

I.E.S Fuengirola N°1

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROYECTO INTEGRADO

StockFlow

Departamento de Informática

Memoria descriptiva

Autor: Eric Villalba Silvester

Índice

1. Sobre este proyecto

1.1 Control de versiones

1.2 Licencia de uso

2. Análisis del problema

2.1 Introducción al problema

2.2 Antecedentes

2.3 Objetivos

2.4 Requisitos

2.4.1 Funcionales

2.4.2 No funcionales

2.5 Recursos

2.5.1 Software

2.5.2 Hardware

3. Diseño de la solución software

3.1 Modelados

3.1.1 Casos de uso

3.1.2 Interacción

3.1.3 Estado

3.1.4 Actividad

3.2 Base de datos

3.2.1 Diseño Conceptual (ER)

3.2.2 Diseño lógico (tablas normalizadas)

4. Implementación

4.1 Codificación

4.1.1 Usabilidad

4.1.2 Backend

4.1.3 Frontend

5. Conclusiones

6. Bibliografía

1 - Sobre este proyecto

1.1 - Control de versiones

El proyecto empleará Git para gestionar el control de versiones. Utilizaremos un repositorio específico creado para el módulo de Proyecto Integrado, donde se almacenarán todas las versiones y actualizaciones del software.

1.2 - Licencia de uso

El software estará bajo una licencia de código abierto, específicamente la Licencia Pública General de GNU, lo que permitirá su libre uso, modificación y distribución.

2 - Análisis del problema

2.1 - Introducción al problema

En muchas pequeñas y medianas empresas, la falta de un sistema de gestión de inventarios y tareas eficiente puede generar serios inconvenientes logísticos, como dificultades para controlar el stock y problemas de comunicación entre departamentos. Una solución digital que centralice y automatice estos procesos podría mejorar significativamente la eficiencia operativa.

2.2 - Antecedentes

Aunque existen múltiples opciones de software para la gestión de inventarios y tareas, muchas de ellas son complejas y costosas, lo que las hace inaccesibles para empresas de menor tamaño. Este proyecto tiene como objetivo ofrecer una alternativa más sencilla, modular y personalizable, adaptada a las necesidades específicas de cada negocio.

2.3 - Objetivos

El propósito de este proyecto es desarrollar una aplicación modular que facilite la gestión de inventarios y tareas dentro de una empresa.

Entre los objetivos específicos se incluyen:

- Crear una plataforma que mejore la organización interna de las empresas y facilite la comunicación.
- Proporcionar herramientas fáciles de usar para tomar decisiones basadas en los datos recopilados.

Funciones previstas para la versión 1.0:

- **Gestión básica de inventarios:** Registro de productos y servicios, control de entradas y salidas, y alertas por bajo inventario.
- **Gestión de usuarios y roles:** Creación de usuarios, asignación de roles y permisos, y control de acceso según el perfil de cada usuario.

2.4 - Requisitos

2.4.1 - Funcionales

- Permitir el registro de productos y servicios con información detallada.
- Generar alertas automáticas cuando los niveles de inventario sean bajos.
- Permitir la creación de usuarios y asignación de roles, con la posibilidad de definir diferentes permisos para cada uno.

2.4.2 - No funcionales

- La aplicación debe contar con una interfaz intuitiva y ágil que facilite su uso.
- Garantizar la seguridad de los datos mediante autenticación y cifrado de la información.

2.5 - Recursos

2.5.1 - Software

- Visual Studio Code, XAMPP, MySQL Workbench, React, Node.js, Docker, Postman, Figma, Trello, Git, y GitHub.

2.5.2 - Hardware

- Se requerirán computadoras personales con especificaciones mínimas, además de acceso a Internet para pruebas y despliegue de la aplicación.

3 - Diseño de la solución software

3.1 - Modelados

3.1.1 - Casos de uso

- Registrar un nuevo producto en el inventario.
- Gestionar las entradas y salidas de productos.
- Crear y asignar roles a los usuarios.

3.1.2 - Interacción

- **Interfaz de usuario:**
 - **Login:** Los usuarios podrán iniciar sesión mediante su correo electrónico y una contraseña.
 - **Panel principal:** Dependiendo del rol del usuario, se le mostrará un menú personalizado con las opciones que tiene habilitadas.
 - **Gestión de inventarios:** Habrá una interfaz gráfica que permitirá gestionar los productos, registrar movimientos y configurar alertas por bajo stock.
- **Flujo de interacción:**
 - Los usuarios inician sesión y acceden al panel principal.
 - Desde allí, seleccionan el módulo correspondiente (Inventarios, Servicios, Roles).
 - Una vez dentro de un módulo, pueden realizar acciones según sus permisos.

3.1.3 - Estado

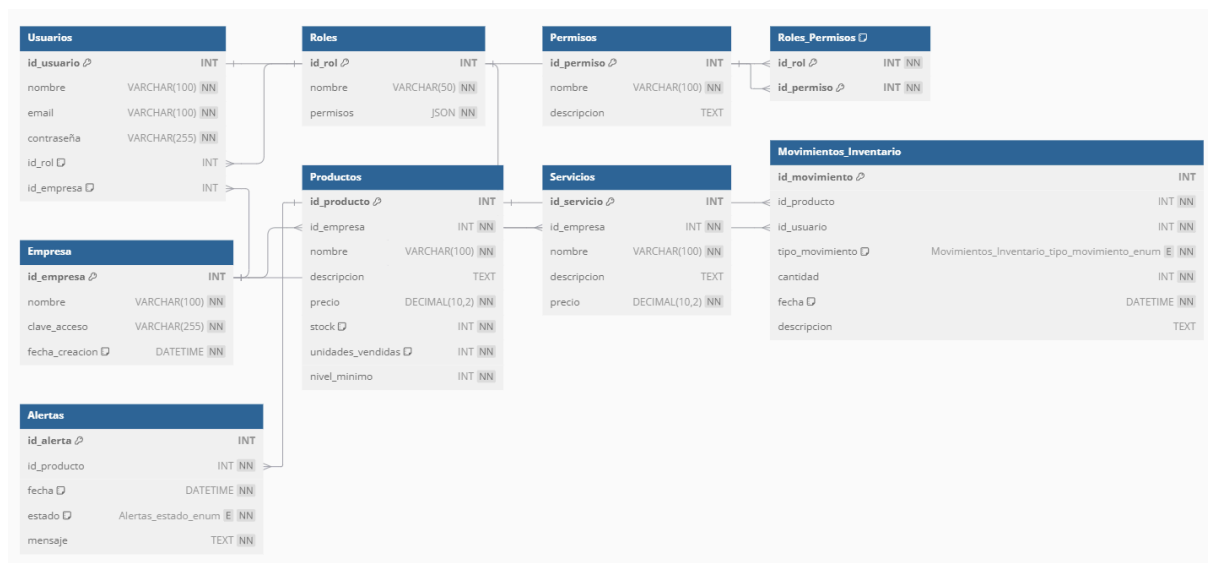
- **Producto:**
 - Estados posibles: "Disponible", "Bajo stock", "Sin stock".
 - Las transiciones ocurren cuando el nivel de un producto se reduce por debajo de un umbral establecido o llega a cero.
- **Usuarios:**
 - Estados posibles: "Activo", "Inactivo", "Bloqueado".
 - Un usuario puede ser bloqueado después de varios intentos fallidos de inicio de sesión o marcado como inactivo si es eliminado del sistema pero sus datos se conservan.

3.1.4 - Actividad

- **Proceso de registro de productos:**
 - El usuario selecciona la opción de "Registrar Producto".
 - Introduce los datos necesarios (nombre, descripción, precio, cantidad inicial).
 - El sistema valida y almacena la información en la base de datos.
- **Gestión de entradas y salidas:**
 - El usuario selecciona un producto del inventario y especifica si es una entrada o salida, junto con la cantidad correspondiente.
 - El sistema actualiza el stock y registra el movimiento.
- **Creación de roles:**
 - El administrador puede crear nuevos roles, definir permisos asociados y almacenarlos para su asignación a usuarios.
- **Alertas de bajo inventario:**
 - El sistema realiza comprobaciones periódicas de inventario y genera alertas cuando el stock de algún producto cae por debajo del umbral mínimo configurado.

3.3 - Base de datos

3.3.1 - Diseño Conceptual (ER)



3.3.2 - Diseño lógico (tablas normalizadas)

Tabla "Usuarios"

- **id_usuario**: Identificador único.
- **nombre**: Nombre completo del usuario.
- **email**: Correo electrónico del usuario.
- **contraseña**: Contraseña cifrada.
- **id_rol**: Relación con la tabla de roles.
- **id_empresa**: Relación con la tabla de empresas.

Tabla "Roles"

- **id_rol**: Identificador único.
- **nombre**: Nombre del rol (administrador, empleado, etc.).
- **permisos**: Lista de permisos asociados al rol en formato JSON.

Tabla "Permisos"

- **id_permiso**: Identificador único.
- **nombre**: Nombre del permiso.
- **descripcion**: Descripción del permiso.

Tabla "Roles_Permisos"

- Relaciona roles con permisos específicos mediante los campos **id_rol** e **id_permiso**.

Tabla "Empresa"

- **id_empresa**: Identificador único.
- **nombre**: Nombre de la empresa.
- **clave_acceso**: Clave para acceder a la empresa.
- **fecha_creacion**: Fecha en que se creó la empresa.

Tabla "Productos"

- **id_producto**: Identificador único del producto.
- **id_empresa**: Relación con la empresa propietaria.
- **nombre**: Nombre del producto.
- **descripcion**: Descripción del producto.
- **precio**: Precio del producto.
- **stock**: Cantidad disponible.
- **unidades_vendidas**: Unidades vendidas hasta la fecha.
- **nivel_minimo**: Umbral de stock mínimo.

Tabla "Movimientos_Inventario"

- **id_movimiento**: Identificador único del movimiento.
- **id_producto**: Producto relacionado.
- **id_usuario**: Usuario que realizó el movimiento.
- **tipo_movimiento**: Tipo de movimiento (entrada/salida).
- **cantidad**: Cantidad del movimiento.
- **fecha**: Fecha del movimiento.
- **descripcion**: Detalles adicionales.

Tabla "Alertas"

- **id_alerta**: Identificador único.
 - **id_producto**: Producto asociado a la alerta.
 - **fecha**: Fecha de la alerta.
 - **estado**: Estado de la alerta (activa o resuelta).
 - **mensaje**: Mensaje detallado de la alerta.
-

4 - Implementación

4.1 - Codificación

4.1.1 - Usabilidad

La aplicación estará diseñada para ser fácil de usar, con una interfaz limpia y un flujo de navegación intuitivo. Esto se logrará mediante el uso de React y CSS, lo que garantizará una experiencia fluida incluso para usuarios sin experiencia técnica.

4.1.2 - Backend

El backend se desarrollará en PHP utilizando MySQL como sistema de gestión de bases de datos. La arquitectura será RESTful, con el objetivo de hacer la integración y el consumo de servicios más sencilla.

4.1.3 - Frontend

El frontend se desarrollará con React, lo que permitirá una interfaz interactiva y de alto rendimiento. Además, se utilizarán herramientas como Redux para gestionar el estado y mejorar la escalabilidad.

5 - Conclusiones

Con la implementación de esta solución, se espera resolver los problemas logísticos comunes en las pequeñas y medianas empresas, como la mala gestión del inventario y la falta de comunicación interna. La aplicación ayudará a las empresas a ser más organizadas y a reducir los errores en sus operaciones diarias.

6 - Bibliografía y documentación

PHP (Backend)

Página oficial: <https://www.php.net/>

Documentación oficial: <https://www.php.net/docs.php>

React (Frontend)

Página oficial: <https://react.dev/>

Documentación oficial: <https://react.dev/learn>

MySQL (Base de Datos)

Página oficial: <https://www.mysql.com/>

Documentación oficial: <https://dev.mysql.com/doc/>

Bootstrap (Framework CSS para diseño)

Página oficial: <https://getbootstrap.com/>

Documentación oficial: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

Visual Studio Code (Editor de Código)

Página oficial: <https://code.visualstudio.com/>

Node.js (Entorno de ejecución para React y utilidades)

Página oficial: <https://nodejs.org/>

Documentación oficial: <https://nodejs.org/en/docs/>

XAMPP (Servidor Local para PHP y MySQL)

Página oficial: <https://www.apachefriends.org/>

MySQL Workbench (Herramienta de diseño y administración para MySQL)

Página oficial: <https://www.mysql.com/products/workbench/>

Git (Control de versiones)

Página oficial: <https://git-scm.com/>

Documentación oficial: <https://git-scm.com/doc>

Trello (Gestión de tareas y planificación)

Página oficial: <https://trello.com/>

Figma (Diseño de interfaces y prototipos)

Página oficial: <https://www.figma.com/>