



Caso sistema móvil 2023

ANDROID APP APLICACIÓN WEB

Correo electrónico: ingdannysaldana@gmail.com

Tel.: 54139354

TABLA DE CONTENIDO

Analisis en componentes y arquitectura	3
Backend: JAVA SPRINGBOOT o JAVA o JAVA SPRING	3
APP: REACT NATIVE o FLUTTER o ANDROID STUDIO	4
Front End: ANGULAR o REACTJS o PRIMEFACES o NODEJS	5
Web Server	6
Funcionalidades	8
Cientes	9
Registro de clientes	9
Historial de clientes	9
CRM	10
Balance de flujo de efectivo	¡Error! Marcador no definido.
Facturación	12
Facturas	12
Integración contable con CRM	13
Información de trabajo	13
Control de inventarios	14
Inventarios	14
Control empleados	16
Empleados	16
Departamentos	16
Planilla	16
Giro de la empresa	18
Giro del negocio	18

ANALISIS EN COMPONENTES Y ARQUITECTURA

Backend: JAVA SPRINGBOOT o JAVA o JAVA SPRING

JAVA SPRINGBOOT es una excelente opción ya que es un marco de aplicaciones de código abierto que se basa en el marco Spring y es especialmente adecuado para desarrollar aplicaciones web de alto rendimiento y escalabilidad. JAVA y JAVA SPRING son también opciones viables, pero SpringBoot proporciona una mayor facilidad de configuración y despliegue.

Característica	Java Spring	Java Spring Boot	Java
Enfoque	Enfocado en la inversión de control y el patrón de diseño de inyección de dependencia	Enfocado en la simplicidad y la rapidez de desarrollo	Lenguaje de programación general
Configuración	Requiere una configuración manual y explícita para cada componente	Utiliza una configuración automática y predeterminada para la mayoría de los componentes	Requiere configuraciones manuales
Inicio de proyecto	El inicio de un proyecto puede ser complicado y requiere un gran esfuerzo	El inicio de un proyecto es rápido y sencillo gracias a las plantillas predefinidas y herramientas de inicio automático	Requiere de una estructura manual del proyecto
Dependencias	Es necesario agregar manualmente las dependencias necesarias	Utiliza una gestión automática de dependencias mediante el uso de Maven o Gradle	Es necesario agregar manualmente las dependencias necesarias
Recursos	Es necesario configurar manualmente los recursos, como la base de datos y el almacenamiento	Utiliza una configuración automática de recursos mediante el uso de herramientas como H2 o Flyway	Es necesario configurar manualmente los recursos
Despliegue	Requiere una configuración manual y explícita para el despliegue	Utiliza una configuración automática y predeterminada para el	Requiere una configuración manual y explícita para el despliegue

		despliegue en contenedores, como Tomcat o Jetty	
--	--	---	--

APP: REACT NATIVE o FLUTTER o ANDROID STUDIO

REACT NATIVE es una buena opción ya que es un marco de código abierto de Facebook que permite desarrollar aplicaciones móviles nativas para iOS y Android utilizando JavaScript y React. FLUTTER es otra opción viable, es un marco de Google para desarrollar aplicaciones móviles multiplataforma. ANDROID STUDIO es una buena opción si se requiere desarrollar solo para Android.

Característica	React Native	Flutter	Android Studio
Enfoque	Desarrollo móvil cross-platform para iOS y Android	Desarrollo móvil cross-platform para iOS y Android	Desarrollo móvil nativo para Android
Configuración	Requiere configuración previa de JavaScript y React	Requiere conocimiento de Dart	Requiere conocimiento de Java y Android SDK
Inicio de proyecto	Inicio rápido con plantillas y herramientas en línea	Inicio rápido con plantillas y herramientas en línea	Requiere más tiempo para configurar y crear un proyecto
Dependencias	Requiere conocimiento y uso de dependencias de JavaScript	Requiere conocimiento y uso de dependencias de Dart	Requiere conocimiento y uso de dependencias de Java y Android
Recursos	Una amplia comunidad de desarrolladores y recursos en línea	Una comunidad en crecimiento y recursos en línea	Una amplia comunidad de desarrolladores y recursos en línea
Despliegue	Despliegue fácil en plataformas móviles mediante herramientas en línea	Despliegue fácil en plataformas móviles mediante herramientas en línea	Requiere más tiempo y conocimiento para desplegar en la tienda de aplicaciones de Google
Lenguaje de programación	JavaScript	Dart	Java

Comunidad y soporte	Fuerte comunidad y amplio soporte	Creciente comunidad y soporte en aumento	Fuerte comunidad y amplio soporte
Desempeño	Bueno, pero requiere un enfoque en la optimización	Excelente	Bueno
Flexibilidad	Alta	Alta	Alta
Tiempo de desarrollo	Moderado	Moderado	Mayor que React Native y Flutter
Aprendizaje	Fácil para aquellos familiarizados con JavaScript	Nuevo lenguaje (Dart) pero fácil de aprender	Requiere conocimiento de Java
Animaciones	Buen soporte, pero requiere un enfoque en la optimización	Excelente	Bueno

Front End: ANGULAR o REACTJS o PRIMEFACES o NODEJS

ANGULAR es una excelente opción ya que es un marco de código abierto de Google para desarrollar aplicaciones web de una sola página. REACTJS es otra opción viable, es un marco de Facebook para desarrollar aplicaciones web de una sola página. PRIMEFACES es una buena opción si se requiere desarrollar aplicaciones web con un enfoque en JavaServer Faces. NODEJS es una opción viable si se requiere desarrollar aplicaciones web con un enfoque en JavaScript.

Característica	Angular	ReactJS	PrimeFaces	NodeJS
Enfoque	Aplicaciones web complejas y escalables con un enfoque en el lado del cliente.	Aplicaciones web interactivas con un enfoque en el lado del cliente.	Interfaz de usuario para aplicaciones Java en la web con un enfoque en el lado del servidor.	Aplicaciones en tiempo real y conectadas a la web con un enfoque en el servidor.
Configuración	Algo más compleja y requiere un mayor	Fácil de configurar y requiere menos conocimiento previo.	Requiere habilidades en Java y conocimiento previo de	Requiere habilidades en JavaScript y conocimiento previo de la

	conocimiento previo.		JavaServer Faces (JSF).	plataforma Node.js.
Inicio de proyecto	Requiere tiempo para configurar y establecer las herramientas adecuadas.	Fácil de comenzar un proyecto gracias a la disponibilidad de plantillas y recursos en línea.	Fácil de comenzar un proyecto gracias a la disponibilidad de plantillas y recursos en línea.	Fácil de comenzar un proyecto gracias a la disponibilidad de plantillas y recursos en línea.
Dependencias	Necesita dependencias externas para funcionar correctamente.	Necesita dependencias externas para funcionar correctamente.	No requiere dependencias externas, ya que está construido en Java.	Requiere dependencias externas para funcionar correctamente.
Recursos	Amplia comunidad y gran cantidad de recursos en línea.	Amplia comunidad y gran cantidad de recursos en línea.	Comunidad y cantidad moderadas de recursos en línea.	Amplia comunidad y gran cantidad de recursos en línea.
Despliegue	Fácil de desplegar en múltiples plataformas.	Fácil de desplegar en múltiples plataformas.	Fácil de desplegar en servidores Java.	Fácil de desplegar en múltiples plataformas.

Para este proyecto se utilizará: Java Spring Boot, React Native, Angular y Weblogic.

Web Server

WEBLOGIC es un servidor web Java que es una excelente opción para desarrollar aplicaciones web escalables y de alto rendimiento.

Ventajas	<p>Integración fácil: WebLogic es una plataforma de Java EE integral que permite a los desarrolladores de Java integrar fácilmente aplicaciones, datos y servicios web.</p> <p>Alta disponibilidad: WebLogic ofrece una alta disponibilidad y tolerancia a fallos para garantizar que las aplicaciones estén disponibles en todo momento.</p> <p>Escalabilidad: WebLogic puede manejar una gran cantidad de tráfico y solicitudes de forma eficiente, lo que lo hace adecuado para aplicaciones de gran escala.</p> <p>Facilidad de uso: WebLogic es fácil de usar y configurar, lo que lo hace ideal para desarrolladores novatos y experimentados.</p> <p>Soporte de Oracle: Oracle brinda soporte completo a WebLogic, lo que significa que los usuarios pueden estar seguros de recibir actualizaciones y soluciones a problemas en un plazo rápido</p>
Desventajas	<p>Costo: WebLogic es un producto de pago y puede ser costoso para algunas empresas o proyectos.</p> <p>Complejidad: Aunque es fácil de usar, WebLogic puede ser complejo de configurar y administrar en comparación con otras plataformas de Java.</p> <p>Requisitos de sistema: WebLogic requiere un hardware y un sistema operativo específicos, lo que puede limitar la capacidad de algunas empresas para implementarlo.</p> <p>Limitaciones de compatibilidad: WebLogic solo es compatible con Java, lo que significa que no es una opción para desarrolladores de otras tecnologías o lenguajes de programación.</p> <p>Actualizaciones frecuentes: Oracle suele lanzar actualizaciones frecuentes de WebLogic, lo que puede requerir una reconfiguración y una nueva formación para los usuarios.</p>

FUNCIONALIDADES

Como parte del proceso de Innovación Digital y mejora continua en conjunto con el área de operaciones del negocio, se ha definido el desarrollo de un nuevo sistema para proveer movilidad, agilidad, control de los recursos, y por supuesto mejorar CX, por lo que se han hecho funciones enfocadas al:

- Servicio ofrecido a los clientes
- Gestión del negocio.



Las funcionalidades del nuevo sistema permitirán una mejora en la experiencia de los clientes, y brindarán una visibilidad clara y precisa en la gestión de los recursos y operaciones del negocio. El objetivo es proporcionar una solución eficiente y a la altura de las expectativas, con una interfaz fácil de usar y accesible en todo momento. Se espera que el nuevo sistema permita al negocio operar con mayor eficiencia y agilidad, mejorando la toma de decisiones y permitiendo un mejor seguimiento de los procesos.

CLIENTES

El objetivo principal de guardar los datos de los clientes y sus necesidades en una tabla CRM (Customer Relationship Management) es mejorar la relación con los clientes y aumentar las ventas. Al tener un registro centralizado de la información de los clientes, los vendedores y el equipo de atención al cliente pueden conocer mejor a sus clientes, identificar sus necesidades y personalizar la comunicación y el ofrecimiento de productos o servicios.

Registro de clientes

Almacena información básica sobre los clientes, como su nombre, apellido, dirección, teléfono, correo, fecha de nacimiento, género, estado civil, profesión y comentarios. Esta información puede ser **utilizada para segmentar a los clientes y personalizar la experiencia de estos.**

```
CREATE TABLE Clientes (  
  id INT PRIMARY KEY,  
  nombre VARCHAR(255),  
  apellido VARCHAR(255),  
  direccion VARCHAR(255),  
  telefono VARCHAR(255),  
  correo VARCHAR(255),  
  fecha_nacimiento DATE,  
  genero ENUM('hombre', 'mujer', 'otro'),  
  estado_civil ENUM('soltero', 'casado', 'viudo', 'divorciado'),  
  profesion VARCHAR(255),  
  comentarios VARCHAR(255),  
  tinyint(1)  
);
```

Historial de clientes

Almacena el historial de interacciones entre la empresa y sus clientes, incluyendo la descripción de la interacción.

-

```
CREATE TABLE Historial_Clientes (
  id INT PRIMARY KEY,
  id_cliente INT,
  id_factura INT,
  descripcion VARCHAR(255),
  tinyint(1),
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id),
  FOREIGN KEY (id_factura) REFERENCES Facturas(id)
);
```

CRM

Almacena información sobre las relaciones con los clientes, incluyendo la necesidad del cliente, el historial de mantenimiento y las oportunidades de venta.

- Seguimiento de clientes: La tabla puede ser utilizada para mantener un registro de la actividad de los clientes, sus necesidades y preferencias, y para llevar un seguimiento de las interacciones con ellos.
- Análisis de datos: La información almacenada en la tabla puede ser utilizada para realizar análisis de datos para identificar patrones y tendencias en el comportamiento de los clientes y mejorar la toma de decisiones de la empresa.
- Identificación de oportunidades de venta: La tabla puede ayudar a identificar oportunidades de venta al mostrar las necesidades y preferencias de los clientes y el historial de mantenimiento de los productos que han comprado.
- Mejora de la relación con los clientes: La tabla puede ser utilizada para mejorar la relación con los clientes al permitir a la empresa ofrecer un servicio personalizado y conocer sus necesidades específicas.
- Automatización de procesos: La tabla puede ser utilizada para automatizar procesos relacionados con el seguimiento de clientes y la identificación de oportunidades de venta.

```
CREATE TABLE CRM (
  id INT PRIMARY KEY,
  id_cliente INT,
  id_factura INT,
  necesidad_cliente VARCHAR(255),
  historial_mantenimiento VARCHAR(255),
  oportunidad_venta VARCHAR(255),
  tinyint(1),
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id),
  FOREIGN KEY (id_factura) REFERENCES Facturas(id)
);
```

En resumen, la tabla CRM puede ser una herramienta valiosa para la empresa al ayudar a mejorar la toma de decisiones, mejorar la relación con los clientes y automatizar procesos.



FACTURACIÓN

El registro de facturas en un sistema permite a las empresas tener un control más eficiente y organizado de sus operaciones financieras y de facturación. Además, almacenar esta información en un sistema permite realizar un seguimiento más sencillo de los pagos pendientes y realizados, así como también permitir la generación de informes y estadísticas que pueden ser utilizadas para tomar decisiones estratégicas y mejorar la gestión empresarial.

Facturas

El registro de facturas en un sistema también puede mejorar la relación con los clientes, ya que permite tener una visibilidad clara de sus compras y pagos, lo que puede ser útil en caso de reclamaciones o para ofrecer un mejor servicio al cliente.



```
CREATE TABLE Factura_Encabezado (
  id INT PRIMARY KEY,
  id_cliente INT,
  fecha DATE,
  monto DECIMAL(10,2),
  forma_pago ENUM('efectivo', 'tarjeta', 'transferencia'),
  tinyint(1),
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id)
);

CREATE TABLE Detalles_Factura (
  id INT PRIMARY KEY,
  id_factura INT,
  id_producto INT,
  cantidad INT,
  precio DECIMAL(10,2),
  tinyint(1),
  FOREIGN KEY (id_factura) REFERENCES Factura_Encabezado(id),
  FOREIGN KEY (id_producto) REFERENCES Productos(id)
);
```

INTEGRACIÓN CONTABLE CON CRM

Información de trabajo

Tener la información puede aportar un valor agregado a la empresa al permitirle:

- Monitorear el desempeño laboral de los empleados y tomar decisiones basadas en esa información.
- Evaluar la productividad y eficiencia de los empleados.
- Identificar y abordar áreas de mejora en la gestión de recursos humano

```
CREATE TABLE Informacion_Trabajo (
  id INT PRIMARY KEY,
  id_factura INT,
  id_cliente INT,
  descripcion VARCHAR(255),
  estado tinyint(1),
  FOREIGN KEY (id_factura) REFERENCES Facturas(id),
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id)
);
```

Además, esta tabla se puede integrar con la información de documentos de respaldo para asegurar la veracidad de la información.

Por lo que se necesita un gestor de archivos.

```
CREATE TABLE Imagenes (
  id INT PRIMARY KEY,
  id_trabajo INT,
  url VARCHAR(255),
  estado tinyint(1),
  FOREIGN KEY (id_trabajo) REFERENCES Informacion_Trabajo(id)
);

CREATE TABLE Documentos (
  id INT PRIMARY KEY,
  id_trabajo INT,
  url VARCHAR(255),
  estado tinyint(1),
  FOREIGN KEY (id_trabajo) REFERENCES Informacion_Trabajo(id)
);

CREATE TABLE Archivos (
  id INT PRIMARY KEY,
  id_trabajo INT,
  url VARCHAR(255),
  estado tinyint(1),
  FOREIGN KEY (id_trabajo) REFERENCES Informacion_Trabajo(id)
);
```

CONTROL DE INVENTARIOS

Inventarios

Tener un control de inventarios permite a las empresas conocer la cantidad de bienes que tienen en existencia en un momento determinado, y esta información se utiliza para tomar decisiones de manejo de inventarios, tales como cuándo realizar un pedido de reposición, qué artículos son más populares y requieren una mayor inversión, y cómo maximizar la utilización de los recursos almacenados.

```
CREATE TABLE Informacion_Trabajo (  
  id INT PRIMARY KEY,  
  id_factura INT,  
  id_cliente INT,  
  descripcion VARCHAR(255),  
  estado tinyint(1),  
  FOREIGN KEY (id_factura) REFERENCES Facturas(id),  
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id)  
);
```

Permite detectar posibles problemas de faltantes o sobre inventarios y tomar medidas para corregirlos, lo que puede mejorar la eficiencia y reducir los costos.

Además, para el manejo es necesario llevar el control de las compras y ventas.

El control de compras y ventas de mercancías es importante porque permite a una empresa llevar un registro detallado y actualizado de sus entradas y salidas de bienes, lo que a su vez ayuda a:

- **Monitorear el inventario:** Conocer las existencias de productos permite a la empresa tomar decisiones informadas sobre cuándo realizar nuevas compras o producciones.
- **Analizar la rentabilidad:** Conociendo los costos de compra y venta de cada producto, es posible determinar los márgenes de ganancia y analizar cuáles son los productos más rentables para la empresa.

- Identificar oportunidades de mejora: El control de compras y ventas permite a la empresa detectar problemas de faltantes, sobrecompra, productos que no se venden, etc. y tomar medidas para mejorar su gestión.
- Cumplir con requisitos fiscales: En muchos países, es obligatorio llevar un registro de las transacciones comerciales para cumplir con los requisitos fiscales.

```
CREATE TABLE Ordenes_Compra (  
  id INT PRIMARY KEY,  
  id_proveedor INT,  
  fecha DATE,  
  monto DECIMAL(10,2),  
  estado tinyint(1),  
  FOREIGN KEY (id_proveedor) REFERENCES Proveedores(id)  
);  
  
CREATE TABLE Detalles_Orden (  
  id INT PRIMARY KEY,  
  id_orden INT,  
  id_producto INT,  
  cantidad INT,  
  precio DECIMAL(10,2),  
  estado tinyint(1),  
  FOREIGN KEY (id_orden) REFERENCES Ordenes_Compra(id),  
  FOREIGN KEY (id_producto) REFERENCES Productos(id)  
);
```

CONTROL EMPLEADOS

Tener un control de empleados permite a una empresa monitorear y evaluar el desempeño laboral de sus trabajadores, así como también conocer y gestionar sus horas trabajadas, vacaciones, salarios y otras informaciones relevantes. Esto ayuda a la empresa a tomar decisiones informadas sobre la gestión de su fuerza laboral y a mejorar su productividad y eficiencia.

Empleados

El valor agregado que le da a la empresa conocer esta información es la capacidad de tomar decisiones más precisas y efectivas sobre la gestión de sus recursos humanos, incluyendo la planificación de la fuerza laboral, la gestión de los salarios y la motivación del personal. También puede mejorar la comunicación y la transparencia interna, lo que a su vez puede mejorar la moral y la satisfacción del trabajador.

```
CREATE TABLE Empleados (  
  id INT PRIMARY KEY,  
  nombre VARCHAR(255),  
  id_departamento INT,  
  estado tinyint(1),  
  FOREIGN KEY (id_departamento) REFERENCES Departamentos(id)  
);
```

Departamentos

Sirve para organizar y administrar de manera eficiente la estructura de la organización. Con este registro, se pueden conocer las funciones y responsabilidades de cada departamento, identificar las áreas que requieren más recursos o apoyo, y coordinar las actividades de los departamentos de manera efectiva para lograr los objetivos de la empresa.

```
CREATE TABLE Departamentos (  
  id INT PRIMARY KEY,  
  nombre VARCHAR(255),  
  estado tinyint(1)  
);
```

Planilla

Registrar la planilla es un proceso importante para una empresa, ya que permite tener un registro preciso y detallado de las remuneraciones y descuentos que se aplican a los

empleados. Esto es útil para cumplir con las obligaciones fiscales y previsionales, así como para llevar un control de los costos laborales y de la nómina de la empresa. Además, la planilla permite establecer un sistema de compensaciones justo para los empleados y mejorar la gestión de recursos humanos.

```
CREATE TABLE Descuentos (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  ID_Empleado INT NOT NULL,
  Descripcion VARCHAR(255) NOT NULL,
  Monto DECIMAL(10, 2) NOT NULL,
  Fecha DATE NOT NULL,
  FOREIGN KEY (ID_Empleado) REFERENCES Informacion_trabajo(ID)
);

CREATE TABLE Dias_trabajados (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  ID_Empleado INT NOT NULL,
  Fecha DATE NOT NULL,
  Dias INT NOT NULL,
  FOREIGN KEY (ID_Empleado) REFERENCES Informacion_trabajo(ID)
);

CREATE TABLE Permisos_licencias (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  ID_Empleado INT NOT NULL,
  Descripcion VARCHAR(255) NOT NULL,
  Fecha_inicio DATE NOT NULL,
  Fecha_fin DATE NOT NULL,
  FOREIGN KEY (ID_Empleado) REFERENCES Informacion_trabajo(ID)
);

CREATE TABLE Ausencias (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  ID_Empleado INT NOT NULL,
  Descripcion VARCHAR(255) NOT NULL,
  Fecha DATE NOT NULL,
  FOREIGN KEY (ID_Empleado) REFERENCES Informacion_trabajo(ID)
);

CREATE TABLE Bonificaciones (
  ID INT PRIMARY KEY AUTO_INCREMENT,
  ID_Empleado INT NOT NULL,
  Descripcion VARCHAR(255) NOT NULL,
  Monto DECIMAL(10, 2) NOT NULL,
  Fecha DATE NOT NULL,
  FOREIGN KEY (ID_Empleado) REFERENCES Informacion_trabajo(ID)
);
```

GIRO DE LA EMPRESA

Registrar las actividades diarias del giro del negocio es importante por varias razones:

- **Control financiero:** Al tener un registro detallado de las transacciones diarias, es posible tener un control financiero más efectivo y una visión clara de los ingresos y los gastos del negocio.
- **Toma de decisiones:** Los registros diarios proporcionan información valiosa para la toma de decisiones empresariales y para evaluar el rendimiento del negocio.
- **Cumplimiento de leyes y regulaciones:** Muchos negocios están obligados por ley a mantener registros detallados y precisos de sus actividades comerciales.
- **Referencia para el futuro:** Los registros diarios son una fuente importante de información para revisar el progreso del negocio a largo plazo y planificar su estrategia futura.

Giro del negocio

Se debe ver los procesos del giro habitual de la empresa y su nivel de cumplimiento.

```
CREATE TABLE despacho (
  id INT PRIMARY KEY,
  id_tecnico INT,
  id_cliente INT,
  id_trabajo INT,
  fecha_inicio DATE,
  fecha_fin DATE,
  hora_inicio TIME,
  hora_fin TIME,
  latitud FLOAT,
  longitud FLOAT,
  cumplimiento ENUM('pendiente', 'en proceso', 'completado'),
  estado tinyint(1),
  FOREIGN KEY (id_tecnico) REFERENCES tecnico(id),
  FOREIGN KEY (id_cliente) REFERENCES cliente(id),
  FOREIGN KEY (id_trabajo) REFERENCES trabajo(id)
);
```