal);
```

```
ALTER TABLE JuegoMesa  
ADD CONSTRAINT PK_JuegoMesa PRIMARY KEY (CodigoInterno);
```

```
ALTER TABLE Libro  
ADD CONSTRAINT PK_Libro PRIMARY KEY (CodigoInterno);
```

```
ALTER TABLE Producto  
ADD CONSTRAINT PK_Producto PRIMARY KEY (CodigoInterno);
```

```
ALTER TABLE ProductosEnviados  
ADD CONSTRAINT PK_ProductosEnviados PRIMARY KEY  
(CodigoInterno,GuiaEnvio);
```

```
ALTER TABLE ProductoVendido  
ADD CONSTRAINT PK_ProductoVendido PRIMARY KEY  
(CodigoInterno,NumVenta);
```

```
ALTER TABLE Sucursal  
ADD CONSTRAINT PK_Sucursal PRIMARY KEY (CodigoSucursal);
```

```
ALTER TABLE Venta
ADD CONSTRAINT PK_Venta PRIMARY KEY (NumVenta);
```

```
ALTER TABLE Envio
ADD (CONSTRAINT R_49 FOREIGN KEY (SucursalOrigen) REFERENCES
Sucursal (CodigoSucursal));
```

```
ALTER TABLE Existencia
ADD (CONSTRAINT R_9 FOREIGN KEY (CodigoInterno) REFERENCES
Producto (CodigoInterno));
```

```
ALTER TABLE Existencia
ADD (CONSTRAINT R_22 FOREIGN KEY (CodigoSucursal) REFERENCES
Sucursal (CodigoSucursal));
```

```
ALTER TABLE JuegoMesa
ADD (FOREIGN KEY (CodigoInterno) REFERENCES
Producto(CodigoInterno) ON DELETE CASCADE);
```

```
ALTER TABLE Libro
ADD (FOREIGN KEY (CodigoInterno) REFERENCES
Producto(CodigoInterno) ON DELETE CASCADE);
```

```
ALTER TABLE ProductosEnviados
ADD (CONSTRAINT R_12 FOREIGN KEY (CodigoInterno) REFERENCES
Producto (CodigoInterno));
```

```
ALTER TABLE ProductosEnviados
ADD (CONSTRAINT R_13 FOREIGN KEY (GuiaEnvio) REFERENCES Envio
(GuiaEnvio));
```

```
ALTER TABLE Envio
ADD (CONSTRAINT R_38 FOREIGN KEY (Telefono) REFERENCES Cliente
(Telefono));
```

```
ALTER TABLE ProductoVendido  
ADD (CONSTRAINT R_10 FOREIGN KEY (CodigoInterno) REFERENCES  
Producto (CodigoInterno));
```

```
ALTER TABLE ProductoVendido  
ADD (CONSTRAINT R_10 FOREIGN KEY (CodigoInterno) REFERENCES  
Producto (CodigoInterno));
```

```
ALTER TABLE ProductoVendido  
ADD (CONSTRAINT R_11 FOREIGN KEY (NumVenta) REFERENCES Venta  
(NumVenta));
```

```
ALTER TABLE Venta  
ADD (CONSTRAINT R_26 FOREIGN KEY (CodigoSucursal) REFERENCES  
Sucursal (CodigoSucursal));
```

```
ALTER TABLE Existencia DROP CONSTRAINT R_9;  
ALTER TABLE ProductosEnviados DROP CONSTRAINT R_12;  
ALTER TABLE ProductoVendido DROP CONSTRAINT R_10;
```

```
ALTER TABLE Existencia  
ADD CONSTRAINT R_9 FOREIGN KEY (CodigoInterno)  
REFERENCES Producto(CodigoInterno) ON DELETE CASCADE;
```

```
ALTER TABLE ProductosEnviados  
ADD CONSTRAINT R_12 FOREIGN KEY (CodigoInterno)  
REFERENCES Producto(CodigoInterno) ON DELETE CASCADE;
```

```
ALTER TABLE ProductoVendido  
ADD CONSTRAINT R_10 FOREIGN KEY (CodigoInterno)  
REFERENCES Producto(CodigoInterno) ON DELETE CASCADE;
```

LÓGICA/REGLAS DEL NEGOCIO

- Los usuarios no necesitan iniciar sesión.
- Los administradores deben iniciar sesión para acceder a:
 - Registro de ventas
 - Registro de envíos
 - Control de inventario
- El stock se actualiza automáticamente al registrar una venta.
- No se almacenan datos de tarjetas en ningún momento.
- Los administradores pueden editar el inventario solo en casos especiales:
 - Artículos dañados
 - Devoluciones
- Opciones de compra:
 - Pago en efectivo (Únicamente en sucursal)
 - Pago con tarjeta
- Servicio de envíos disponible cuando no se puede asistir a sucursal.
- El cliente puede ver que sucursales tienen disponible cualquier producto y directamente pueden ordenarlo mediante la pagina
- Por cada artículo se debe realizar una compra
- Al realizar una compra en línea los envíos se comienzan a realizar al día siguiente.
- Todos los artículos comprados el mismo día serán enviados en el mismo paquete a pesar de ser compras separadas.
- La información del envío está disponible en el sistema y todas las guías son de la paquetería FedEx
- En caso de algún problema con el envío o con la existencia de algún producto la empresa se pondrá en contacto con el cliente

DESCRIPCIÓN INTERFAZ DE LA APLICACIÓN

Se muestra como primer página, una página que muestra un botón en caso de ser administrador(empleado) pueda iniciar sesión o registrarse, en caso de ser cliente se presiona en el link debajo del link de administrador.

En caso de ser administrador:

Se muestran por medio de cards, los distintas tablas, al ser presionado alguna opción, manda a la página de la tabla correspondiente, además de tener en dos botones, la posibilidad de editar existencias en libros y juegos de mesa

- Tablas:

1. Se ven los registros de cada tabla de la base de datos

Cliente:

Se muestran 3 opciones en un carrusel, comprar un libro, comprar un juego de mesa o ver envíos.

- Permite al cliente comprar un libro o juego de mesa mediante el catálogo que son cards
- Al presionar una card abre un modal que permite el comprarlo
- Permite buscar el envío con el número de telefono

DESCRIPCIÓN DE REGLAS DE SEGURIDAD

- Los datos de pago no se almacenan en ningún lugar.
- Las contraseñas no especifican cómo deben estar compuestas, pero de preferencia deben ser contraseñas seguras.
- Solo los administradores pueden dar de alta más cuentas de administrador.
- Solo los administradores tienen acceso a la información completa del sistema.
- Seguridad en la capa de datos por la base de datos Oracle.
- Entity Framework proporciona seguridad contra inyección SQL.
- Se puede aplicar un cifrado de datos.
- Se cuenta con 2 niveles de acceso, donde uno tiene mayor acceso a información que el otro.

CONCLUSIÓN

WormBook es una solución tecnológica robusta y eficiente, desarrollada con Entity Framework y una base de datos Oracle, que brinda una experiencia de usuario segura para la gestión de compras y consulta de inventario en tiempo real.

Para los clientes, el sistema ofrece:

- Acceso rápido y confiable a la disponibilidad de productos en distintas sucursales, reduciendo la incertidumbre al momento de buscar un artículo.
- Compra en línea como alternativa cuando el producto no está disponible localmente, garantizando mayor conveniencia.
- Interfaz atractiva, diseñada para destacar información clave sin sacrificar la estética.

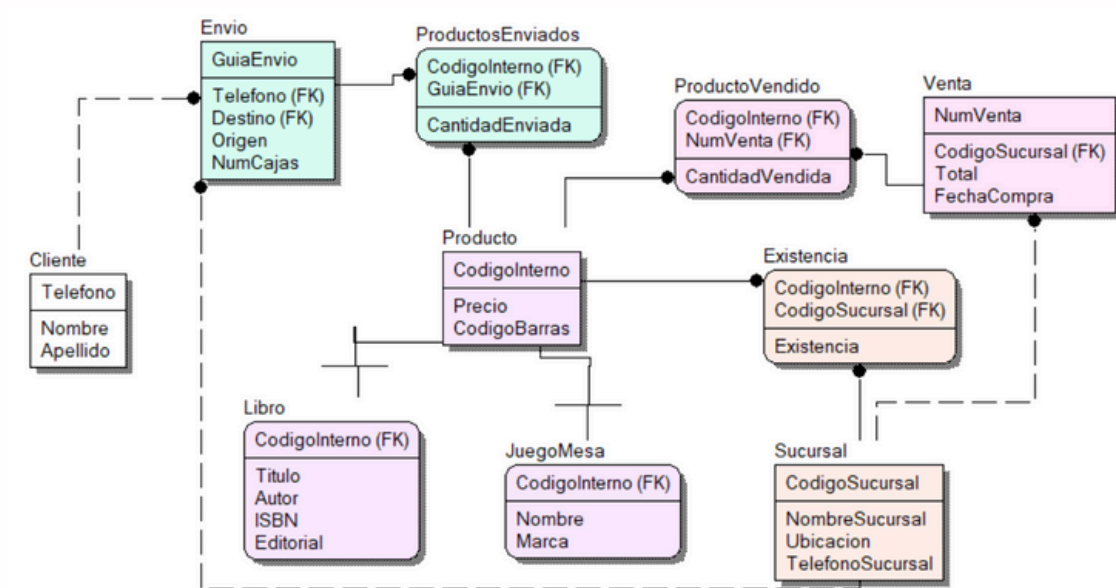
Para los administradores, la plataforma incluye:

- Control total sobre inventarios, ventas y envíos, con permisos restringidos para asegurar la integridad de los datos.
- Herramientas eficientes para actualizar stock y analizar datos comerciales.
- Vistas simplificadas pero funcionales, optimizadas para tareas administrativas sin descuidar la usabilidad.

Además, WormBook no solo mejora la experiencia de compra al ofrecer mayor certeza que los métodos tradicionales, sino que también se posiciona como una plataforma escalable, segura y adaptable a futuras necesidades comerciales. Su accesibilidad web garantiza que usuarios y administradores puedan operarla desde cualquier lugar, consolidándola como una herramienta indispensable para modernizar el comercio minorista.

ANEXOS

Diagrama de la base de datos:



Informacion sobre como usar el sistema tanto como usuairo como administrador disponible en el manual de usuario.

Evidencia del login:

```
--Agregar Login
CREATE TABLE Login (
  userId NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  email VARCHAR2(255) NOT NULL,
  pass VARCHAR2(255) NOT NULL,
  cliente NUMBER(1) DEFAULT 0 NOT NULL
);
ALTER TABLE Login RENAME COLUMN pass TO password;
```


BIBLIOGRAFÍA

REFERENCIAS

- Tecnologías Principales
 - Microsoft (2025). Documentación de ASP.NET MVC.
<https://learn.microsoft.com/en-us/aspnet/core/mvc/>
 - Microsoft (2025). Documentación de Entity Framework Core.
<https://learn.microsoft.com/en-us/ef/core/>
 - Microsoft (2025). Sintaxis de Razor Pages.
<https://learn.microsoft.com/en-us/aspnet/core/razor-pages/>
- Herramientas de Desarrollo
 - Microsoft (2025). Documentación de Visual Studio.
<https://visualstudio.microsoft.com/>
 - Oracle (2025). Documentación de Oracle Database.
<https://docs.oracle.com/en/database/>