

Instituto Tecnológico de Costa Rica  
Centro Académico de Alajuela  
IC4302. Bases de Datos II



**Proyecto #1:**  
**Aplicación Web de apoyo para las PYMES en tiempos de COVID-19**

**Estudiantes:**  
Valery Mishel Carvajal Oreamuno - 2022314299  
Ion Dolanescu Bravo - 2022049034

**Profesor:**  
Alberto Shum Chan

**Fecha de entrega**  
08/04/2024 16:59

I Semestre, 2024

## Índice

<b>Índice.....</b>	<b>1</b>
<b>Descripción del problema y la solución propuesta.....</b>	<b>2</b>
Solución propuesta:.....	2
<b>Descripción del sistema y su alcance.....</b>	<b>3</b>
Alcance del sistema:.....	3
<b>Descripción detallada del modelo de datos (entidades, atributos, relaciones, cardinalidad y participación).....</b>	<b>4</b>
• Backend:.....	4
Entidades y atributos:.....	4
Tablas Transaccionales y atributos:.....	4
Tabla de Bitácora:.....	5
Secuencias:.....	5
Relaciones, Cardinalidad y Participación:.....	5
• FrontEnd:.....	6
Clases.....	6
Lógica.....	7
Persistencia.....	7
Servlets.....	7
WebApp.....	8
<b>Diagrama Entidad-Relación (normalizado al menos en la tercera forma normal).....</b>	<b>10</b>

## **Descripción del problema y la solución propuesta.**

La pandemia del COVID-19 ha tenido un impacto significativo en las pequeñas y medianas empresas (PYMES), afectando su operación comercial y su capacidad para adaptarse a las nuevas condiciones del mercado. Muchas PYMES enfrentan desafíos en términos de gestión de inventario, ventas, seguimiento de transacciones y análisis de datos, entre otros aspectos. La falta de herramientas adecuadas para enfrentar estos desafíos puede llevar a una disminución en la eficiencia operativa y a dificultades para mantenerse competitivas en el mercado.

### **Solución propuesta:**

Para abordar los desafíos mencionados anteriormente, se propone desarrollar una aplicación web de apoyo para las PYMES en tiempos de COVID-19. Esta aplicación estará diseñada para proporcionar a las PYMES las herramientas necesarias para gestionar eficazmente sus operaciones comerciales, incluso en medio de la pandemia. La solución propuesta incluirá las siguientes características clave:

**Gestión de inventario:** La aplicación permitirá a las PYMES llevar un registro detallado de su inventario, incluyendo la cantidad de productos disponibles, su ubicación, precios y detalles de proveedores.

**Ventas y transacciones:** Se proporcionarán herramientas para gestionar las ventas, registrar transacciones y generar informes para un análisis detallado.

**Autenticación de usuarios:** Se implementará un sistema de autenticación de usuarios para garantizar que solo usuarios autorizados tengan acceso a la aplicación y a los datos sensibles.

**Análisis de datos:** La aplicación ofrecerá funciones de análisis de datos para ayudar a las PYMES a comprender mejor su rendimiento comercial, incluyendo informes de resumen sobre ventas registradas en la base de datos.

**Auditoría de cambios:** Se implementará una funcionalidad de bitácora mediante disparadores (triggers) en la base de datos para llevar un registro de los cambios realizados en ciertas entidades, proporcionando una mayor transparencia y trazabilidad en las operaciones.

En resumen, la solución propuesta proporcionará a las PYMES una herramienta integral y fácil de usar para gestionar sus operaciones comerciales durante la pandemia del COVID-19, ayudándolas a adaptarse a las nuevas condiciones del mercado y a mantener su competitividad.

## **Descripción del sistema y su alcance.**

El sistema propuesto es una aplicación web diseñada para brindar apoyo a las pequeñas y medianas empresas (PYMES) durante la pandemia del COVID-19. Está desarrollada utilizando tecnologías Java, Apache Tomcat y una base de datos Oracle. La aplicación seguirá una arquitectura de 3 capas para separar las responsabilidades y mejorar la modularidad del sistema.

El sistema incluye una interfaz de usuario intuitiva y fácil de usar, que permite a los usuarios acceder a funcionalidades de la aplicación. Se implementa un sistema de autenticación de usuarios para garantizar la seguridad y la privacidad de los datos, y se proporcionan pantallas de mantenimiento para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

Además, se utilizan procedimientos y funciones en la base de datos para implementar la lógica empresarial y las operaciones de consulta, garantizando un rendimiento óptimo y una fácil mantenibilidad del sistema.

### **Alcance del sistema:**

1. Gestión de inventario: Registro y seguimiento de productos, precios, existencias y proveedores.
2. Gestión de ventas y compras: Permite registrar las ventas de productos a los clientes y las compras de productos a los proveedores.
3. Autenticación de usuarios: Sistema de inicio de sesión seguro para usuarios autorizados.
4. Análisis de datos: Generación de informes de resumen sobre ventas.
5. Auditoría de cambios: Registro de cambios realizados en ciertas entidades de la base de datos mediante la implementación de disparadores (triggers).
6. Gestión de facturas: Permite registrar las ventas realizadas a los clientes, incluyendo la fecha, hora, cliente, empleado que realizó la venta y el total de la factura. También permite realizar consultas detalladas sobre las facturas.

## Descripción detallada del modelo de datos (entidades, atributos, relaciones, cardinalidad y participación)

- **Backend:**

### Entidades y atributos:

#### 1. Cliente:

- Atributos:
  - id (PK): Identificador único del cliente.
  - nombre: Nombre del cliente.
  - direccion: Dirección del cliente.
  - telefono: Número de teléfono del cliente.

#### 2. Empleado:

- Atributos:
  - id (PK): Identificador único del empleado.
  - nombre: Nombre del empleado.
  - telefono: Número de teléfono del empleado.
  - salario\_mensual: Salario mensual del empleado.
  - puesto: Puesto del empleado.

#### 3. Proveedor:

- Atributos:
  - id (PK): Identificador único del proveedor.
  - nombre: Nombre del proveedor.
  - telefono: Número de teléfono del proveedor.

#### 4. Producto:

- Atributos:
  - id (PK): Identificador único del producto.
  - nombre: Nombre del producto.
  - precio\_unidad: Precio unitario del producto.
  - id\_proveedor: Identificador del proveedor que suministra el producto.
- Relaciones:
  - id\_proveedor (FK) -> Proveedor (id)

### Tablas Transaccionales y atributos:

#### 5. Factura:

- Atributos:
  - id\_factura (PK): Identificador único de la factura.
  - fecha: Fecha de la factura.
  - hora: Hora de la factura.
  - id\_cliente: Identificador del cliente asociado a la factura.
  - id\_cajero: Identificador del empleado (cajero) que realizó la factura.
  - total: Total de la factura.
- Relaciones:
  - id\_cliente (FK) -> Cliente (id)
  - id\_cajero (FK) -> Empleado (id)

#### 6. Usuario:

##### - Atributos:

- id\_empleado (PK, FK): Identificador único del empleado, también clave primaria de Usuario.

- contraseña: Contraseña del usuario.

##### - Relaciones:

- id\_empleado (FK) -> Empleado (id)

#### 7. Venta\_Producto:

##### - Atributos:

- id\_factura (PK, FK): Identificador de la factura asociada a la venta.

- id\_producto (PK, FK): Identificador del producto vendido.

- precio\_venta: Precio de venta del producto.

- cantidad: Cantidad de unidades vendidas.

##### - Relaciones:

- id\_factura (FK) -> Factura (id\_factura)

- id\_producto (FK) -> Producto (id)

#### 8. Compra\_Proveedor:

##### - Atributos:

- id\_producto (PK, FK): Identificador del producto comprado.

- id\_proveedor (PK, FK): Identificador del proveedor al que se compró el producto.

- cantidad\_producto: Cantidad de unidades compradas.

- precio\_venta: Precio de venta del producto por parte del proveedor.

- fecha: Fecha de la compra.

- hora: Hora de la compra.

##### - Relaciones:

- id\_producto (FK) -> Producto (id)

- id\_proveedor (FK) -> Proveedor (id)

#### **Tabla de Bitácora:**

#### 9. Producto\_Log:

##### - Atributos:

- id\_producto (FK): Identificador del producto asociado al cambio de precio.

- fecha\_hora: Fecha y hora del cambio de precio.

- precio\_anterior: Precio anterior del producto.

- precio\_nuevo: Nuevo precio del producto.

#### **Secuencias:**

- Se crean secuencias (que inician en 1 e incrementan 1) para la generación automática de valores para los id de las entidades: Cliente, Empleado, Factura, Proveedor y Producto.

#### **Relaciones, Cardinalidad y Participación:**

- Cliente - Factura:

Cardinalidad: 1 a muchos (Un cliente puede tener varias facturas)

Participación: Total (Cada factura debe estar asociada con un cliente)

- Empleado - Factura:

Cardinalidad: 1 a muchos (Un empleado puede registrar varias facturas como cajero)

Participación: Total (Cada factura debe estar asociada con un empleado)

- Producto - Proveedor:

Cardinalidad: muchos a uno (Varios productos pueden ser suministrados por un proveedor)

Participación: Total (Cada producto tiene un proveedor)

- Usuario - Empleado:

Cardinalidad: uno a uno (Cada usuario está asociado con un empleado)

Participación: Parcial (No todos los empleados necesariamente tienen un usuario)

- Venta\_Producto - Factura:

Cardinalidad: uno a muchos (Una factura puede contener varios productos)

Participación: Parcial (No todas las factura necesariamente tienen productos vendidos)

- Venta\_Producto - Producto:

Cardinalidad: uno a muchos (Un producto puede ser vendido en múltiples facturas)

Participación: Parcial (No todos los productos necesariamente se venden)

- Compra\_Proveedor - Proveedor:

Cardinalidad: uno a muchos (Un proveedor puede suministrar varios productos)

Participación: Total (Cada proveedor tiene compras asociadas)

- Compra\_Proveedor - Producto:

Cardinalidad: uno a muchos (Un producto puede ser comprado a varios proveedores)

Participación: Total (Cada producto tiene su proveedor)

- **FrontEnd:**

### **Clases**

Las clases implementadas en el frontend ubicadas en el paquete clases, funcionan son representantes de tablas en la base de datos, funcionando para traer los datos de la misma y poder visualizarlos.

Las clases implementadas son:

- Cliente
- Empleado
- Usuario
- Producto
- Venta\_Producto
- Factura

## Lógica

Las clases implementadas en el frontend ubicadas en el paquete lógica, son necesarias para la capa entre el servidor y el cliente web, a su vez previniendo de accesos indebidos en los datos.

Las clases implementadas son:

- Autenticación: se encarga de las operaciones lógicas para la prevención del acceso sin un usuario válido.
- Sesión: se encarga de preservar parte de la información para evitar hacer llamadas concurrentes a la base de datos y brindarle el acceso a esta a los demás controladores e interfaces.
- Controladora<Inicial de la tabla>: son los operadores lógicos que se encargan de administrar la creación de las clases correspondientes e iniciar el proceso de almacenamiento, edición y eliminación de los mismos en la base de datos.

## Persistencia

Aquí se encuentran los controladores necesarios para la persistencia de la base de datos, siendo el paquete responsable de enviar y recibir información de la base de datos, estos controladores están vinculados a una PersistenceUnit para poder funcionar correctamente.

Las clases implementadas son:

- CtrlPer<Inicial de la tabla>: Permite la llamada al JpaController correspondiente.
- <Nombre de la tabla>JpaController: es una clase que cumple con la funcionalidad de ejecutar sentencias sql a la base de datos, iniciando y finalizando transacciones para que estas se vean reflejadas.

## Servlets

En este paquete se encuentran las unidades lógicas para el correcto funcionamiento de la interfaz web, cada Sv<nombre>, recibe una sentencia de POST o GET para iniciar acciones en el paquete de la lógica y finalmente enviar un response con una nueva página.

Las clases implementadas son:

- SvIndex
  - doPOST: este método únicamente hace un response a index.jsp.
- SvFallo
  - doPOST: este método es activado desde falloCredencial.jsp con el method POST, este método únicamente hace un response a index.jsp
- SvCaja:
  - doGet: este método es activado desde index.jsp con el method GET, primero hace la solicitud a la lógica para comprobar que el usuario tenga permisos necesarios para continuar y administre la contraseña correcta, si esto es así avanza a la hora de hacer un response a prefacturacion.jsp y en caso contrario hará un response a falloCredencial.jsp.
- SvCliente



- doGet: este método es activado desde prefacturacion.jsp con el method GET, este se encarga de solicitar la existencia de un cliente con el ID indicado, para hacer un response a facturacion.jsp y creando la factura inicial, en caso de ingresar un ID fuera de la base de datos, hará un response recargando la página.
- SvFactura
  - doPOST: este método es activado desde facturacion.jsp con el method POST, este metodo se encarga de solicitar a la logica la eliminación de la factura que se encuentra en proceso de ser generada junto a los productos relacionados, y hacer un response a index.jsp.
- SvIncluir
  - doPOST: este metodo es activado desde incluirProd.jsp con el method POST, este metodo verifica la solicitud de incluir un nuevo producto y cantidad a la lista de compras, y mediante los controladores de la logica incluye este producto a la relación de la compra con los productos, a la vez que actualiza la factura en la tabla, finalmente hace un response a facturacion.jsp para reflejar la inserción del producto.
- SvLista
  - doPOST: este método es llamado desde facturacion.jsp con el method POST, se encarga de hacer únicamente un response a incluirProd.jsp.
- SvPago
  - doGET este método es activado desde facturacion.jsp con el method GET, únicamente hace response hacia pagoCompletado.jsp finalizando con las operaciones sobre la factura.
  - doPOST: este método es llamado desde facturacion.jsp con el method POST, este cumple la función de limpiar la información de Sesion y hacer un response a index.jsp

## WebApp

Corresponde a la parte grafica del cliente web de la API, las paginas cargadas corresponden a:

- index.jsp
  - Consta de un formulario para iniciar la autorizacion de empleado a la caja
- falloCredencial.jsp
  - Es una página indicando que alguna de las credenciales ingresadas no corresponden con información almacenada en la base de datos
- prefacturacion.jsp
  - Consta de un formulario donde se debe ingresar el ID del cliente quien realiza la compra.
- facturacion.jsp
  - Consta de información relacionada a la factura, como el nombre del cajero responsable, una lista de los productos que están vinculados a la misma y el total actual de la factura

- Consta de 3 botones de formulario que cumplen con la función para iniciar el proceso de añadir un nuevo producto a la factura, completar el pago o cancelar la factura
- incluirProd.jsp
  - Cuenta con un formulario para poder añadir el ID de la base de datos y la cantidad del mismo a comprar
- pagoCompletado.jsp
  - Cuenta con un mensaje indicando que se completó el pago y un botón para regresar a index.jsp

## Diagrama Entidad-Relación (normalizado al menos en la tercera forma normal).

