CENTRO DE ESTUDIOS SUPERIORES DEL NOROESTE

Carrera:
Ingeniería en Desarrollo de Software.

Alumno :
Mario Quiñones Castro

Profesora:
Erika Denisse Arreola Xicali

Materia:
CALIDAD Y EVALUACIÓN EN PROYECTOS DE SOFTWARE

Actividad:
Proyecto Final

Tijuana, Baja California.

Sistema de Gestión de Clientes para Tortillería

ÍNDICE

1. INTRODUCCIÓN

El presente documento describe el desarrollo del Sistema de Gestión de Clientes para una tortillería local, el cual fue implementado como una aplicación de escritorio utilizando Windows Forms en C# bajo una arquitectura en capas (Presentación, Lógica de Negocio y Acceso a Datos).

El propósito del documento es detallar los objetivos, requerimientos funcionales, estimaciones de tiempo, recursos, costos, métricas de calidad, modelo aplicado, metodología de calidad, plan de pruebas y resultados obtenidos. Este proyecto fue desarrollado con fines académicos, aplicando principios de calidad de software y pruebas estructuradas.

2. OBJETIVOS DEL PROYECTO

2.1 Objetivo General

Desarrollar un sistema de escritorio utilizando Windows Forms en C# que permita gestionar la información de los clientes de una tortillería, facilitando el registro, consulta, edición y eliminación de datos para mejorar el control administrativo del negocio.

2.2 Objetivos Específicos

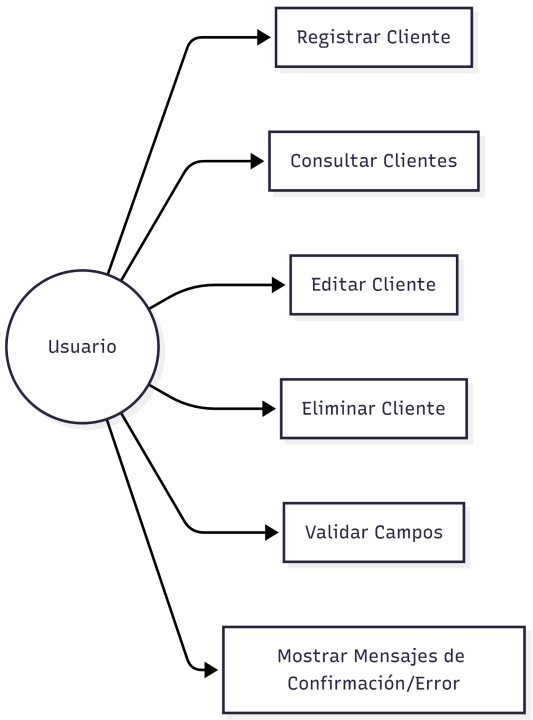
Implementar un módulo CRUD de clientes.
Aplicar arquitectura en capas (Presentación, Lógica de Negocio y Acceso a Datos).
Utilizar Entity Framework para la conexión con la base de datos.
Implementar validaciones que aseguren la integridad de la información.
Garantizar almacenamiento persistente y estructurado de los datos.

3. REQUERIMIENTOS FUNCIONALES

El sistema deberá cumplir con los siguientes requerimientos:
Permitir registrar nuevos clientes.
Mostrar el listado completo de clientes almacenados.

Permitir editar la información de un cliente existente.
Permitir eliminar clientes del sistema.
Validar que los campos obligatorios estén completos antes de guardar.
Mostrar mensajes de confirmación y error al usuario.
Establecer conexión correcta con la base de datos.
Mantener persistencia de la información registrada.

3.1 Diagrama de Casos de Uso



Descripción:

El diagrama de casos de uso representa la interacción entre el usuario y el Sistema de Gestión de Clientes para la tortillería. Se muestran las principales funcionalidades disponibles, como registrar, consultar, editar y eliminar clientes, así como la validación de datos y la visualización de mensajes de confirmación o error. Este diagrama permite identificar claramente el alcance funcional del sistema.

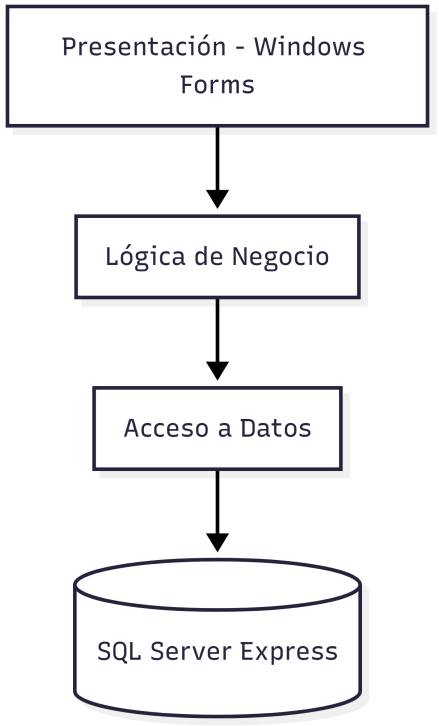
3.2 Modelo Entidad–Relación

CLIENTE		
int	IdCliente	PK
string	Nombre	
string	Telefono	
string	Direccion	
string	Email	
datetime	FechaRegistro	

Descripción:

El modelo entidad–relación representa la estructura de la base de datos utilizada por el sistema. Se muestra la entidad Cliente con sus respectivos atributos, incluyendo la clave primaria (IdCliente). Este modelo permite visualizar la organización de la información y garantiza la correcta persistencia de los datos en SQL Server Express.

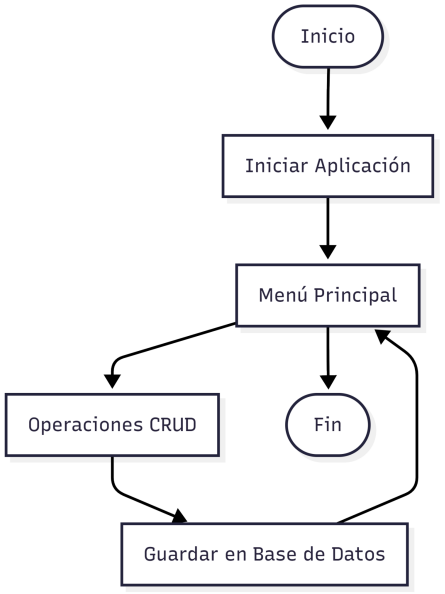
3.3 Arquitectura del Sistema



Descripción:

El sistema fue desarrollado bajo una arquitectura en capas compuesta por Presentación (Windows Forms), Lógica de Negocio y Acceso a Datos, conectadas a SQL Server Express. Esta estructura permite una mejor organización del código, facilita el mantenimiento y mejora la escalabilidad del sistema.

3.4 Diagrama de Flujo del Sistema



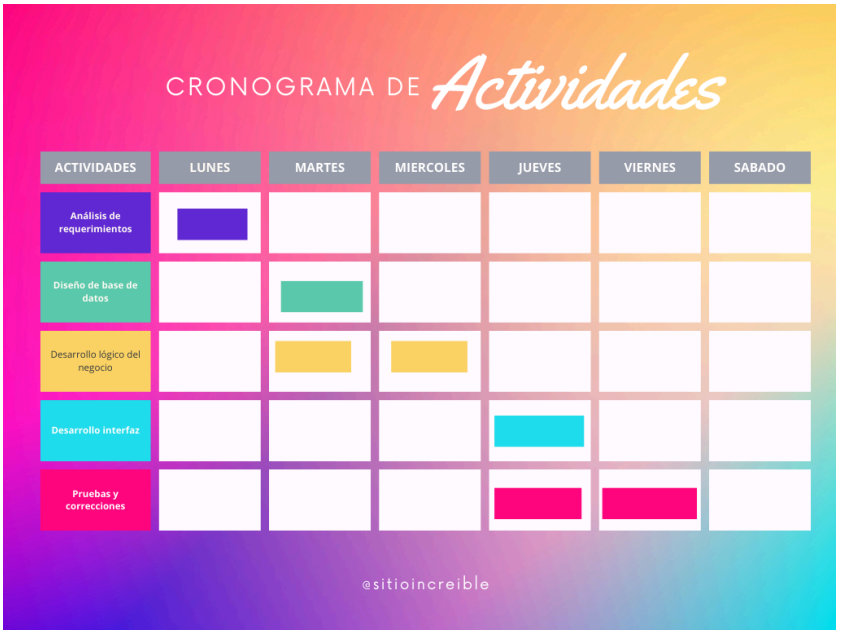
Descripción sugerida para pegar debajo:

El diagrama de flujo representa el proceso general de ejecución del sistema. Inicia con la apertura de la aplicación, continúa con el acceso al menú principal y la ejecución de operaciones CRUD (crear, leer, actualizar y eliminar). Posteriormente, los datos son almacenados en la base de datos y el sistema permite regresar al menú principal o finalizar la ejecución. Este diagrama permite visualizar la lógica general de funcionamiento del sistema.

4. ESTIMACIÓN DE TIEMPO

Se utilizó la técnica PERT considerando escenarios optimista, probable y pesimista.
Tiempo estimado total del proyecto: 40 horas
Distribución aproximada:
Análisis de requerimientos: 6 horas
Diseño de base de datos: 5 horas
Desarrollo de lógica de negocio: 8 horas
Desarrollo de interfaz Windows Forms: 10 horas
Pruebas y correcciones: 11 horas
Tiempo real invertido: 42 horas.

4.1 Cronograma del Proyecto



Descripción:

El cronograma muestra la distribución de las actividades del proyecto a lo largo del tiempo estimado, permitiendo visualizar la organización y planificación del desarrollo.

Reflexión:
La estimación fue cercana al tiempo real invertido. La ligera variación se debió a ajustes en validaciones y pruebas de conexión con la base de datos. La planificación permitió una organización adecuada del desarrollo.

5. ESTIMACIÓN DE RECURSOS

Recursos Humanos

1 desarrollador responsable del análisis, diseño, desarrollo y pruebas.
Recursos de Hardware
Laptop con 8GB de RAM
Sistema operativo Windows 10/11
Conexión a internet
Recursos de Software
Visual Studio
Lenguaje C#
Windows Forms
SQL Server Express
Entity Framework
Conocimientos Requeridos
Programación orientada a objetos
Manejo de bases de datos relacionales
Arquitectura en capas
Validación de datos y control de errores

6. ESTIMACIÓN DE COSTO

Tarifa estimada: $350 MXN por hora
Tiempo estimado: 40 horas

Costo directo:
40 × $350 = $14,000 MXN

Costos indirectos:
Energía e internet: $800 MXN
Uso de equipo: $1,200 MXN
Herramientas y licencias: $1,000 MXN

Total indirectos: $3,000 MXN

Costo total estimado:
$14,000 + $3,000 = $17,000 MXN

7. MÉTRICAS DE CALIDAD

Para evaluar la calidad del sistema se establecieron las siguientes métricas:
Cobertura de pruebas: 100% de métodos críticos evaluados.
Tasa de éxito de casos de prueba: **≥ 95%.**
Densidad de defectos: Número de errores detectados por módulo.
Tiempo de respuesta: Menor a 2 segundos en operaciones CRUD.
Estas métricas permiten evaluar funcionalidad, estabilidad y confiabilidad del sistema.

8. MODELO DE CALIDAD

Se aplicó el modelo ISO/IEC 25010, el cual evalúa:
Funcionalidad
Fiabilidad
Usabilidad
Eficiencia
Mantenibilidad
Este modelo fue seleccionado por su enfoque integral en calidad de software.

9. METODOLOGÍA DE CALIDAD

Se aplicó un enfoque de mejora continua basado en:
Pruebas unitarias durante el desarrollo.
Validaciones funcionales por módulo.
Corrección inmediata de defectos.
Documentación de errores y evidencias.
Este proceso permitió garantizar estabilidad antes de finalizar el sistema.

10. PLAN DE PRUEBAS

Se realizaron pruebas unitarias, funcionales e integración:
Pruebas Unitarias:
Agregar cliente
Editar cliente
Eliminar cliente
Validaciones

Pruebas Funcionales:

Interacción completa desde Windows Forms.

Pruebas de Integración:
Comunicación entre Presentación, Lógica de Negocio, Acceso a Datos y SQL Server Express.

Criterios de aceptación:
100% pruebas ejecutadas.
≥ 95% exitosas.
Sin errores críticos.

11. REPORTE DE PRUEBAS

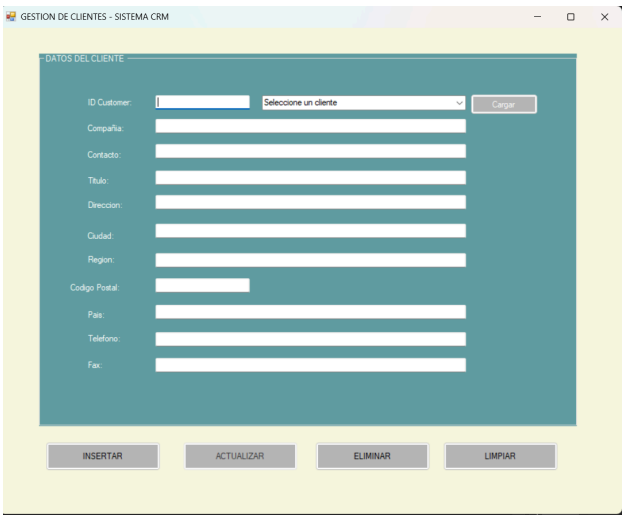
Total de casos ejecutados: 10
Casos exitosos: 10
Casos fallidos: 0
Errores críticos: 0
Errores menores corregidos: 2

Conclusión del reporte:
El sistema cumple con los requerimientos funcionales establecidos y demuestra estabilidad en la persistencia y manipulación de datos.

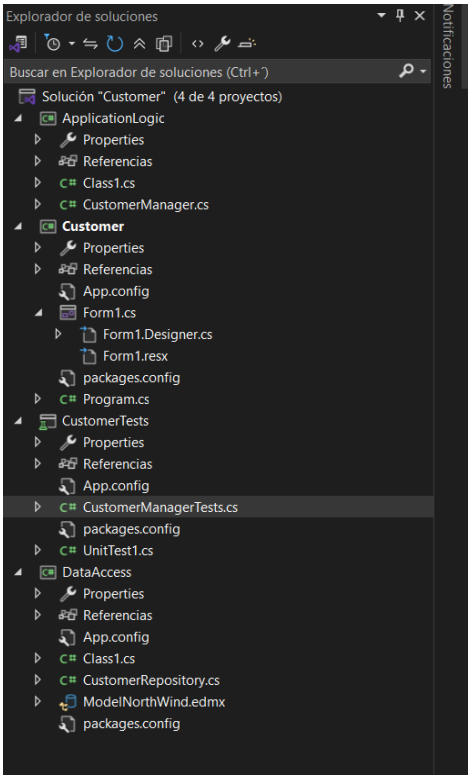
12. EVIDENCIAS

Se adjuntan capturas de pantalla donde se muestra:

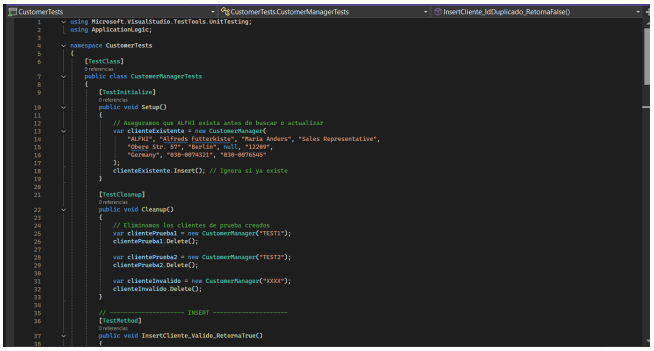
Interfaz de usuario funcionando.



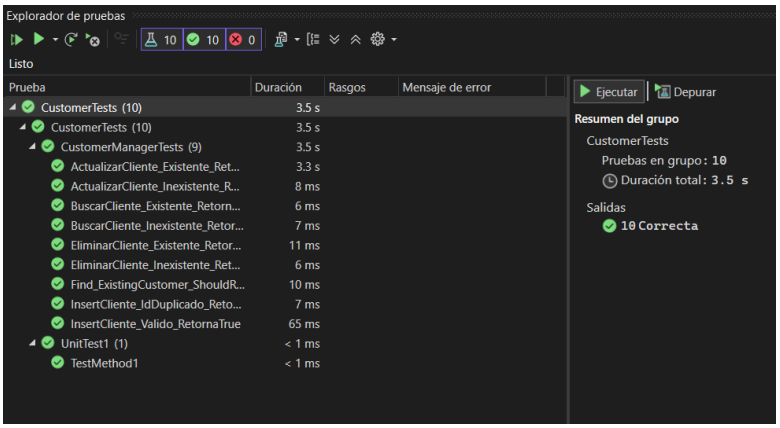
Estructura de carpetas del proyecto.



Código de pruebas unitarias.



Resultados de ejecución de pruebas.



13. CONCLUSIÓN

El desarrollo del Sistema de Gestión de Clientes permitió aplicar conocimientos de arquitectura en capas, bases de datos relacionales, pruebas unitarias y calidad de software. La implementación de métricas de calidad y un plan estructurado de pruebas garantizó el cumplimiento de los requerimientos funcionales.

A pesar de limitaciones de tiempo y recursos, el proyecto logró estabilidad, funcionalidad y correcta persistencia de datos. Esta actividad permitió reforzar la importancia de la planificación, validación continua y documentación en el desarrollo de software, consolidando competencias técnicas fundamentales para el ámbito profesional.