Universidad del Valle de Guatemala
Departamento de Ingeniería
Sección 20
Carné 23053
Carné 23648
Carné 23559
Carné 23778
Carné 23813

Miércoles 19 de marzo de 2025
Catedrático: Marroquín Rodriguez, Erick Francisco
Ingeniería de Software
Daniela Ramírez de León
Leonardo Dufrey Mejía Mejía
María José Girón Isidro
Melisa Dayana Mendizabal Meléndez
Renato Manuel Rojas Roldán

CORTE 3 DEL PROYECTO – ANÁLISIS Y DISEÑO
"ENTERPRISE RESOURCE PLANNING PARA LA FARMACIA -ECONOFARMA-"

Resumen

El presente proyecto consiste en el desarrollo de una solución para la farmacia Econofarma, ubicada en Quiché, Guatemala. La necesidad surge de las deficiencias en la gestión manual del inventario, ventas y finanzas, que esencialmente es la causa principal de errores y perdida de funcionalidad acertada. La solución propuesta plantea una automatización y sincronización de procesos prácticos, mejorando la eficiencia operativa. La optimización del control de inventario, la digitalización de procesos administrativos, la mejora en la gestión de visitadores médicos y calendarización de citas, son los principales objetivos del proyecto. Esta solución facilitará la toma de decisiones, reducirá errores humanos y permitirá a la farmacia mantener una operatividad óptima y tomar decisiones estratégicas para ofrecer un mejor servicio.

Introducción

La farmacia Econofarma, ubicada en Quiché, Guatemala, es un negocio dedicado a la comercialización de medicamentos y productos relacionados con la salud. A pesar de su importancia en la comunidad, existen desafíos significativos debido a la falta de un sistema eficiente para la gestión de inventario, ventas y procesos administrativos. Actualmente, muchos de estos procedimientos se realizan de manera manual, lo que genera retrasos, errores en los registros y dificultades en la toma de decisiones estratégicas. Esta situación no solo afecta la eficiencia interna, sino también impacta la experiencia del cliente y la rentabilidad del negocio.

El desarrollo de una técnica de administración surge como una solución integral para abordar estos problemas planteados inicialmente. Un software permitirá automatizar

tareas esenciales como la actualización del inventario en tiempo real, la generación de facturación electrónica y el seguimiento de ventas. Además, incluirá herramientas que facilitarán la gestión de proveedores y visitadores médicos, para asegurar un proceso sin problemas ni complicaciones.

La implementación de una solución no solo optimizará sus operaciones, sino que también reducirá la posibilidad de pérdidas económicas debido a errores humanos, como el vencimiento de productos sin un adecuado control o la disponibilidad de un medicamento supuestamente dentro del registro del inventario. Adicionalmente, al mejorar la organización de la información y la accesibilidad a los datos clave, se fortalecerá la capacidad de la farmacia para tomar decisiones informadas y estratégicas.

Los objetivos de este informe son:

- Enlistar los requisitos funcionales identificados para que el sistema se acomple a las necesidades de cada usuario.
- Describir, por medio de diagramas, el funcionamiento planteado general del código del software, tomando en cuenta clases, paquetes, clases persistentes y entidades-relaciones de la base de datos.
- Determinar la tecnología a utilizar, por medio de comparaciones de frameworks, para el futuro desarrollo del frontend y backend del proyecto.

Proceso de prototipado

Nota: Se le consultó a Erick e indicó que la sección de prototipado se podría copiar directamente del corte anterior. Eso fue lo que se realizó a continuación:

El proceso inicial de prototipado se centró en definir las pantallas clave del software para asegurar que se ajuste a las necesidades específicas del negocio. Se priorizó la creación de interfaces intuitivas que permitieran la clasificación de usuarios, propuestas para el formato de ventas, inventarios, cierre de caja, agendar visitas, entre otros. En esta fase se realizó un prototipo inicial en el cual todos los participantes del equipo contribuyeron con propuestas e ideas a considerar previo a la entrevista de retroalimentación. Se utilizó la herramienta de “Canva” para tener un registro de todas las fases y que la reestructuración fuera lo más eficiente posible.

Enlace al primer prototipo:

https://www.canva.com/design/DAGgBgKmcag/nTZ6IOBIStxdf9MJOIKTPA/edit?utm_content=DAGgBgKmcag&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Para el segundo prototipo se tuvieron que hacer varios cambios en las diversas pantallas mostradas debido a la falta de datos, o bien, para mejorar la interacción que se da entre el cliente y la pantalla. Fue por esto por lo que en la pantalla de administradora tuvimos que darle acceso a que ella fuese capaz de realizar ventas, ver el calendario de visitas médicas, aprobar las diversas acciones que el dependiente pueda realizar, etc.

Por otro lado, se tuvo que modificar la pantalla de ventas para que fuera más clara junto a la del inventario, para que se pudiese ver el laboratorio al que pertenecía el producto. Se le cambió la dinámica de la pantalla de contador a una donde tendría que pulsar botones para descargar los archivos y negarles acceso a ciertos procesos.

Enlace al segundo prototipo:

https://www.canva.com/design/DAGgH2sWcWU/cBATmaIURutZrs_drWOj9w/edit?utm_content=DAGgH2sWcWU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Luego volver a realizar los cambios con el segundo prototipo, este volvió a ser enviado para ser revisado. En este parte, la cantidad de correcciones disminuyó. Los principales cambios se dieron en el diseño de inicio de sesión, ya que en este se debía incorporar el logo y los colores de la farmacia. Se ajustaron las opciones de inventario, permitiendo la actualización de precios y la gestión de existencias por laboratorio en vez de lote. También se realizaron correcciones en funcionalidades existentes y la incorporación de nuevas opciones.

Enlace al tercer prototipo:

https://www.canva.com/design/DAGfeJj8rpA/PJx6Kt946faX00foHM-Jeg/edit?utm_content=DAGfeJj8rpA&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Sección Análisis

Requisitos funcionales: Al analizar e interpretar las historias de usuario, se identificaron cuáles son los requerimientos que los usuarios del sistema necesitan y quisieran tener en el mismo, para así tener una experiencia satisfactoria y eficiente al momento de interactuar con el software. A continuación, se indican los requisitos funcionales encontrados, con su respectiva historia asociada.

- **Historias de usuario y funcionalidades**

- Como administradora, quiero tener toda la información de mi empresa asequible de manera sencilla, para tomar decisiones que impacten a nivel empresarial.
  Requisitos funcionales identificados:
  - Gestión de cierre de caja
  - Visualizar los movimientos realizados de inventario y compra-venta (en ambos locales)
  - Sistema de facturación para registrar las transacciones ocurridas en el negocio
  - Acceso total a todos los documentos indispensables de los otros perfiles
  - Acceso total a todos los procesos de administración de dependientes
  - Visualizar un resumen por local de la empresa
  - Generar recordatorios para comprar medicamentos, según falta de existencia o momento del año

- Como dependienta (servicio al cliente), quiero tener organizados sencillamente los productos al checar el inventario y realizar una venta, para poder gestionar eficazmente los productos y realizar mejor las ventas.
  Requisitos funcionales identificados:
  - Revisión de inventario en tiempo real (según local atendido)
  - Realización de venta en tiempo real
  - Actualización automática de venta-inventario
  - Modificación de precios a causa de descuentos puntuales
  - Capacidad de subir recetas, para ciertos medicamentos que lo requieran
  - Filtros para los medicamentos, tanto en inventario como en ventas
  - Seguimiento de fecha de vencimiento de medicinas
  - Funcionalidad de subir sus documentos personales sanitarios
  - Editar y eliminar productos, con solicitud de autorización

- Como contador, quiero recibir el informe financiero lo más organizado posible de parte de la empresa, para poder optimizar mi tiempo a la hora de hacer mi trabajo.
  Requisitos funcionales identificados:
  - Tener perfil que únicamente dé acceso a los documentos necesarios de contaduría
  - Funcionalidad de descarga de los documentos
  - Funcionalidad de subida de documento financiero actualizado y aprobado por parte del contador
  - Registro de procedimientos realizados al momento de realizar su tarea

- Como visitador médico (proveedor), quiero tener un canal de comunicación seguro, confidencial y eficiente con mis clientes, para poder ahorrar tiempo de negociación al momento de realizar mis visitas y ventas, así como preservar la confidencialidad de precios de mi farmacéutica/laboratorio.
  Requisitos funcionales identificados:
  - Creación de perfil de proveedor, con apartado para subir el catálogo
  - Funcionalidad de calendario para proponer fechas tentativas de visita
  - Registro y actualización automática para el ingreso de nuevos medicamentos
- Como química farmacéutica, quiero tener la documentación de la empresa, así como todos los registros obligatorios, actualizados y en orden, para poder optimizar el tiempo a la hora de renovar la vigencia de la licencia sanitaria.
  Requisitos funcionales identificados:
  - Poseer perfil que únicamente dé acceso a los documentos necesarios de control sanitario
  - Funcionalidad de descarga de los documentos
  - Funcionalidad de subida de documento de licencia sanitaria por parte del químico farmacéutico
  - Acceso completo a la información sanitaria de los empleados de la empresa

Clases preliminares

i. **Diagrama de clases:**

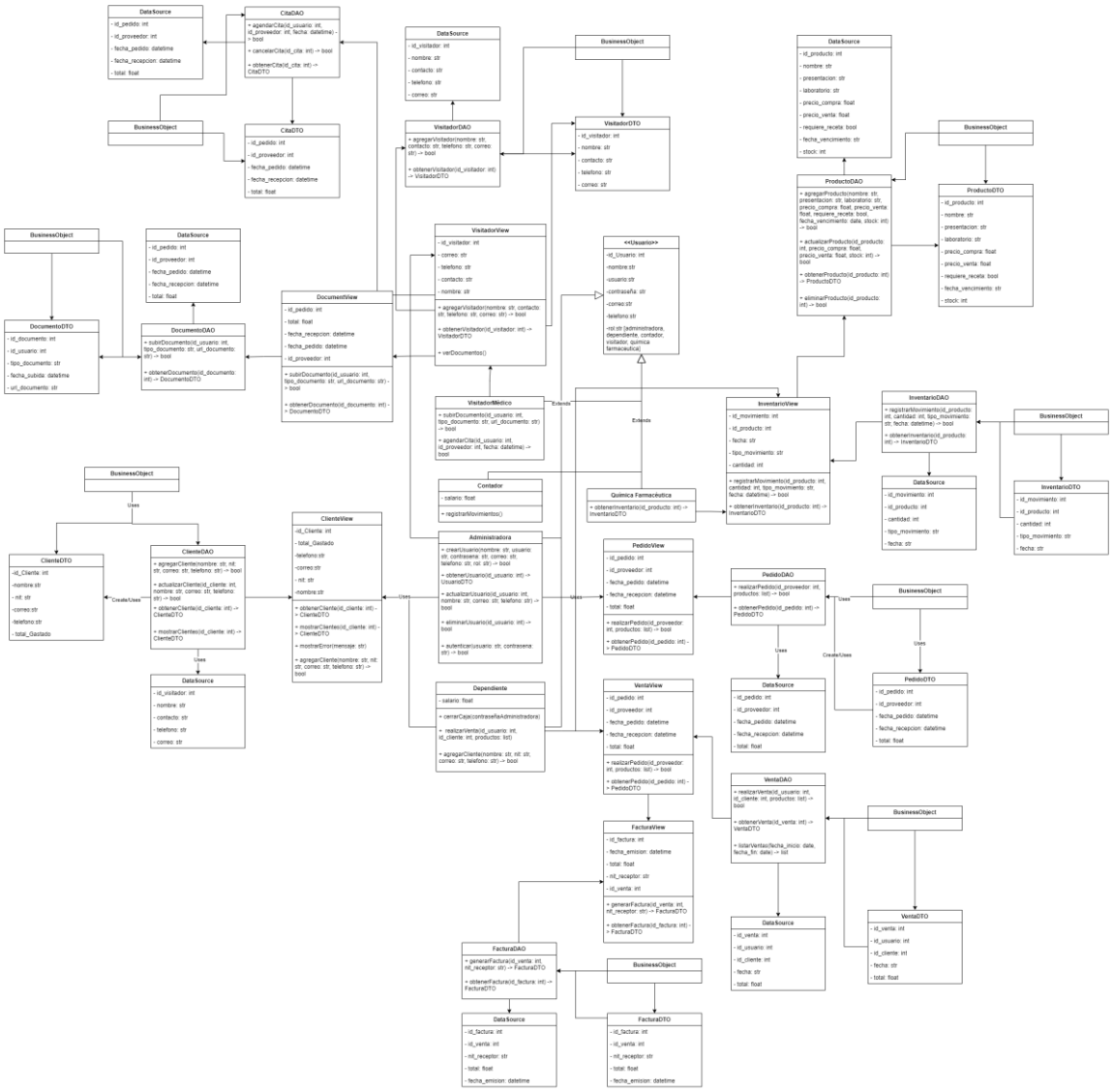


Figura 1. Diagrama de clases, patrón de diseño DAO y DTO.

[Link al diagrama de clases]

ii. Descripción diagrama de clases:

Debido a que se considera que, en el proyecto, se estará consultando, de manera continua, la base de datos, se evaluará el uso del patrón de diseño Data Acces Object (DAO). Según GeeksforGeeks (2024a), este patrón de diseño es una forma de organizar el código para gestionar la comunicación entre su programa y una base de datos.

Por otro lado, el patrón Data Transfer Object (DTO) será implementado para facilitar la transferencia de datos entre procesos, por ejemplo, cuando se registre un nuevo producto.

Las clases que deberían implementar estos patrones de diseño deberán ser las siguientes:

- Cliente.
- Documento.
- Visitador Médico.
- Cita.
- Producto.
- Inventario.
- Pedido.
- Venta.
- Receta.
- Factura.

Debido a que en la fase de prototipado se determinó que dichas clases deberán ser almacenadas en la base de datos, una acción importante para las diferentes tareas que los actores llevan a cabo en su trabajo.

iii. Diagrama de paquetes:

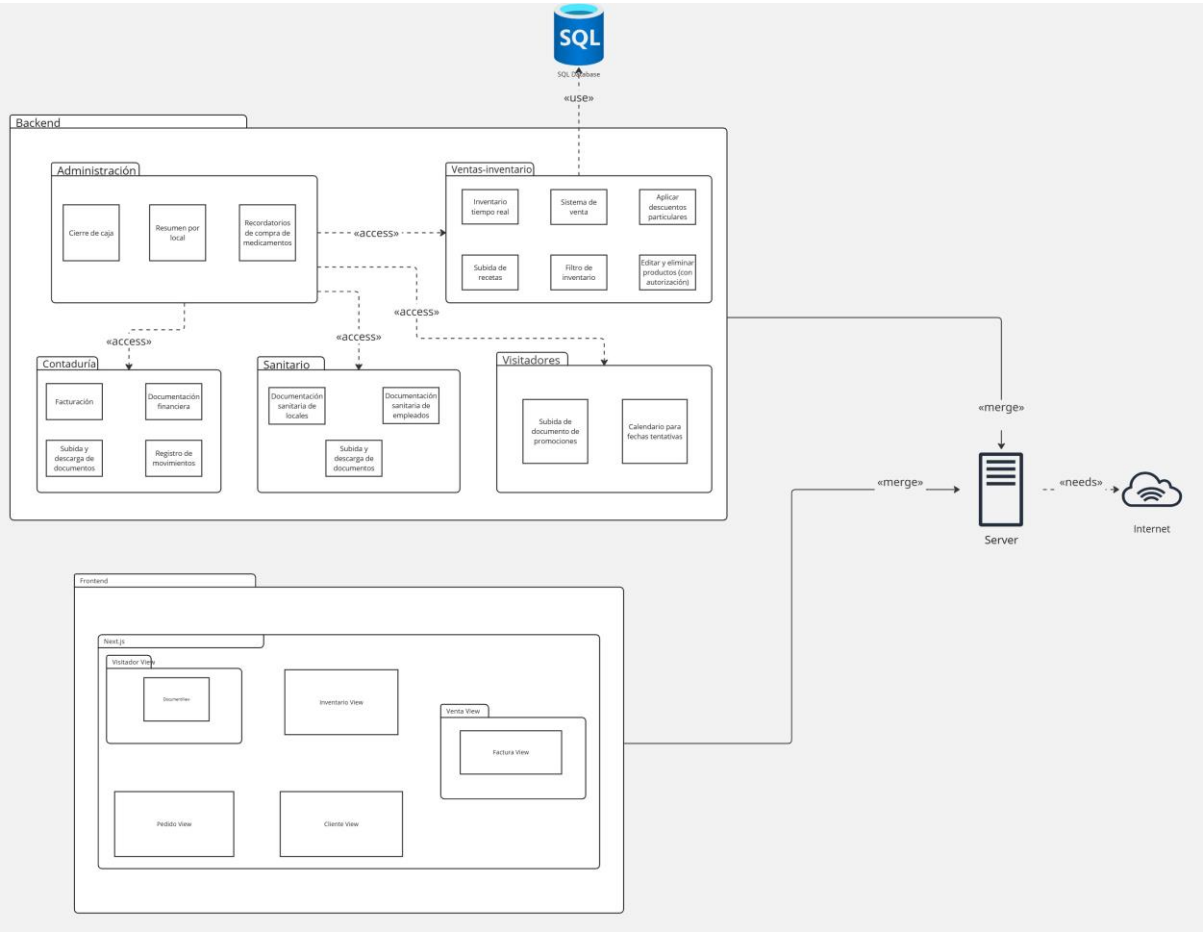


Figura 2. Diagrama de Paquetes

Link al diagrama de paquetes

iv. **Descripción diagrama de paquetes:**

En el diagrama, se tienen 4 elementos centrales e importantes: Paquete backend, paquete frontend, base de datos y servidor.

El paquete de backend es el más extenso. Este contiene todos los subsistemas de paquetes, los cuales están divididos según el departamento de pertenencia de cada rol de usuario, ya que cada uno de estos roles debe mantener cohesión únicamente con su tipo de actividad. Estos subsistemas se dividen en Administración, Ventas-inventario, Contaduría, Sanitario y Visitadores. En cada uno de estos, hablando de manera general, se colocaron los diversos requisitos funcionales previamente identificados que cada usuario requiere, para que estas funcionalidades estén relacionadas únicamente con el departamento al cual pertenece cada rol de los usuarios. Además, se observa cómo el paquete de administrador tiene acceso al resto de paquetes, pues el control general del sistema es uno de los requisitos funcionales identificados en el rol de Administrador. Agregado a esto, se determinó que el paquete de Ventas-inventario es el que debe mantener conexión y comunicación con la base de datos, pues este paquete la utiliza para realizar sus procesos.

El paquete de frontend contiene el framework en el que estará basado todo el frontend (Next.js). Luego, dentro de este, se encuentran todas las vistas (previamente detalladas en el diagrama de clases) que se mostrarán como pantallas de interfaz gráfica para los diferentes roles de usuario.

Luego, se tiene la base de datos, la cual, como se indicó, está relacionada únicamente con el paquete de Ventas-inventario, pues es usada por este. Esta poseerá formato de SQL, ya que, para este proyecto, se utilizará una base de datos relacional.

Finalmente, se tiene el server, el cual manejará las solicitudes de los usuarios, que se estarán realizando en el frontend, con sus respuestas de funciones específicas, que serán procesadas en el backend. Se resalta que el server necesita tener conexión a internet para funcionar.

Clases persistentes

i. **Diagrama de clases persistentes:**

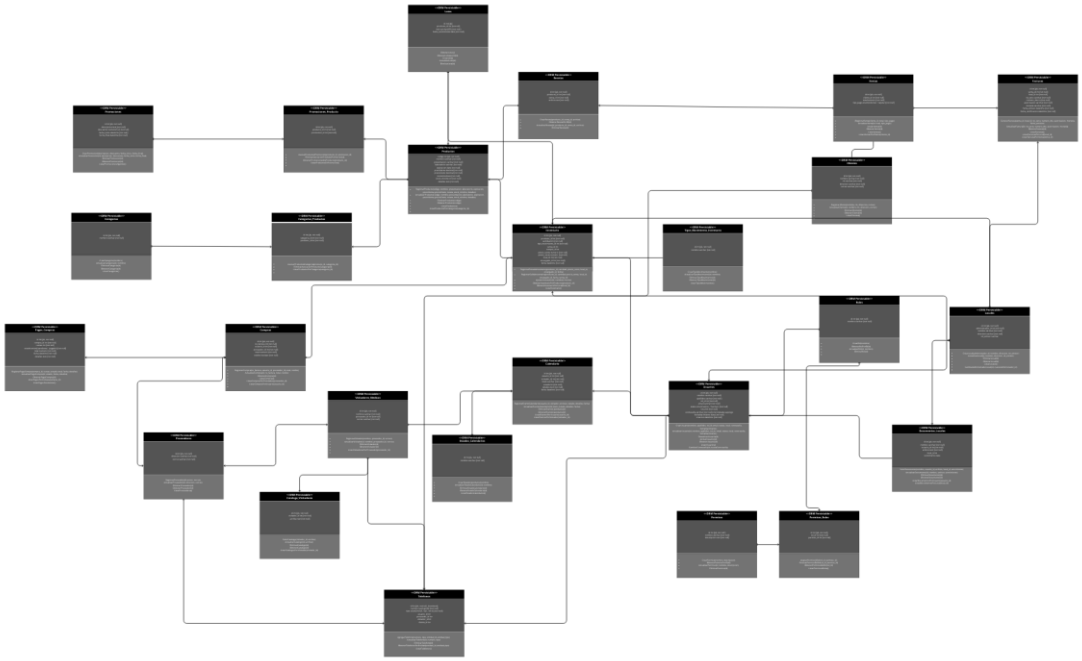


Figura 3. Diagrama de clases persistentes

Enlace al diagrama: https://www.canva.com/design/DAGh9eZ08yc/2hGCvXBloLp-j53-hUoFBw/view?utm_content=DAGh9eZ08yc&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h423b0019e7

El diagrama de clases persistentes para este proyecto, el cual es un ERP para farmacias, define cómo se va a mantener la persistencia en las clases como Usuarios, Lotes, Ventas, Inventario, Proveedores, Clientes. Implementando las funciones CRUD que necesitamos como: crearUsuarios, ActualizarProductos, RegistrarVentas, entre otras. Estas funciones van a permitir administrar todas las reglas del negocio y mantener una trazabilidad de los productos.

ii. Diagrama Entidad Relación:

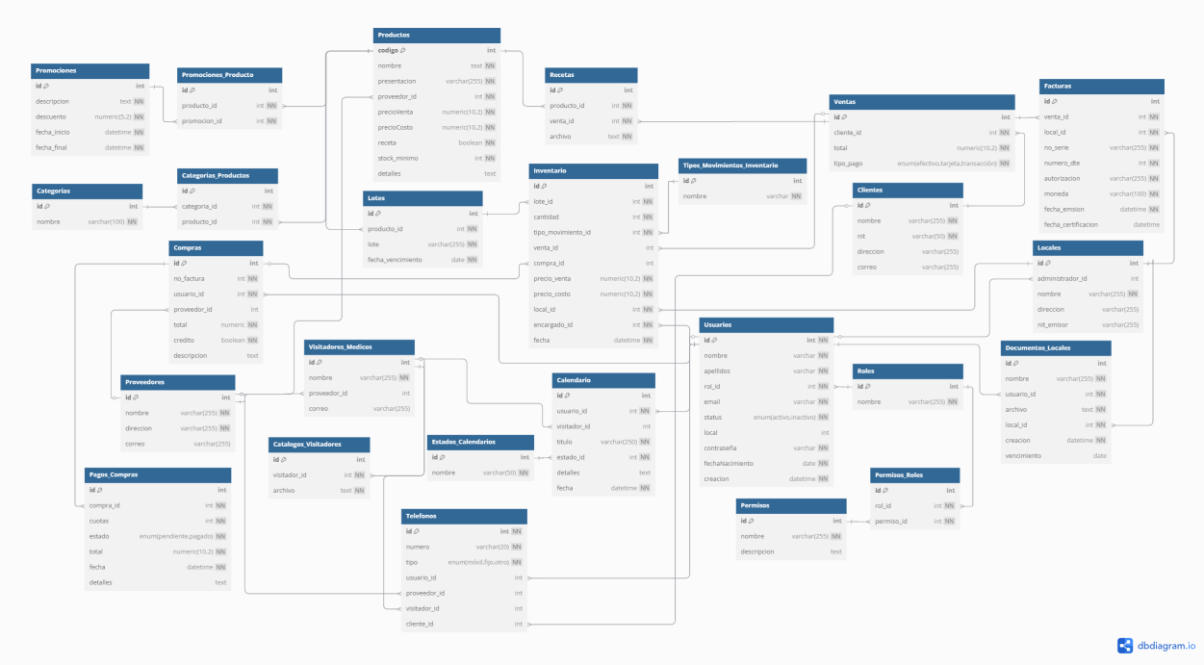


Figura 4. Diagrama Entidad-Relación

Enlace del diagrama: https://dbdiagram.io/d/Software-67d0dc5e75d75cc844badca0

Para el almacenamiento y persistencia de datos en el proyecto, se decidió utilizar una base de datos relacional debido a sus múltiples beneficios. Gracias a la estructura de claves primarias y foráneas, se garantiza la consistencia e integridad de los datos, además de facilitar las referencias entre la información sin necesidad de duplicarla.

Para optimizar el diseño, se aplicaron las fases de normalización en la mayoría de los casos. Sin embargo, en tablas como Inventario y Teléfonos, se optó por separar la información en más atributos, para simplificar las consultas y mejorar el rendimiento en las solicitudes frecuentes. Otro gran beneficio es la facilidad que ofrece SQL para realizar consultas eficientes. Dado que la naturaleza del negocio requiere actualizaciones constantes de información en tiempo real, el uso de consultas optimizadas es fundamental. Finalmente, la escalabilidad es una ventaja clave, ya que permite agregar nuevas entidades o modificar relaciones sin afectar la estructura general de la base de datos, garantizando así su evolución y adaptabilidad a futuros cambios.

Selección de tecnología a utilizar (estimaciones, lenguajes, frameworks y DBMS):

1. **Cantidad de usuarios del sistema**
   Considerando que operan máximo 5 a 6 usuarios (dependientes en turno, contador, jefa, química farmacéutica y visitadores médicos), se estima que este será el número máximo de usuarios activos.

2. **Tiempo de respuesta**
   El sistema debe responder a las solicitudes en menos de 6 segundos, cumpliendo con los requisitos de rendimiento y eficiencia, incluso en horarios de mayor actividad (cierre del día, revisión de inventarios, etc.).

3. **Usuarios trabajando simultáneamente**
   Se espera un uso concurrente promedio de 2 a 3 usuarios durante las horas de mayor flujo de clientes.

4. **Escalabilidad**
   Promedio diario de ventas: 700 ventas por día.
   Promedio anual: 700 x 365 ≈ 255,500 ventas anuales.
   Volumen de transacciones estimadas en 5 años: 255,500 x 5 ≈ 1,277,500 ventas.
   Considerando los datos, cada venta puede estar relacionada con uno o más productos, clientes, método de pago, y documentos (como recetas), por lo que se espera manejar:
   - Aproximadamente 1.2 millones de registros de ventas en 5 años.
   - Historial de productos, clientes, usuarios, proveedores, documentos adjuntos.

   Estimación de almacenamiento:
   Cada venta con sus datos y documentos podría ocupar entre 1 y 5 KB dependiendo si incluye recetas o detalles adicionales.
   Estimación conservadora de espacio total:
   1,277,500 ventas x 5 KB ≈ 6.4 GB solo en ventas.
   Sumando otros registros y respaldos, se estima un total de 8 a 12 GB de base de datos en 5 años.

5. **Disponibilidad de la Información**
   El sistema debe estar 100% disponible durante el horario de atención al público (aproximadamente 12 horas diarias), con acceso rápido a inventarios, ventas y clientes.
   Administradores deben tener acceso fuera del horario para reportes o revisiones.

6. **Confiabilidad, tolerancia a fallos, replicación**
   Por la importancia de los datos:
   Respaldos automáticos diarios y semanales. Almacenamiento en la nube o servidor externo para recuperación ante fallos. Cifrado de información sensible, incluyendo datos personales y recetas.

Reformulación Requisitos no Funcionales

Requisito no funcional	Categoría	Forma en que se medirá su cumplimiento
Generación de factura para cada venta	Seguridad y privacidad	Se generará un PDF y XML por cada venta, verificando que se almacenen correctamente y sean accesibles según los permisos de usuario.
Manuales de uso por rol de usuario	Ayudas y documentación en línea	Se medirá la cantidad de documentos creados para cada usuario y su facilidad de acceso en la interfaz del sistema.
Respaldo automático de la base de datos	Confiabilidad	Se implementará un script en Node.js que genere respaldos automáticos cada X horas. Se medirá la cantidad de respaldos generados en un mes y su integridad.
Tiempo de respuesta del sistema < 6 segundos	Rendimiento	Con una herramienta como DevTools medir el tiempo de respuesta.
Colores diferenciados por tipo de usuario	Apariencia e interfaz externa	Conteo de ocasiones en las que un usuario identificó estar en la cuenta equivocada.
Colores de prioridad para los productos	Apariencia e interfaz externa	Se verificará que cada producto tenga el color correcto según su fecha de vencimiento y fecha actual, asegurando una correcta identificación visual.
Soporte para múltiples solicitudes simultáneas	Rendimiento	Se medirá con pruebas de estrés verificando que Node.js maneje múltiples requests sin afectar el rendimiento.

Uso de la línea gráfica de la farmacia	Apariencia e interfaz externa	La aplicación usará la paleta de colores, logo y tipografía de la farmacia, asegurando consistencia visual con la marca.
Cifrado de recetas y datos personales	Seguridad y privacidad	Se implementará cifrado de datos en Node.js asegurando que los datos sensibles solo sean accesibles por usuarios autorizados.
Diseño responsivo en múltiples dispositivos	Usabilidad	Se debe medir en diferentes dispositivos para ver cómo se adapta la aplicación.
Interfaz intuitiva y fácil de navegar	Interfaz interna	Se medirán métricas de UX como clicks hasta la acción deseada y feedback de usuarios en pruebas piloto.
Optimización del uso de CPU y RAM	Rendimiento	Se analizará el consumo de recursos con herramientas como Node.js.
Gestión de documentación legal	Legales	Se implementará un módulo en Node.js que registre y notifique sobre el vencimiento de documentos como la licencia sanitaria, patente de comercio y tarjeta de salud.

Tecnologías Consideradas

Tecnología	Descripción
Node.js	Entorno de ejecución de JavaScript en el servidor, basado en el motor V8 de Chrome. Permite el desarrollo escalable y asincrónico.
.NET	Plataforma de desarrollo de Microsoft que permite construir aplicaciones seguras y escalables con C#.
NestJS	Framework de Node.js basado en TypeScript, con arquitectura modular e inspirada en Angular, ideal para APIs escalables.
Express	Framework minimalista para Node.js que facilita la creacioón de APIs con sintazis sencilla y flexible.

Fresh	Framework moderno para Deno, que evita la necesidad de build y optimiza el tiempo de ejecución.
Oak	Middleware para Deno basado en Koa.js, diseñado para construir aplicaciones web y APIs.
Next.js	Framework de React para frontend, que soporta SSR Y SSG.
Deno	Entorno de ejecución para JavaScript y TypeScript con mejor seguridad que Node.js.
Vite	Herramienta de construcción rápida para frontend, optimizada para desarrollo con ES modules y HMR.
Vue	Framework progresivo de JavaScript para construir interfaces interactivas y SPA.
Angular	Framework robusto basado en TypeScript para aplicaciones empresariales de gran escala.
React	Biblioteca basada en componentes reutilizables para interfaces de usuario dinámicas.

Cuadro Comparativo – Frontend

Característica	**Next.js**	**Vite**	**Vue.js**	**Angular**	**React**
Lenguaje	JavaScript/ TypeScript	JavaScript/ TypeScript	JavaScript	TypeScript	JavaScript
Curva de aprendizaje	Media	Baja	Baja	Alta	Media
Escalabilidad	Alta	Media	Media	Muy alta	Alta
Seguridad	Alta	Media	Media	Alta	Media
Uso de recursos	Medio	Bajo	Bajo	Alto	Medio
Facilidad de migración	Alta	Alta	Media	Baja	Alta
Modularidad	Alta	Media	Alta	Muy alta	Alta
Costo	Bajo	Bajo	Bajo	Alto	Bajo

Cuadro Comparativo – Backend

Característica	**Node.js**	**.NET**	**NestJS**	**Express**	**Fresh**	**Deno**
Lenguaje	JavaScript / TypeScript	C#	TypeScript	JavaScript	TypeScript	TypeScript
Curva de aprendizaje	Moderada	Alta	Media	Baja	Media	Alta
Escalabilidad	Alta	Muy alta	Muy alta	Media	Media	Alta
Seguridad	Media	Alta	Alta	Media	Alta	Muy alta

Uso de recursos	Bajo	Alto	Bajo	Bajo	Muy bajo	Medio
Facilidad de migración	Alta	Baja	Media	Alta	Baja	Baja
Modularidad	Media	Alta	Muy alta	Media	Alta	Media
Costo	Bajo	Alto	Bajo	Bajo	Bajo	Bajo

Ventajas y Desventajas - Frontend

Tecnología	Ventajas	Desventajas
Node.js	Código abierto, gran comunidad, escalabilidad, ideal para apps en tiempo real, eficiente en recursos.	No tipado por defecto (a menos que se use TypeScript), menor seguridad que .NET, dificultad en debugging en proyectos grandes.
.NET	Seguridad alta, compatibilidad con aplicaciones empresariales, robusto, optimización en sistemas grandes.	Costo de licencias, curva de aprendizaje alta, alto consumo de recursos.
NestJS	Modularidad avanzada, arquitectura limpia, basado en TypeScript, ideal para microservicios.	Mayor curva de aprendizaje que Express, más sobrecarga en comparación con Express.js.
Express.js	Ligero, fácil de aprender, comunidad grande, integración sencilla con otros módulos.	No modular por defecto, requiere middleware adicional, menor seguridad que NestJS.
Fresh	Bajo uso de recursos, excelente para renderizado en servidor, optimizado para Deno.	Ecosistema en crecimiento, menos soporte que Node.js, menos madurez en la comunidad.
Deno	Seguridad nativa, mejor manejo de permisos, sin dependencias no controladas.	Compatibilidad limitada, curva de aprendizaje más alta, ecosistema más pequeño que Node.js.

Ventajas y Desventajas - Frontend

Tecnología	Ventajas	Desventajas

Next.js	Soporte para SSR y SSG, rendimiento optimizado, escalabilidad alta, comunidad grande.	Mayor complejidad en configuración; requiere más backend.
Vite	Rápido en desarrollo, optimización eficiente, fácil configuración.	No soporta SSR directamente, requiere configuración adicional para grandes proyectos.
Vue.js	Fácil de aprender, progresivo, buen rendimiento en proyectos medianos.	Menor comunidad que React, menos uso en grandes empresas.
Angular	Muy robusto, altamente escalable, seguridad avanzada.	Curva de aprendizaje alta, más pesado en comparación con otras opciones.
React	Gran comunidad, componentes reutilizables, versátil.	No ofrece SSR nativo (sin Next.js), más boilerplate.

Comparación Bases de Datos

Característica	**MySQL**	**PostgreSQL**	**MongoDB**	**SQLite**
Tipo de Base de Datos	Relacional	Relacional	No Relacional	Relacional
Escalabilidad	Horizontal/Vertical	Horizontal/Vertical	Horizontal	Limitada
Rendimiento	Alto	Alto	Alto	Bajo
Integridad de Datos	Soporta claves foráneas	Soporta claves foráneas	No soporta claves foráneas	Soporta claves foráneas
Costo	Medio (Costo bajo en versión comunitaria, pero puede aumentar con soporte comercial)	Medio (Gratis, pero puede requerir más recursos)	Bajo (Licencia libre, pero alto costo de infraestructura)	Bajo (Totalmente gratuito, ideal para aplicaciones ligeras)

Ventajas y Desventajas Bases de Datos

Base de Datos	**Ventajas**	**Desventajas**
MySQL	Facilidad de uso. Rápido en operaciones de lectura.	Limitado en funciones avanzadas Menos flexible para datos no estructurados.

	Gran comunidad y soporte.	
PostgreSQL	Altamente extensible y flexible. Soporte completo de ACID. Ideal para consultas complejas.	Rendimiento más lento en operaciones simples que MySQL. Requiere más recursos de hardware.
MongoDB	Escalabilidad horizontal. Flexible con datos no estructurados. Buena para grandes volúmenes de datos.	No soporta transacciones ACID completamente. No es adecuado para aplicaciones con datos altamente estructurados.
SQLite	Ligero y fácil de usar. Ideal para aplicaciones locales y embebidas. No requiere un servidor.	No es adecuado para aplicaciones de gran escala. Soporte limitado para concurrencia y transacciones complejas.

Decisión Final

Para el backend, se eligió Node.js con NestJS por su escalabilidad, modularidad y facilidad de mantenimiento, permitiendo manejar múltiples solicitudes simultáneas y asegurar un rendimiento óptimo para procesar entre 500 y 1000 ventas diarias, además de facilitar la integración de autenticación JWT y encriptación de datos.

Para el frontend, se optó por Next.js con Vite, lo que garantiza una interfaz responsiva y optimizada en rendimiento, con tiempos de respuesta menores a 6 segundos, y acelera el desarrollo gracias a la velocidad de compilación de Vite.

La base de datos seleccionada fue PostgreSQL por su estabilidad, soporte para grandes volúmenes de datos y seguridad avanzada, garantizando la integridad de la información.

Bitácora trabajo con el cliente (prototipos):

BITÁCORA CLIENTES

Fecha: 21 de febrero de 2025 Presencial - 10pm

Usuario(s): Dependiente de farmacia Eunice Vargas

Proyecto: ERP para gestión de Farmacia Econofarma

Contenidos abordados

Se abordo el diseño del primer prototipo para el ERP de la farmacia, tocando temas como la navegación, el diseño y funciones especificas que debería tener como dependienta además de factores que no siempre aplican como los descuentos.

Lo que se observo	Imagen reunión
Al momento de ir mostrando el documento que contenía las pantallas se noto la confusión de como iba a pasar de una pantalla a otra, y dentro de las pantallas el acceso a funciones especificas como modificar el NIT.	

Conclusiones

Se debe mejorar la interfaz de la aplicación para que sea más intuitiva

Hay que mantener un diseño simple para evitar la confusión debido a que el usuario no domina el uso de dispositivos inteligentes.

Puntos importantes

La farmacia es una farmacia social por lo que los precios pueden variar según diferentes condiciones.
Un menú que permita el acceso a todas las funciones es importante.
Se deben modificar fácilmente los datos del cliente.

Figura 5. Entrevista 1 – prototipo 1

BITÁCORA CLIENTES

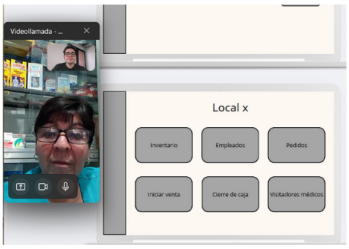
Fecha: 22 de febrero de 2025
Vía zoom - 3pm

Usuario(s): Administradora
Lesvia Mejía

Proyecto: ERP para gestión de Farmacia Econofarma

Contenidos abordados

Se abordo el diseño del primer prototipo para el ERP de la farmacia, tocando temas como los permisos, la creación de usuarios, los accesos que tendrán estos usuarios, los datos que son más importantes y la navegación entre las pantallas.

Lo que se observo	Imagen reunión
Un disgusto con el diseño de las pantallas, además de los accesos que tienen los diferentes usuarios, porque es información confidencial que puede afectar a la farmacia de forma negativa.	

Conclusiones

Hay que comprender que accesos y a que documentos debería poder acceder cada usuario.

Se debe mejorar la navegación y corregir campos dentro las pantallas que no son necesarios o se les dio mayor importancia que la que se debía.

Puntos importantes

Hay que tomar en cuenta que no todas las personas tienen familiaridad con una computadora o dispositivo inteligente y menos con un software por lo que debe ser amigable.

Figura 6. Entrevista 2 – prototipo 1

BITÁCORA CLIENTES

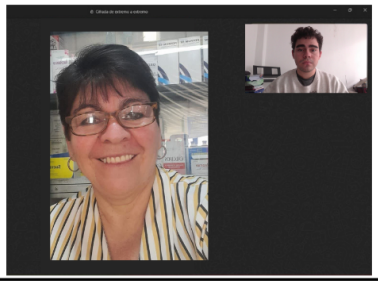
Fecha: 24 de febrero de 2025
Videollamada - 12pm

Usuario(s): Administradora
Lesvia Mejía

Proyecto: ERP para gestión de Farmacia Econofarma

Contenidos abordados

Tomando en cuenta sus comentarios se abordó el segundo prototipo del proyecto temas abordados interfaz gráfica, algunas funciones extra e información complementaria de las pantallas, otras mejoras hacía la navegación en pantallas y al user friendly.

Lo que se observo	Imagen reunión
Se observó confusión en ciertas pantallas aún, sobre todo en la comunicación de la sucursal de la farmacia, la información faltante es importante por lo que se debería comprender aún más la lógica del negocio.	

Conclusiones

Hay que respetar los datos de algunas pantallas ya que proporcionan información importante.

Hay que estructurar bien la lógica de la relación entre las sucursales de la farmacia y asignar bien los permisos de acceso entre ellas.

Puntos importantes

Diferenciar los productos según el laboratorio, es decir productos pueden tener el mismo nombre y la misma presentación pero son de diferentes laboratorios.
Se acordó que el software debe permitir la actualización de la licencia sanitaria para una mejor gestión y control

Figura 7. Entrevista 1 – prototipo 2

BITÁCORA CLIENTES

Fecha: 24 de febrero de 2025
Vía zoom - 11 pm

Usuario(s): Visitador Médico (Andifar)
Estiven Cardona

Proyecto: ERP para gestión de Farmacia Econofarma

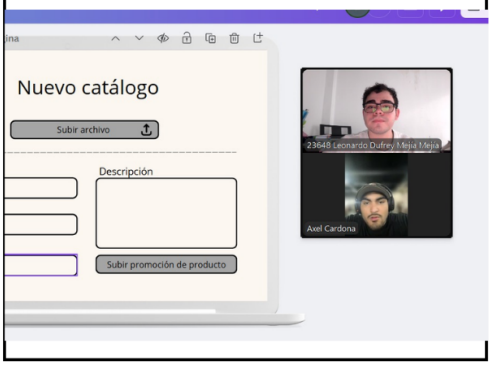
Contenidos abordados

Basándose en el segundo prototipo, principalmente se abordo si estaba de acuerdo con las pantallas que se le habían asignado, si eran comprensibles y estaba de acuerdo con compartir esa información, la estructura de las pantallas y la navegación entre ellas. Se toco temas de permisos y usuarios.

Lo que se observo

No entendió si se le va a generar un usuario o cómo va a dar la visita la primera vez, considera que de verdad para que sea útil debería poderse hacer sin usuario porque así podría conectar a sus colegas para iniciar su comunicación con la farmacia.
Al mismo tiempo quisiera en vez de escribir descripción o algo que sea un comentario hacía la farmacia, al final de cuentas para vender primero se debe ganar al cliente, el programa funcionaría para una primera interacción más rápida pero considera que actualmente entorpece más el trabajo que beneficiarlo.

Imagen reunión



Conclusiones

El usuario es algo que se debe manejar, ya que obviamente la farmacia no lo conocería cuando sea la primera vez.

Se debe tomar en cuenta que debe agilizar el proceso del visitador y la farmacia aumentando la comunicación con ella y no disminuyéndola.

Puntos importantes

Se debe designar si para que un visitador pueda mandar un archivo debe tener un usuario o sea de uso libre.

¿Cómo determinar el uso libre de la aplicación?

Figura 8. Entrevista 2 – prototipo 2

GitHub: https://github.com/dannyrmrz/ERP.git

Presentación de clase: https://www.canva.com/design/DAGiGuhkhNM/Q9udjCoLTF1wK0N0G0PpaQ/edit?utm_content=DAGiGuhkhNM&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Documento (editable): https://uvggt-my.sharepoint.com/:w:/g/personal/roj23813_uvg_edu_gt/EcgcCnNIRF9Bio1kzvVwIY0B5sJ1bHb5HfMTMrFLo74Hhw?e=UBOWW7

Informe de gestión del tiempo (grupal):

Integrante	Tarea a realizar	Inicio	Fin
Todos	Presentación Corte 3	18/03/2025 12:00 p.m.	19/03/2025 2:00 p.m.
Renato Rojas	Requisitos funcionales	12/03/2025 12:20 p.m.	12/03/2025 1:30 p.m.
Renato Rojas	Diagrama de paquetes	15/03/2025 5:34 p.m.	15/03/2025 7:00 p.m.
Melisa Mendizabal	Diagrama Entidad Relación	11/03/2025 7:00 p.m.	18/03/2025 10:20 p.m.
María Girón	Diagrama de clases	16/03/2025 6:00 p.m.	16/03/2025 7:00 p.m.
Leonardo Mejía	Diagrama de clases persistentes	13/03/2025 10:00 a.m.	18/03/2025 5:50 p.m.
Daniela Ramírez	Tecnologías a utilizar	17/03/2025	18/03/2025

Aspectos positivos:

- En esta entrega, la organización fue más eficiente. Como compañeros, nos apoyamos entre todos para dar retroalimentación sobre cada uno de los diagramas creados, con el propósito de mejorarlos.

Aspectos por mejorar:

- La velocidad de trabajo, pues varias de las secciones de las entregas se realizan con tiempo bastante reducido por delante, lo cual da poco margen de revisión y corrección de errores.

Referencias

De Jesus, M. R. (2023, 6 febrero). Data Access Patterns: the Features of the Main Data Access Patterns Applied in Software Industry. *Medium*. https://medium.com/mastering-software-engineering/data-access-patterns-the-features-of-the-main-data-access-patterns-applied-in-software-industry-6eff86906b4e

GeeksforGeeks. (2022, 22 mayo). *Data Transfer Object (DTO) in Spring MVC with Example*. GeeksforGeeks. https://www.geeksforgeeks.org/data-transfer-object-dto-in-spring-mvc-with-example/

GeeksforGeeks. (2024a, enero 31). *Data Access Object(DAO) design pattern*. GeeksforGeeks. https://www.geeksforgeeks.org/data-access-object-pattern/

GeeksforGeeks. (2024b, octubre 27). *Structural design patterns*. GeeksforGeeks. https://www.geeksforgeeks.org/structural-design-patterns/

Lucid Software Español. (2019, 4 febrero). *Tutorial - Diagrama de clases UML* [Vídeo]. YouTube. https://www.youtube.com/watch?v=Z0yLerU0g-Q

Oblancarte. (2018, 10 diciembre). *Data Access Object (DAO) pattern - Oscar Blancarte - Software Architecture*. Oscar Blancarte - Software Architecture. https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/