



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL
INTERDISCIPLINARIA DE INGENIERÍA Y
CIENCIAS SOCIALES Y ADMINISTRATIVAS

6NM60 Ingeniería de Pruebas

**Plan de Proyecto: Calculadora digital
intuitiva y multifuncional.**

Alumnos:

García Méndez Juan Carlos

Conde Basilio Leonardo

Ramos Velázquez Felipe

Villaseñor Trejo Enrique

Docente:

Cruz Martínez Ramón

Fecha: 20 de marzo del 2025



Contenido

Introducción	3
Alcance y Contexto.....	3
2.1 Objetivos Generales	4
2.2 Objetivos Específicos	4
3. Alcance Detallado.....	4
3.2 Exclusiones	5
4. Entregables y Cronograma	6
4.2 Cronograma Expandido (Diagrama de Gantt)	6
5. Recursos y Presupuesto	7
5.2 Recursos Tecnológicos	7
6. Gestión de Riesgos	8
6.2 Plan de Contingencia.....	8
7. Gestión de la Calidad.....	8
7.2 Proceso de Pruebas.....	9
8. Comunicación y Gobernanza.....	9
9. Gestión de Cambios	10
10. Cierre del Proyecto.....	10
11. Anexos	11

Introducción

Desarrollar una calculadora programada con funcionalidades básicas, gestión de memoria y una interfaz intuitiva, utilizando tecnologías Java. El proyecto busca resolver la necesidad de una herramienta educativa y profesional con alta precisión en cálculos y usabilidad accesible.

Alcance y Contexto

- **Stakeholders Principales:**
 - **Cliente:** Institución educativa que requiere una herramienta para enseñanza de matemáticas.
 - **Usuario Final:** Estudiantes y profesionales que necesitan realizar operaciones básicas y gráficos simples.
 - **Equipo Técnico:** Analista, diseñador UI/UX, programador Java, tester.

2.1 Objetivos Generales

- Entregar una calculadora funcional en 30 días.
- Garantizar una tasa de satisfacción del usuario del 90%.

2.2 Objetivos Específicos

Objetivo	Métrica de Éxito
Implementar operaciones básicas	100% de los casos de uso críticos (CUC01-CUC03) funcionando.
Integrar gestión de memoria	Botones M+, MR, MC operativos sin errores en pruebas de estrés.
Optimizar rendimiento	Respuesta ≤ 1 segundo en el 95% de las operaciones.
Asegurar compatibilidad	Funcionamiento en Windows 10, macOS 12 y navegadores modernos.

3. Alcance Detallado

3.1 Funcionalidades Incluidas

Módulo de Cálculo

- **Operaciones Básicas:**
 - Suma, resta, multiplicación y división con soporte para números negativos y decimales (2 dígitos).
 - Validación de división por cero (RF02).
- **Gestión de Memoria:**
 - Almacenamiento (M+), recuperación (MR), limpieza (MC) y acumulación (RF08-RF11).

Módulo de Interfaz

- Diseño tipo teclado telefónico (3x4) con botones numéricos y operadores (RF06).
- Historial de operaciones visible (últimas 20) en ventana lateral (RF04).
- Feedback visual mediante colores y tooltips (RNF02).

Módulo de Seguridad

- Validación de entradas para evitar inyección de código (RNF04).

3.2 Exclusiones

- **No Incluye:**
 - Persistencia de datos tras cerrar la aplicación (limitación técnica aceptada).
 - Funciones trigonométricas o estadísticas (priorizadas para una fase futura).
 - Soporte para dispositivos móviles (iOS/Android).

4. Entregables y Cronograma

4.1 Entregables Principales

Entregable	Descripción	Fecha	Responsable
Documento de Requisitos	RF y RNF detallados con casos de uso críticos.	Día 3	Analista
Prototipo en Figma	Mockups interactivos validados por usuarios.	Día 10	Diseñador UI/UX
Código Fuente	Módulos GUI, lógica y memoria en Java.	Día 20	Programador
Versión Beta	Build funcional con pruebas internas.	Día 25	Tester
Manual de Usuario	Guía de instalación y uso paso a paso.	Día 28	Analista
Versión Final	JAR ejecutable + documentación técnica.	Día 30	Equipo Completo

4.2 Cronograma Expandido (Diagrama de Gantt)

Fase	Actividades	Duración	Dependencias
Planificación	- Reunión de kick-off.	Días 1-3	Ninguna
	- Priorización de requisitos.		
Diseño	- Creación de wireframes en Figma.	Días 4-10	Requisitos aprobados.
	- Definición de arquitectura modular.		
Desarrollo	- Implementación GUI (Java Swing).	Días 11-25	Prototipos validados.

Pruebas	- Lógica de operaciones y memoria.		
	- Integración de gráficos con Matplotlib.		
	- Pruebas unitarias (JUnit).	Días 26-28	Código estable.
	- Pruebas de usabilidad con 10 usuarios.		
Despliegue	- Generación de JAR ejecutable.	Días 29-30	Pruebas exitosas.
	- Redacción de manuales técnicos y de usuario.		

5. Recursos y Presupuesto

5.1 Recursos Humanos

Rol	Responsabilidades	Horas Estimadas
Analista	- Levantamiento de requisitos.	40 horas
	- Validación de casos de uso.	
Diseñador UI/UX	- Creación de mockups y prototipos.	30 horas
Programador	- Codificación en Java.	120 horas
Tester	- Ejecución de pruebas funcionales.	50 horas

5.2 Recursos Tecnológicos

Recurso	Descripción	Costo (USD)
NetBeans 25	IDE para desarrollo en Java.	\$0 (gratis)
Figma	Diseño de interfaces.	\$12/mes (Plan Estudiantil)
GitHub Pro	Repositorio privado para código.	\$4/mes

6. Gestión de Riesgos

6.1 Matriz de Riesgos

Riesgo	Probabilidad	Impacto	Mitigación	Dueño
Retraso en integración GUI	Alta	Alto	Revisiones diarias de código.	Programador
Errores en cálculos con decimales	Media	Crítico	Pruebas unitarias con JUnit.	Tester
Incompatibilidad en macOS	Baja	Alto	Pruebas tempranas en VM con macOS.	Tester
Cambios en requisitos	Media	Medio	Reuniones semanales con stakeholders.	Analista

6.2 Plan de Contingencia

- **Escenario 1:** Fallo crítico en la GUI.
 - **Acción:** Revertir a una versión estable y asignar recursos adicionales.
- **Escenario 2:** Baja disponibilidad del programador.
 - **Acción:** Contratar un freelancer para tareas urgentes.

7. Gestión de la Calidad

7.1 Estándares

- **Código:** Convenciones de Java (camelCase, comentarios Javadoc).
- **Pruebas:** Cobertura >85% con JUnit.
- **UI/UX:** Cumplimiento de WCAG 2.1 para accesibilidad.

7.2 Proceso de Pruebas

Tipo de Prueba	Herramienta	Criterios de Aceptación
Unitarias	JUnit 5	100% de los RF validados.
Rendimiento	JMeter	Respuesta ≤ 1 segundo en operaciones.
Usabilidad	Encuestas	90% de usuarios navegan sin ayuda.
Compatibilidad	VirtualBox	Funcionamiento en SO y navegadores.

8. Comunicación y Gobernanza

8.1 Estructura de Comunicación

Reunión	Frecuencia	Participantes	Objetivo
Daily Stand-up	Diaria	Equipo técnico	Reportar avances y bloqueos.
Revisión de Hitos	Semanal	Stakeholders + Equipo	Validar entregables y ajustar alcance.
Retrospectiva	Al finalizar	Equipo técnico	Identificar lecciones aprendidas.

8.2 Canales de Comunicación

- **Slack:** Coordinación diaria y resolución rápida de dudas.
- **GitHub Issues:** Seguimiento de bugs y tareas.
- **Google Meet:** Reuniones formales con stakeholders

9. Gestión de Cambios

9.1 Proceso de Solicitud de Cambios

1. **Solicitud:** Stakeholder envía formulario detallando el cambio.
2. **Evaluación:** Equipo técnico analiza impacto en cronograma y costos.
3. **Aprobación:** Comité de cambios (analista + cliente) decide.
4. **Implementación:** Programador actualiza requisitos y código.

9.2 Ejemplo de Cambio

- **Solicitud:** Añadir función de potenciación (x^y).
- **Impacto:** +5 días de desarrollo, +\$800 en costos.

10. Cierre del Proyecto

10.1 Actividades de Cierre

- Entrega formal de la aplicación y documentación.
- Capacitación a usuarios finales (videollamada de 1 hora).
- Encuesta de satisfacción del cliente.

10.2 Lecciones Aprendidas

- Documentar errores frecuentes en integración GUI-lógica para futuros proyectos.
- Incluir pruebas de compatibilidad desde la fase de diseño.

11. Anexos

11.1 Diagramas Técnicos

- **Diagrama de Clases:** Relación entre Calculadora, Memoria y Pantalla.
- **Diagrama de Secuencia:** Flujo de una operación con memoria ($M+ \rightarrow MR$).

11.2 Glosario de Términos

- **BigDecimal:** Tipo de dato para cálculos precisos en Java.
- **ActionListener:** Interfaz en Swing para gestionar eventos de clic.