



PRESENTACION

NOMBRE: ANGEL LUIS ROSARIO

MATRICULA: 2024-0079

PROFESOR: Kelyn Tejada Belliard

SECCION: Programación III 2025-C-3

FECHA: 08/12/2025

SANTO DOMINGO ESTE/LA CALETA

ÍNDICE

- 1. ESTRATEGIA DE TRABAJO (PLANIFICACIÓN)**
 - 1.1 Nombre del proyecto de software**
 - 1.2 Tecnología para aplicar**
 - 1.3 Objetivo del proyecto**
 - 1.4 Alcance del proyecto**
 - 1.5 Cronograma del proyecto**
 - 1.6 Definición del primer Release**
- 2. METODOLOGÍA SCRUM**
 - 2.1 Equipo de Trabajo**
 - 2.2 Herramientas utilizadas**
 - 2.3 Definición de Épicas**
 - 2.4 Ceremonias de Scrum**
 - 2.5 Historias de Usuario**
- 3. PLAN DE PRUEBAS**
 - 3.1 Lista de requerimientos funcionales y no funcionales**
 - 3.2 Criterios de aceptación y rechazo de pruebas**
 - 3.3 Herramientas de pruebas**
 - 3.4 Casos de Prueba**
 - 3.5 Evidencia de Automatización**
- 4. DEMOSTRACIÓN Y ENTREGABLES**
 - 4.1 Video demostrativo**
 - 4.2 Enlaces al Repositorio y Gestión del Proyecto**

1. Estrategia de Trabajo (Planificación)

1.1 Nombre del proyecto de software

Sistema Gestor de Tareas Personales (Task Manager)

1.2 Tecnología para aplicar

Para el desarrollo de este sistema se han seleccionado tecnologías modernas y robustas del ecosistema Microsoft .NET, garantizando escalabilidad y rendimiento:

- **Lenguaje de Programación:** C# (Versión 12/13).
- **Framework:** .NET 9 con ASP.NET Core MVC (Model-View-Controller).
- **Base de Datos:** SQLite (Base de datos relacional ligera y portable).
- **ORM:** Entity Framework Core 9 (Code-First con Migraciones Automáticas).
- **Frontend:** Razor Views (.cshtml), HTML5, CSS3 (Bootstrap).
- **IDE:** Visual Studio / VS Code.

1.3 Objetivo del proyecto

Desarrollar una aplicación web eficiente que permita a los usuarios organizar sus actividades diarias mediante la creación, gestión y priorización de tareas, facilitando el seguimiento de pendientes y mejorando la productividad personal.

1.4 Alcance del proyecto

El sistema abarca la gestión completa del ciclo de vida de una tarea.

- **Incluye:** Registro de tareas, edición de contenido, eliminación, marcado de estado (completado/pendiente), asignación de prioridades (Alta, Media, Baja) y filtrado avanzado.
- **No incluye:** Gestión de usuarios múltiples login, notificaciones por correo electrónico ni integración con calendarios externos en esta versión.

1.5 Cronograma del proyecto

Actividad	Responsable	Fecha de Inicio	Fecha de Fin	Estado
Configuración del entorno y Repositorio	Angel Luis	03/12/2025	03/12/2025	Completado
Definición de Historias en Azure DevOps	Angel Luis	05/12/2025	05/12/2025	Completado
Desarrollo del Backend (Modelos y BD)	Angel Luis	06/12/2025	07/12/2025	Completado
Desarrollo del Frontend (Vistas y Controladores)	Angel Luis	06/12/2025	07/12/2025	Completado
Pruebas y Depuración	Angel Luis	07/12/2025	08/12/2025	Completado

Actividad	Responsable	Fecha de Inicio	Fecha de Fin	Estado
Documentación Final	Angel Luis	07/12/2025	08/12/2025	Completado

1.6 Definición del primer Reléase

La versión 1.0 del sistema es un producto mínimo viable (MVP) completamente funcional que permite:

- **Gestión CRUD:** Crear, Leer, Actualizar y Eliminar tareas.
- **Organización:** Clasificación por prioridades (Alta, Media, Baja).
- **Búsqueda:** Filtrado de tareas por título.
- **Visualización:** Listado ordenado automáticamente por urgencia y estado.

Requerimientos Funcionales:

1. El sistema debe permitir crear tareas con título, descripción y fecha de vencimiento.
2. El sistema debe permitir filtrar tareas por "Completadas" y "Pendientes".
3. El sistema debe validar que el título no esté vacío.

Requerimientos No Funcionales:

1. La base de datos debe crearse automáticamente al iniciar la aplicación (Migración automática).
2. La interfaz debe ser responsive y clara para el usuario.
3. El sistema debe utilizar SQLite para portabilidad sin instalación de servidores.

2. Metodología Scrum

2.1 Equipo de Trabajo

- **Product Owner:** Angel Luis (Definición de requisitos y prioridades).
- **Scrum Máster:** Angel Luis (Gestión del tablero en Azure DevOps y eliminación de bloqueos).
- **Development Team:** Angel Luis (Codificación, diseño de base de datos y pruebas).

2.2 Herramientas utilizadas

- **Gestión del Proyecto:** Azure DevOps (Boards).
- **Control de Versiones:** Git / GitHub.

2.3 Definición de Épicas

1. **Gestión de Tareas (Core):** Funcionalidades básicas de creación y almacenamiento.
2. **Productividad y Filtros:** Herramientas para organizar y encontrar tareas.

2.4 Ceremonias de Scrum

- **Sprint Planning:** 05/12/2025 - Se seleccionaron las 10 historias para el Sprint 1.

- **Daily Stand-ups:** Realizados diariamente de forma asíncrona para verificar progreso.
- **Sprint Review:** 08/12/2025 - Verificación del software funcionando antes de la entrega.

2.5 Historias de Usuario Detalladas

ID	Título	Descripción	Criterios de Aceptación	Puntos
HU-01	Crear Básica Tarea	Como usuario, quiero registrar una tarea nueva para recordarla.	1. El sistema debe validar que el título no esté vacío. 2. Al guardar, debe aparecer en la lista principal. 3. La fecha por defecto debe ser hoy o futura.	3
HU-02	Editar Tarea	Como usuario, quiero modificar una tarea si cometo un error.	1. El formulario debe cargar los datos actuales. 2. Al guardar, los cambios (título, prioridad) deben persistir en la base de datos. 3. No debe duplicar la tarea al editar.	3
HU-03	Eliminar Tarea	Como usuario, quiero borrar tareas que ya no necesito.	1. Debe existir un botón de eliminar visible. 2. Al confirmar, la tarea debe desaparecer de la lista y de la base de datos inmediatamente.	2
HU-04	Marcar como Completada	Como usuario, quiero marcar tareas listas para ver mi progreso.	1. Debe permitir cambiar el estado de "Pendiente" a "Completada" con un clic. 2. Visualmente debe distinguirse (ej. tachado o check verde).	1
HU-05	Ver listado de Tareas	Como usuario, quiero ver todas mis tareas en una sola pantalla.	1. La lista debe cargar rápido al inicio. 2. Debe mostrar título, fecha y prioridad de cada elemento.	2

ID	Título	Descripción	Criterios de Aceptación	Puntos
			3. Si no hay tareas, mostrar mensaje "No hay tareas".	
HU-06	Asignar Prioridad	Como usuario, quiero clasificar tareas (Alta/Media/Baja).	1. El formulario de creación debe tener un selector de prioridad. 2. En la lista, la prioridad debe ser visible (ej. etiquetas de colores).	2
HU-07	Definir Fecha de Vencimiento	Como usuario, quiero poner fechas límite a mis tareas.	1. El sistema debe permitir seleccionar una fecha y hora. 2. La fecha debe guardarse y mostrarse en el listado principal.	2
HU-08	Filtrar Estado por	Como usuario, quiero ocultar las tareas completadas.	1. Debe existir un filtro para ver "Todas", "Pendientes" o "Completadas". 2. La lista debe actualizarse sin recargar toda la página si es posible.	3
HU-09	Filtrar Prioridad por	Como usuario, quiero ver solo lo urgente (Prioridad Alta).	1. Al seleccionar "Alta", solo deben aparecer esas tareas. 2. El filtro debe poder limpiarse para ver todo de nuevo.	3
HU-10	Visualización de Vencidas	Como usuario, quiero identificar rápidamente qué tareas ya vencieron.	1. Las tareas con fecha anterior a "hoy" deben aparecer primero o resaltadas. 2. El sistema debe ordenarlas automáticamente por fecha.	3

3. Plan de Pruebas

3.1 Estrategia de Pruebas

Se han realizado pruebas manuales exhaustivas y se ha diseñado un plan de pruebas unitarias para garantizar la estabilidad del sistema.

3.2 Criterios de Aceptación

- Una prueba se considera **Exitosa** si el resultado obtenido coincide con el resultado esperado y no genera errores en tiempo de ejecución.

- Una prueba se considera **Fallida** si la aplicación se detiene o los datos no se guardan en la base de datos SQLite.

3.3 Herramientas de Pruebas

- **Manuales:** Navegador Web
- **Automatizadas:** xUnit Framework para pruebas unitarias en .NET.

3.4 Casos de Prueba

ID Caso	Descripción	Resultado Esperado	Estado
CP-01	Crear tarea con título vacío (Manual)	El sistema muestra alerta roja "El título es obligatorio".	Aprobado
CP-02	Guardar tarea válida (Manual)	La tarea aparece en la lista principal con su ID generado.	Aprobado
CP-03	Filtrar por Prioridad Alta (Manual)	La lista se actualiza y solo muestra tareas con etiqueta "Alta".	Aprobado
CP-04	Persistencia de datos (Manual)	Al reiniciar la aplicación, las tareas guardadas siguen existiendo.	Aprobado
CP-05	HU-01: Crear Tarea (Automatizada xUnit)	El método Crear() debe incrementar el conteo en BD a 1.	Aprobado
CP-06	HU-02: Editar Tarea (Automatizada xUnit)	El método Editar() debe modificar el título en la BD.	Aprobado
CP-07	HU-03: Eliminar Tarea (Automatizada xUnit)	El método Eliminar() debe dejar el conteo de la BD en 0.	Aprobado
CP-08	HU-04: Completar Tarea (Automatizada xUnit)	El método Completar() debe cambiar EstaCompletada a True.	Aprobado

3.5 Evidencia de Ejecución de Pruebas Automatizadas

A continuación, se presenta la captura de pantalla de la terminal de Visual Studio Code tras ejecutar el comando dotnet test. Se evidencia que las pruebas unitarias diseñadas con xUnit pasaron exitosamente (Passed), validando la lógica del negocio sin errores.

The screenshot shows the Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a GitHub Copilot icon. The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom shows 'Ready' and other system information.

In the center-left, there's a code editor window titled 'PruebasTareas.cs' with the following content:

```
1 using Xunit;
2 using Microsoft.EntityFrameworkCore;
3 using GestorTareas.Controllers;
4 using GestorTareas.Data;
5 using GestorTareas.Models;
6 using System.Linq;
7 using System;
```

The status bar below the code editor says 'No issues found'.

To the right of the code editor is the 'Solution Explorer' window, which lists 'Solution tarea3' (2 of 2 projects) containing 'GestorTareas' and 'GestorTareas.Tests'.

At the bottom left, there's a 'Developer PowerShell' window showing the output of an xUnit test run:

```
[xUnit.net 00:00:00.13] Starting: GestorTareas.Tests
[xUnit.net 00:00:01.02] Finished: GestorTareas.Tests
GestorTareas.Tests test succeeded (1.9s)

Test summary: total: 4, failed: 0, succeeded: 4, skipped: 0, duration: 1.9s
Build succeeded in 4.5s

Workload updates are available. Run 'dotnet workload list' for more information.
PS C:\Users\angel\OneDrive - Instituto Tecnológico de Las Américas (ITLA)\Escritorio\tarea3>
```

4. Enlaces y Entregables

Repositorio de Código (GitHub): [Issues · ALR29/p3](#)

Gestión del Proyecto (Azure DevOps): https://dev.azure.com/Practica9-20240079/ProTasker/_sprints/taskboard/ProTasker%20Team/ProTasker/Sprint%201

Video Demostrativo: <https://youtu.be/GIOCmUdUmGc>