



Desarrollo de Aplicaciones con Software Propietario DAS 901 G01T

Proyecto Fase 1

- Marcos Daniel Ibáñez Guevara IG243224
- Andrea Marcela Rico Figueroa RF160050
- Frank Alberto Hernández Silva HS171707
- Isaias Alexander Ortega Monterrosa OM161293

Introducción

En el presente informe se les presentará el diseño teórico de un sistema de gestión empresarial desarrollado en la plataforma ASP.NET Core con C# como lenguaje de código, fusionado con una base de datos relacional en MySQL. Este sistema tiene como objetivo principal facilitar el manejo eficiente de la información de hasta 50 empleados y 50 clientes, permitiendo operaciones de creación, lectura, actualización y eliminación (CRUD) de datos esenciales para la organización y la maximización de sus ganancias por medir la eficiencia que el sistema les otorga.

El sistema está orientado a pequeñas y medianas empresas conocidas como PYMES que buscan optimizar sus procesos administrativos mediante una solución tecnológica escalable, accesible y segura. Las herramientas tecnológicas escogidas responden a criterios de rendimiento, soporte, compatibilidad multiplataforma y la posibilidad de integrarse con APIS que permitan ampliar sus funcionalidades en el futuro.

A lo largo del documento se detallan temas como la arquitectura general del sistema, el modelo de datos propuesto, la justificación de las decisiones tecnológicas como criterio solicitado, así como un análisis de los posibles beneficios y limitaciones del proyecto en un entorno real de implementación, ya que justamente es una simulación de lo que sería el sistema en un entorno profesional real.

OBJETIVOS

Objetivo General:

Diseñar y desarrollar un sistema de gestión empresarial utilizando ASP.NET como framework principal, que le dé la posibilidad de administrar de forma eficiente los datos de una organización compuesta por aproximadamente 50 empleados y 50 clientes, pero con el potencial de que sean muchos más, integrando una base de datos MySQL como núcleo para el almacenamiento, consulta y mantenimiento de la información.

Objetivos Específicos:

1. **Establecer una arquitectura funcional y escalable de forma exponencial** que permita gestionar operaciones básicas de tipo CRUD (Crear, Leer, Actualizar y Eliminar), enfocadas en entidades clave como empleados, clientes, servicios y procesos administrativos internos.
2. **Implementar una base de datos relacional en MySQL** estructurada y optimizada para soportar el volumen estimado de registros o más si fuese necesario, garantizando integridad referencial, velocidad en las consultas y facilidad de uso y mantenimiento.
3. **Desarrollar una interfaz web dinámica en ASP.NET** que facilite la interacción del usuario o usuarios, con el sistema de forma intuitiva, permitiendo la administración de la información sin requerir conocimientos técnicos avanzados, sino más bien para entornos básicos.
4. **Aplicar principios de buenas prácticas de desarrollo seguro y eficiente**, incluyendo validaciones en formularios, control de errores y una estructura clara para el consumo de datos mediante APIs.
5. **Adaptación del proyecto a condiciones reales de desarrollo**, teniendo en cuenta posibles restricciones de tiempo, retrasos, presupuesto y recursos humanos, lo que implica el uso eficiente de tecnologías accesibles y de código abierto.

Identificación de Plataforma Tecnológica

Para el desarrollo del sistema de gestión empresarial, se ha optado por utilizar **ASP.NET Core** como plataforma principal de desarrollo, en conjunto con el lenguaje de programación **C#** y el sistema de gestión de bases de datos relacional **MySQL**. Esta combinación tecnológica ha sido seleccionada por su equilibrio entre robustez, escalabilidad, eficiencia y compatibilidad con entornos empresariales.

ASP.NET Core es un framework de desarrollo web de código abierto por lo cual nos facilita en el tema de los costos, es un multiplataforma, desarrollado por Microsoft. Está diseñado para crear aplicaciones modernas, de alto rendimiento y altamente escalables, lo que lo convierte en una excelente opción para sistemas de gestión empresarial que deben manejar múltiples usuarios y operaciones simultáneamente. Además, su estructura modular y su capacidad para implementar APIS RESTFUL facilitan la construcción de servicios backend, que pueden interactuar fácilmente con interfaces de usuario modernas o aplicaciones móviles y da la facilidad a personal no tan capacitado de desenvolverse bien.

El uso de **C#Cronograma detallado con las actividades clave del proyecto, tiempos estimados y recursos asignados para cada fase del desarrollo.** como lenguaje principal garantiza una sintaxis sólida, tipado estático y múltiples herramientas de desarrollo, lo que reduce errores durante el proceso de codificación y permite mantener un código fuente más limpio y organizado. Asimismo, C# Cuenta con una gran comunidad de desarrolladores, documentación extensa y recursos de aprendizaje, lo que facilita el mantenimiento y evolución del sistema en el futuro.

Por otro lado, se ha elegido **MySQL** como sistema de base de datos debido a su fiabilidad, rendimiento comprobado y compatibilidad con ASP.NET Core. MySQL es una base de datos de código abierto ampliamente utilizada en la industria, con soporte para consultas SQL complejas, integridad referencial y una buena capacidad de escalado, especialmente adecuada para estructuras de datos que manejan hasta 500 empleados y 500 clientes, como es el caso del sistema propuesto.

La elección de esta arquitectura tecnológica responde también a criterios de **eficiencia presupuestaria y de tiempo**, considerando que tanto ASP.NET Core como MySQL son tecnologías de libre uso que no requieren licencias costosas para su implementación básica. Esto permite maximizar los recursos disponibles en el proyecto, sin comprometer la calidad ni el rendimiento del sistema.

Además, la interoperabilidad entre ASP.NET Core y MySQL está bien documentada y respaldada por numerosos recursos en línea, lo cual reduce significativamente la curva de aprendizaje y los tiempos de desarrollo. También ofrece flexibilidad para ser desplegado tanto en servidores locales como en servicios de nube como **Microsoft Azure** o **Amazon Web Services**, lo cual deja abierta la posibilidad de escalar el sistema en el futuro si se amplía la base de usuarios.

En resumen, ASP.NET Core con C# MySQL constituye una **plataforma tecnológica moderna, segura, rentable y preparada para el crecimiento**, lo cual la convierte en una

elección estratégica adecuada para el desarrollo del sistema de gestión empresarial propuesto.

Descripción completa del modelo de datos.

El modelo de datos propuesto para el sistema de gestión empresarial desarrollado en ASP.NET Core con base de datos MySQL ha sido diseñado con el objetivo de ofrecer una estructura clara, coherente y escalable. Este modelo permite una administración eficiente de la información relacionada con empleados, clientes, servicios y operaciones comerciales.

El sistema contempla un conjunto de entidades principales que representan los elementos fundamentales del negocio. A continuación, se describen las entidades más relevantes:

- **Empleado:** Contiene información del personal que labora en la empresa. Incluye campos como `ID_Empleado (INT)`, `Nombre (VARCHAR)`, `Apellido (VARCHAR)`, `Fecha de Ingreso (DATE)`, `Puesto (VARCHAR)` y `Departamento ID (INT)`.
- **Cliente:** Almacena los datos de los clientes registrados. Incluye `ID_Cliente (INT)`, `Nombre (VARCHAR)`, `Correo Electrónico (VARCHAR)`, `Telefono (VARCHAR)` y `Direccion (VARCHAR)`.
- **Departamento:** Define las distintas áreas funcionales dentro de la empresa, como Recursos Humanos, Finanzas, IT, etc. Campos: `ID Departamento (INT)`, `Nombre (VARCHAR)`.
- **Factura:** Representa las transacciones realizadas con los clientes. Incluye `IDFactura (INT)`, `Fecha (DATE)`, `Total (DECIMAL)` y `ID_Cliente (INT)`.
- **Producto/Servicio:** Contiene los productos o servicios ofrecidos. Campos: `ID_Producto (INT)`, `Nombre (VARCHAR)`, `Precio (DECIMAL)`, `Descripción (TEXT)`.
- **Detalle Factura:** Tabla intermedia para representar la relación muchos a muchos entre Facturas y Productos. Contiene `ID_Detalle (INT)`, `IDFactura (INT)`, `ID_Producto (INT)`, `Cantidad (INT)`, `Subtotal (DECIMAL)`.
- **Usuario:** Para el acceso al sistema, incluyendo roles. Campos: `ID_Usuario (INT)`, `NombreUsuario (VARCHAR)`, `Contraseña (VARCHAR)`, `Rol (VARCHAR)`.

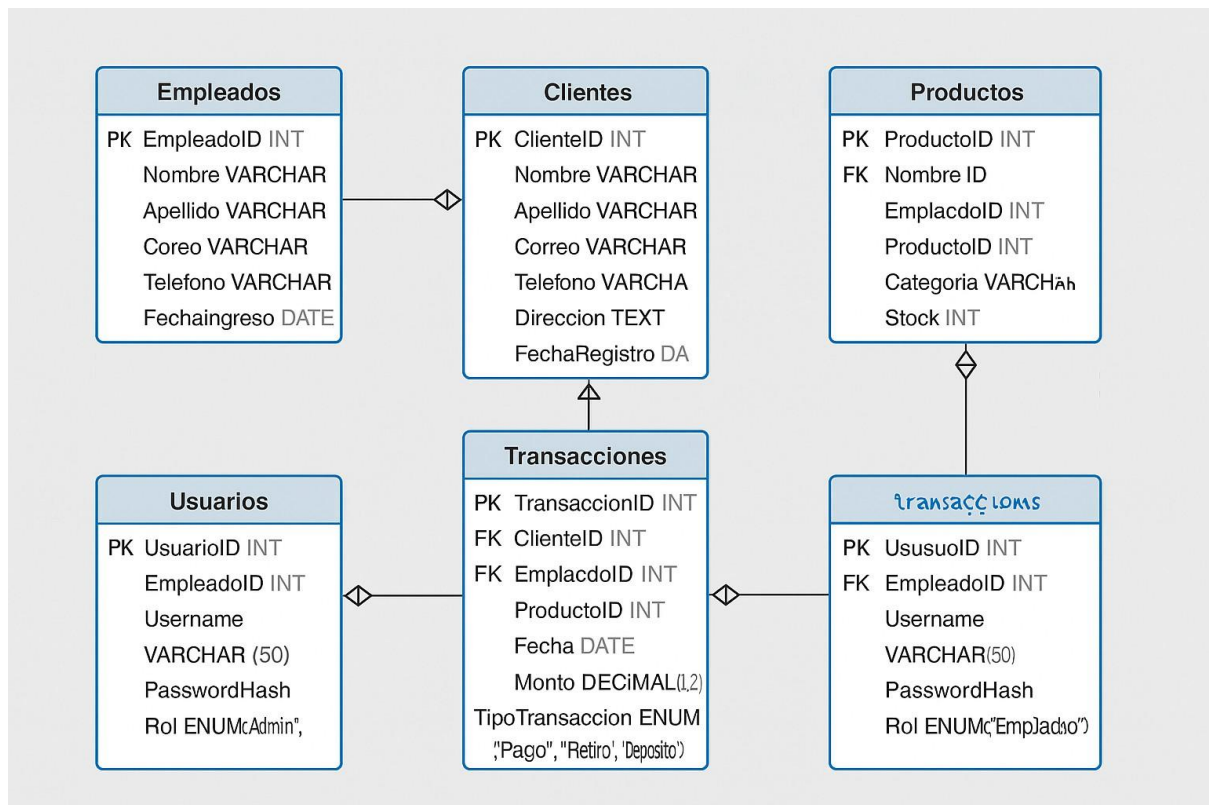
Las relaciones entre estas entidades están estructuradas de la siguiente manera:

- Un **Departamento** puede tener muchos **empleados**.
- Un **Cliente** puede generar múltiples **Facturas**.
- Una **Factura** puede incluir varios **Productos**, y un **Producto** puede estar en varias facturas, lo que se resuelve con la entidad intermedia **Detalle Factura**.

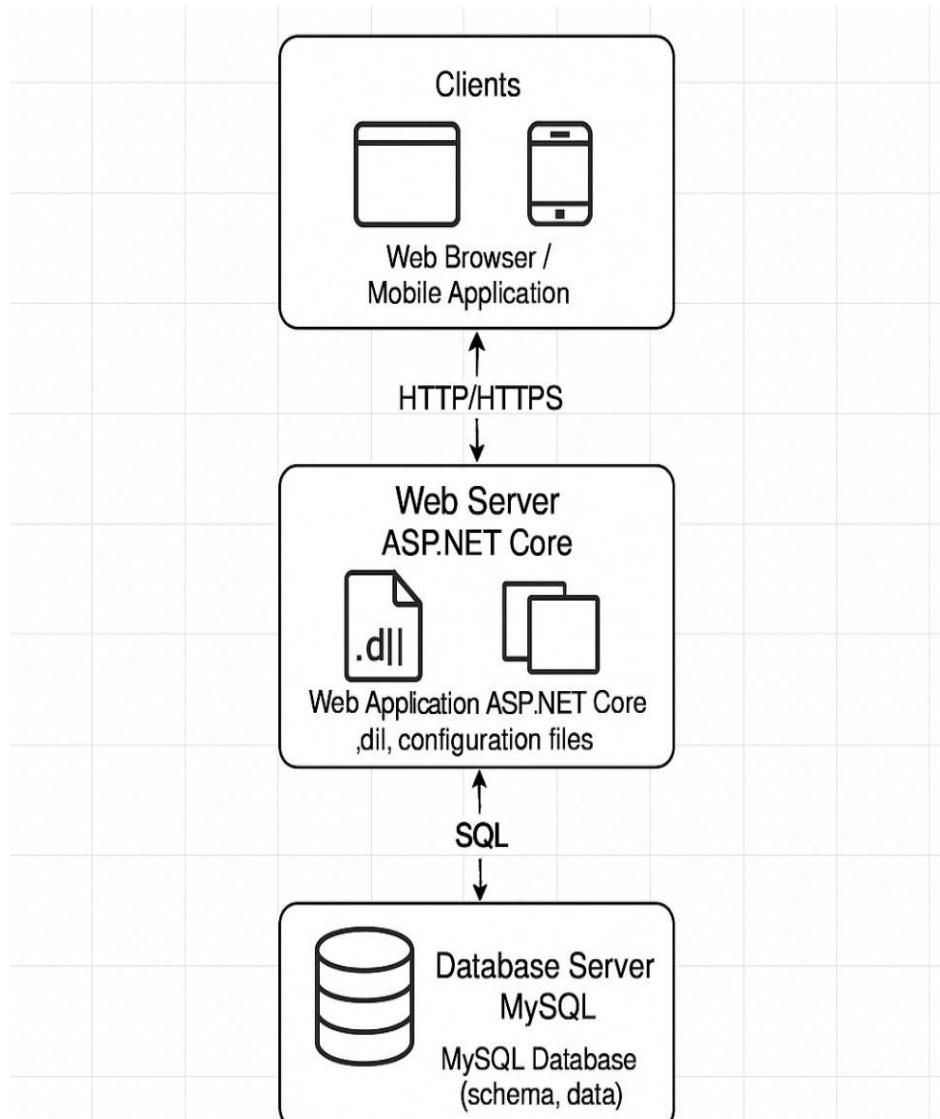
Este modelo permite un control robusto de la información y facilita futuras ampliaciones. La elección de tipos de datos ha sido realizada considerando la eficiencia y la integridad de los datos, asegurando compatibilidad con MySQL y buenas prácticas de diseño.

En resumen, el modelo de datos responde adecuadamente a las necesidades del sistema, garantizando flexibilidad, integridad referencial y la posibilidad de escalar conforme crezcan los requerimientos de la empresa.

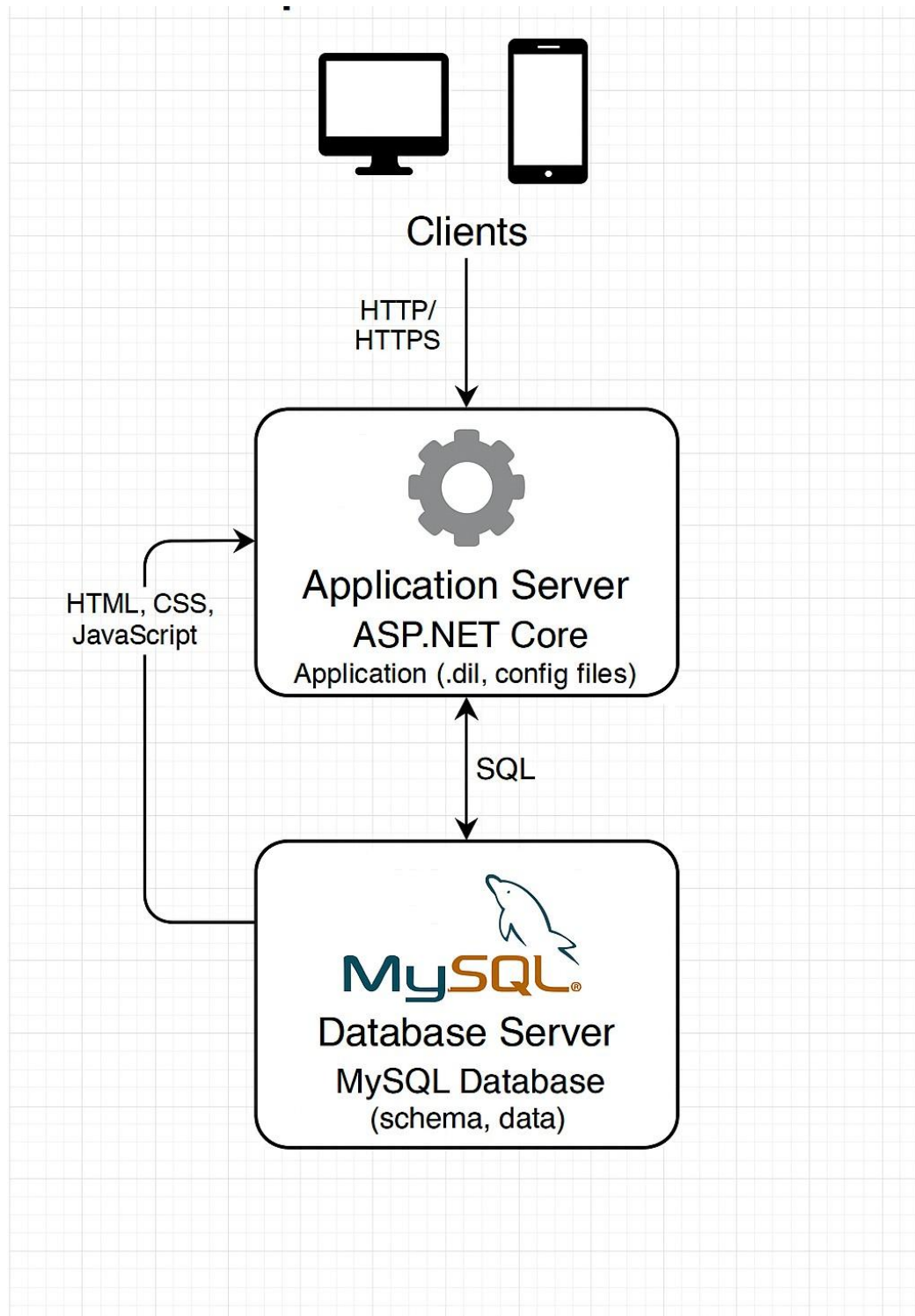
Diagrama del modelo físico de datos



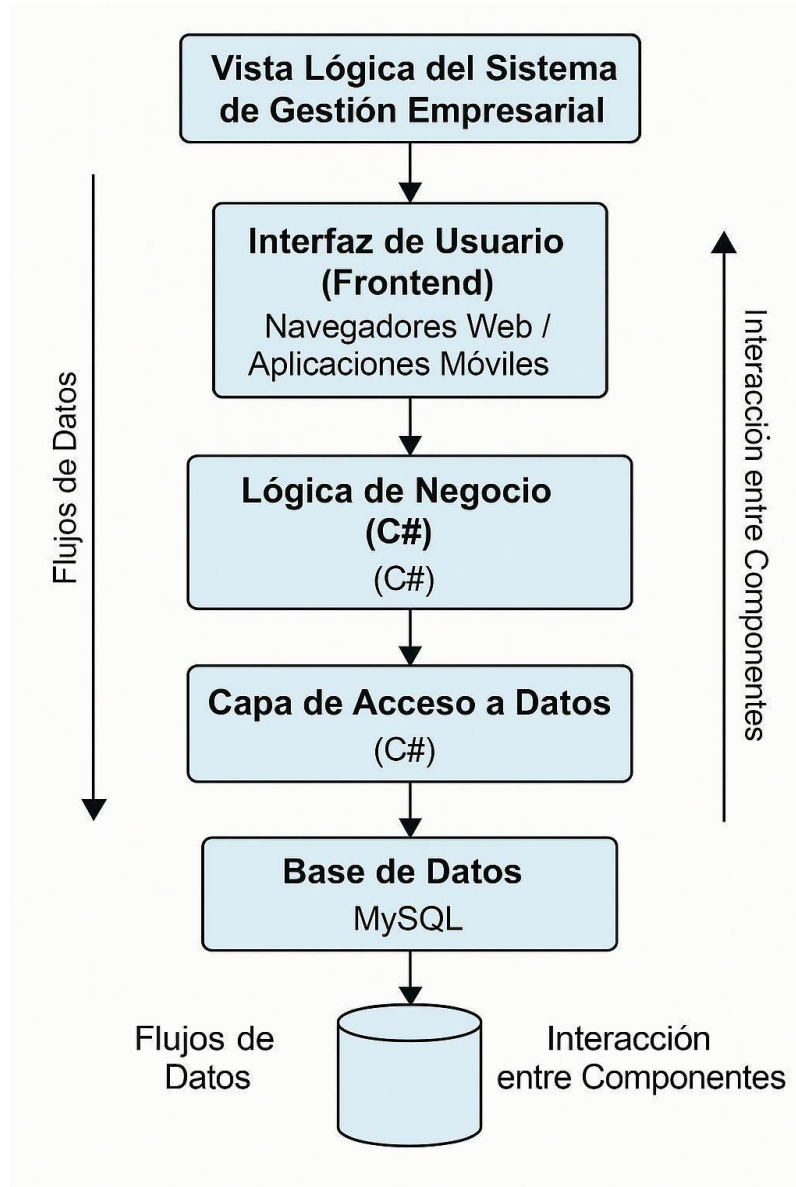
Vista de despliegue



Vista de implementación



Vista lógica



Vista de Procesos, explicación de los procesos

El proyecto presentado se enfoca en el desarrollo de un sistema de gestión empresarial orientado a pequeñas y medianas empresas, con el objetivo de optimizar sus procesos internos mediante una solución tecnológica moderna, eficiente y escalable. Se plantea el uso de ASP.NET Core con C# para el backend y MySQL como sistema de base de datos, lo cual responde a criterios técnicos de compatibilidad, rendimiento y bajo costo.

Esta solución permitirá manejar de manera organizada la información de empleados, clientes, productos/servicios, transacciones y usuarios del sistema, ofreciendo operaciones completas de creación, lectura, actualización y eliminación (CRUD) sobre estos elementos.

Plataforma Tecnológica

- ASP.NET Core (Backend): framework multiplataforma, de código abierto y altamente escalable.
- C# (Lenguaje de programación): seguro, moderno y con sintaxis limpia que garantiza mantenibilidad.
- MySQL (Base de datos): sistema robusto y gratuito, con soporte para grandes volúmenes de datos y relaciones complejas.

Esta arquitectura permite que el sistema pueda evolucionar fácilmente, integrarse con APIs en el futuro y desplegarse tanto en servidores locales como en la nube (Azure, AWS, etc.).

Modelo de Datos y Diseño Físico

El modelo de base de datos ha sido diseñado con un enfoque relacional, aplicando principios de integridad referencial y normalización. Las entidades principales incluyen:

Tablas del sistema:

- Empleados: Información del personal (nombre, cargo, fecha de ingreso).
- Clientes: Datos básicos del cliente (correo, dirección, teléfono).
- Productos/Servicios: Inventario de artículos y servicios con precios y stock.
- Transacciones: Registro de operaciones financieras como compras, pagos, depósitos y retiros, asociadas a clientes, empleados y productos.
- Usuarios: Acceso al sistema por roles ('Admin', 'Empleado') vinculado a un empleado.

Relaciones Clave:

- Un empleado puede atender múltiples transacciones.
- Un cliente puede generar múltiples transacciones.
- Una transacción involucra un producto específico.
- Cada usuario del sistema corresponde a un empleado de la empresa.

Beneficios del Sistema

- Mejora la organización y acceso a los datos internos.
- Aumenta la eficiencia en la gestión administrativa.
- Permite generar reportes y medir la productividad de empleados y movimientos financieros.
- Ofrece una interfaz amigable, pensada para usuarios sin conocimientos técnicos avanzados.
- Escalable para incluir más clientes, empleados o servicios.
- Estimación de Costos

Estimación del costo de desarrollo y mantenimiento de la aplicación, considerando licencias de software, hardware, tiempo de desarrollo, etc.

Para el desarrollo y mantenimiento del sistema de gestión empresarial en **ASP.NET Core con MySQL**, se deben considerar distintos factores que impactan directamente en el costo total del proyecto. A continuación, se detallan las estimaciones más relevantes:

1. Licencias de Software

- **ASP.NET Core:** Gratuito. Es de código abierto y no requiere licencias.
- **Visual Studio Community Edition:** Gratuito para equipos pequeños o proyectos educativos.
- **MySQL Community Edition:** Gratuito.
- **Sistema Operativo (Windows Server):** Aprox. **\$500 USD** por licencia estándar.
- **Licencias adicionales (por ejemplo, antivirus, herramientas de gestión, backup, etc.):** Estimado **\$200 USD**.

2. Costo del equipo de desarrollo (recursos humanos)

Suponiendo un equipo pequeño que trabaje durante 3-4 meses:

Rol	Cantida d	Costo mensual estimado	Tiempo (meses)	Total
Desarrollador Backend (C# / ASP.NET)	1	\$1,500 USD	4	\$6,000
Desarrollador Frontend (HTML, CSS, JS)	1	\$1,300 USD	4	\$5,200
Diseñador UI/UX	1	\$1,000 USD	2	\$2,000

Administrador de base de datos (MySQL)	1	\$1,200 USD	2	\$2,400
Tester / QA	1	\$1,000 USD	2	\$2,000
Total estimado	-	-	-	\$17,600 USD

3. Infraestructura / Hardware

- **Servidor físico o VPS:** Puede alquilarse un VPS por unos **\$20–\$50 USD/mes** (dependiendo de la capacidad).
- **Dominio y hosting:** **\$100 USD/año** (estimado).
- **Computadoras para desarrollo:** Ya asumidas por el equipo, pero una buena laptop puede costar **\$800–\$1,200 USD**.
- **Total anual estimado en infraestructura:** Aproximadamente **\$1,000 USD**.

4. Mantenimiento anual

- **Soporte técnico y corrección de errores:** \$2,000–\$3,000 USD/año.
- **Actualizaciones y mejoras menores:** \$1,000–\$2,000 USD/año.
- **Total de mantenimiento anual estimado:** **\$4,000–\$5,000 USD**.

Los costos del capital humano son estimaciones y promedios según algunos sitios web de empleos y ofertas laborales, mayormente se tomó de referencia a Glassdoor.

Cronograma detallado con las actividades clave del proyecto, tiempos estimados y recursos asignados para cada fase del desarrollo.

Fase / Actividad	Duración Estimada	Recursos Asignados	Descripción
1. Planificación y análisis de requerimientos	1 semana	Analista de sistemas, Líder de proyecto	Definir objetivos, alcance, funcionalidades y limitaciones del sistema.
2. Diseño del modelo de datos y arquitectura	1 semana	Arquitecto de software, DBA	Diseñar las entidades, relaciones, y el modelo ER de la base de datos.
3. Configuración del entorno de desarrollo	2 días	DevOps / Desarrollador Senior	Instalación de herramientas: Visual Studio, MySQL, frameworks, etc.
4. Desarrollo del backend (APIs, lógica de negocio)	2 semanas	Desarrolladores (2)	Crear los endpoints para CRUD, autenticación, lógica empresarial
5. Desarrollo del frontend (interfaces web)	1.5 semanas	Desarrolladores Frontend (2)	Diseñar e implementar las vistas para usuarios, formularios y navegación.
6. Integración backend-frontend	4 días	Desarrollador Full Stack	Conectar APIs con frontend, validaciones y pruebas básicas.
7. Pruebas funcionales y correcciones	1 semana	QA Tester	Verificar que el sistema funcione correctamente y corregir errores.
8. Documentación técnica y manual de usuario	4 días	Documentador técnico	Redactar guía de instalación, mantenimiento y uso del sistema.

9. Despliegue en entorno de producción	2 días	DevOps / Líder de proyecto	Implementación final en servidor o nube y configuración del entorno.
--	--------	----------------------------	--

Conclusiones

Ambos documentos muestran la solidez de la propuesta tanto desde la perspectiva técnica (modelo de datos bien estructurado, arquitectura moderna, lenguaje robusto), como desde una perspectiva administrativa (viabilidad, cronograma, costos reales).

Esta documentación refleja una simulación muy cercana a un entorno profesional de desarrollo y demuestra una comprensión profunda de cómo diseñar, planificar y justificar un sistema real de gestión empresarial.