**Universidad Autónoma de Zacatecas**

**Unidad Académica de Ingeniería Eléctrica**

**Ingeniería de Software**

**Análisis y Diseño Orientado a Objetos**

**PROYECTO FINAL – INCEPTION**

**Alumno:**

**Marcos de Jesús Arredondo Rebollo**

**Profesor: Cristian Boyain**

**Sistema de Control de Metas Personales**

**1. Visión del Proyecto**

**Problema a Resolver:** Las personas tienen dificultades para organizar, dar seguimiento y completar sus metas personales, académicas y profesionales. Actualmente registran sus objetivos de manera informal (notas, memoria) lo que resulta en desorganización y baja tasa de cumplimiento.

**Solución Propuesta:** Desarrollar una aplicación de escritorio en WPF que permita a los usuarios gestionar sistemáticamente sus metas personales, con seguimiento de progreso, estadísticas y un panel centralizado de control.

**Objetivo Principal:** Crear una herramienta digital que mejore la productividad personal y aumente significativamente las probabilidades de alcanzar objetivos mediante un enfoque disciplinado y visual del progreso.

**2. Alcance del Proyecto**

**2.1 Funcionalidades Incluidas**

* **Gestión de Metas**: Crear, modificar, consultar y eliminar metas con información completa
* **Seguimiento de Progreso**: Marcar metas como completadas y controlar estados
* **Panel Principal**: Dashboard con resumen ejecutivo del estado general
* **Estadísticas**: Visualización de métricas de rendimiento y cumplimiento
* **Navegación Intuitiva**: Acceso fácil a todas las funciones del sistema

**2.2 Funcionalidades Excluidas (Esta Versión)**

* Sincronización en la nube
* Aplicación móvil o web
* Metas colaborativas o compartidas
* Notificaciones push o recordatorios
* Integración con calendarios externos

**3. Objetivos y Criterios de Éxito**

**3.1 Objetivos Específicos**

* Permitir gestión completa de metas personales offline
* Proporcionar seguimiento visual del progreso de objetivos
* Generar estadísticas automáticas de rendimiento
* Mantener todos los datos seguros localmente
* Ofrecer interfaz intuitiva siguiendo estándares Windows

**3.2 Criterios de Éxito**

* **Rendimiento**: Aplicación responde en menos de 1 segundo
* **Capacidad**: Soporta hasta 1000 metas sin degradación
* **Usabilidad**: Usuario puede crear meta en menos de 3 clics
* **Estabilidad**: Inicia correctamente 99% de las veces
* **Adopción**: Usuario se familiariza con funciones en 10 minutos

**4. Stakeholders y Usuarios**

**4.1 Usuarios Objetivo**

* **Perfil Principal**: Personas de 16+ años interesadas en mejora personal
* **Características**: Conocimientos básicos de computación, acceso a PC Windows
* **Motivaciones**: Organización personal, mejora de productividad, logro de objetivos

**4.2 Stakeholders del Proyecto**

* **Usuarios Finales**: Personas que usarán la aplicación
* **Equipo de Desarrollo**: Desarrolladores, diseñadores, testers
* **Project Manager**: Responsable de coordinar el proyecto

**5. Arquitectura y Tecnología**

**5.1 Plataforma y Tecnologías**

* **Plataforma**: Windows Desktop (.NET Framework 4.7.2+)
* **Tecnología Principal**: WPF (Windows Presentation Foundation)
* **Base de Datos**: SQLite (almacenamiento local)
* **Arquitectura**: MVVM (Model-View-ViewModel)

**5.2 Patrones de Diseño**

* **MVVM**: Separación clara entre lógica de negocio y presentación
* **Singleton**: GestorDeMetas para coordinar operaciones centralizadas
* **Observer**: Actualización automática de interfaces cuando cambian datos
* **Repository**: Abstracción del acceso a datos

**6. Análisis de Riesgos**

**6.1 Riesgos Altos**

| **Riesgo** | **Probabilidad** | **Impacto** | **Mitigación** |
| --- | --- | --- | --- |
| **Subestimación de tiempo** | Alta | Medio | Estimaciones detalladas + 25% buffer de tiempo |
| **Cambios en requisitos** | Media | Alto | Control formal de cambios + desarrollo iterativo |
| **Interfaz poco intuitiva** | Media | Alto | Pruebas con usuarios + seguir estándares Windows |
| **Problemas de pruebas** | Media | Alto | Pruebas automatizadas + criterios de aceptación |

**6.2 Riesgos Medios**

| **Riesgo** | **Probabilidad** | **Impacto** | **Mitigación** |
| --- | --- | --- | --- |
| **Corrupción de datos** | Baja | Muy Alto | Respaldos automáticos + transacciones atómicas |
| **Rendimiento de SQLite** | Media | Medio | Índices optimizados + pruebas de carga |
| **Complejidad MVVM** | Media | Medio | Capacitación previa + frameworks establecidos |
| **Falta experiencia WPF** | Media | Medio | Entrenamiento + prototipos de aprendizaje |

**6.3 Riesgos Bajos**

* **Incompatibilidad .NET**: Verificación de versión en instalador
* **Actualizaciones Windows**: Pruebas en diferentes versiones del SO
* **Adopción baja de usuarios**: Demostrar valor agregado claro

**7. Cronograma General**

**Periodo del Proyecto:** 12 de abril al 5 de junio de 2025  
**Duración:** 8 semanas (4 sprints de 2 semanas + fase de cierre)  
**Metodología:** Desarrollo ágil con sprints de 2 semanas

**7.1 Sprint 1: Preparación y Análisis**

**Fechas:** 12 de abril - 25 de abril (2 semanas)

**Actividades:**

* Elaboración del documento de inicio del proyecto
* Definición detallada de requisitos funcionales y no funcionales
* Análisis de dominio del problema
* Desarrollo de diagramas UML iniciales (casos de uso y clases)
* Diseño de la arquitectura siguiendo el patrón MVVM
* Configuración del entorno de desarrollo y repositorio de código

**Entregables:**

* Documento de inicio del proyecto completo
* Backlog del producto priorizado
* Diagrama de casos de uso
* Diagrama de clases inicial
* Documento de arquitectura del sistema
* Repositorio de código configurado

**7.2 Sprint 2: Diseño y Base de Implementación**

**Fechas:** 26 de abril - 9 de mayo (2 semanas)

**Actividades:**

* Diseño detallado de las clases del modelo (Meta, GestorDeMetas, Estadísticas)
* Diseño de la base de datos para almacenamiento de metas
* Creación de prototipos de interfaces de usuario
* Implementación inicial de las clases del modelo
* Desarrollo de la base de datos y componentes básicos de acceso a datos
* Implementación del patrón Singleton para GestorDeMetas

**Entregables:**

* Diseño detallado de clases y base de datos
* Prototipos de interfaz de usuario
* Modelo de dominio implementado a nivel básico
* Estructura de base de datos funcionando

**7.3 Sprint 3: Implementación Core**

**Fechas:** 10 de mayo - 23 de mayo (2 semanas)

**Actividades:**

* Implementación completa de las clases del modelo
* Desarrollo completo de componentes de acceso a datos
* Implementación de ViewModels básicos
* Creación de interfaces de usuario principales
* Implementación del CRUD básico de metas
* Testing unitario de componentes core

**Entregables:**

* Modelo de dominio completamente implementado
* Componentes de acceso a datos funcionando
* Interfaces de usuario básicas operativas
* Primera versión funcional del sistema (MVP)
* Conjunto inicial de pruebas unitarias

**7.4 Sprint 4: Funcionalidades Principales**

**Fechas:** 24 de mayo - 6 de junio (2 semanas)

**Actividades:**

* Implementación completa del CRUD de metas
* Desarrollo de la funcionalidad de estadísticas
* Implementación del patrón Observer para actualizaciones de UI
* Mejoras en la interfaz de usuario según feedback
* Implementación de características adicionales prioritarias
* Validaciones y manejo de errores
* Testing de integración