

Documento de la Arquitectura del Software

1. Introducción

Las matemáticas requieren cálculos específicos, es por eso por lo que el uso de maquinas es algo común, sobre todo cuando se necesita precisión. Las calculadoras se crearon para agilizar procesos cognitivos como la suma de grandes números, la resta, multiplicación, etc. Una maquina siempre será mas precisa que un humano por lo que la creación de una calculadora que asista en operaciones siempre será de mucha ayuda.

1.1 Propósito

La creación de esta calculadora es mas que nada por un proyecto de software. Pero su propósito general es que pueda realizar operaciones aritméticas consideradas sencillas.

1.2 Alcance

Se espera que esta calculadora sea usada por usuarios que necesiten realizar alguna operación matemática, las calculadoras sencillas tratan de agilizar estas tareas pequeñas con el objetivo de ahorrar tiempo en operaciones simples.

1.3 Definiciones

Es conveniente declarar algunas definiciones y acrónimos presentados en la documentación del software, con el objetivo de llegar a una correcta interpretación y evitar confusiones.

termino	Definición
Arquitectura de software	Conjunto de elementos estáticos, propios del diseño Intelectual del sistema, que definen y dan forma al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto
Caso de uso	Secuencia del acciones que realiza el sistema, la cual proporciona un resultado de valores observable
Diseño	Actividad creativa que tiene por fin proyectar objetos después de fabricarlos
Sistema operativo	Software que actúa de interfaz entre los dispositivos de hardware y los

	programas usados por el usuario para utilizar un computador
IDE	Entorno de Desarrollo Integrado (Integrated Development Environment).
GUI	Interfaz Gráfica de Usuario (Graphical User Interface).
display	Pantalla que muestra algo del software.

1.4 Referencias

- Java SE Documentation: <https://docs.oracle.com/javase/8/docs/>
- NetBeans IDE Documentation: <https://netbeans.apache.org/kb/docs/java/index.html>

2. Descripción General del Sistema

2.1. Perspectiva del Producto

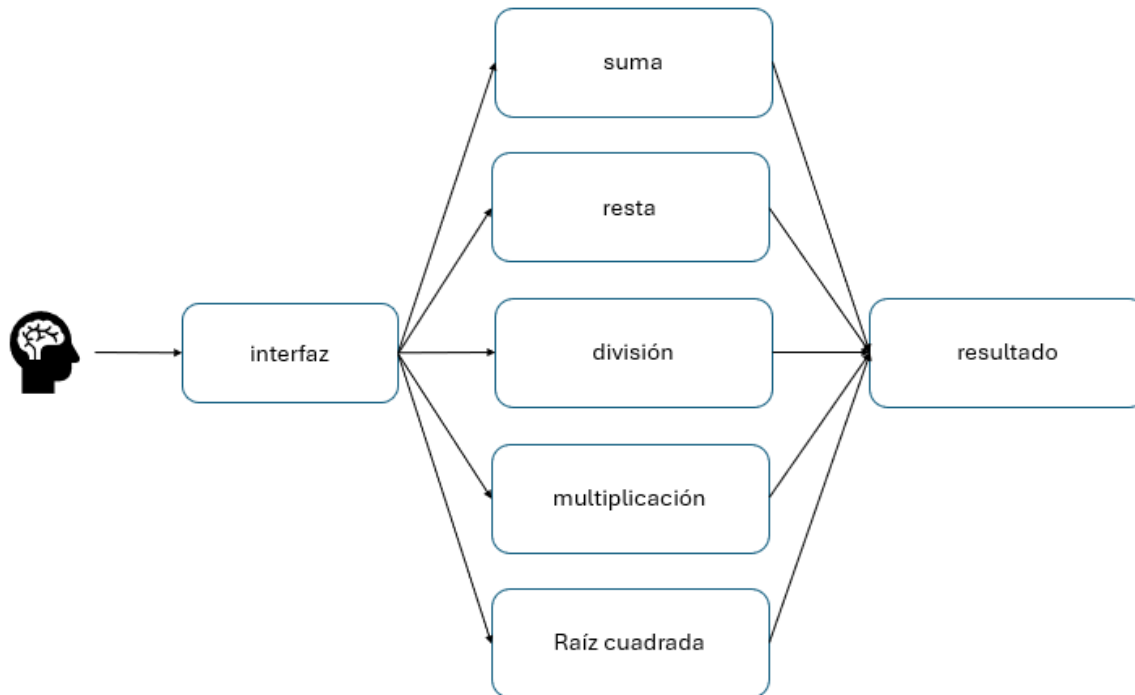
La calculadora es una aplicación de escritorio que proporciona una interfaz gráfica para realizar cálculos matemáticos básicos. Está implementada en Java y utiliza las bibliotecas Swing para la creación de la GUI.

2.2. Funcionalidades del Sistema

- **Operaciones Básicas:** Suma, resta, multiplicación, división y raíz cuadrada.
- **Interfaz de Usuario:** Botones para cada dígito y operación, así como un display para mostrar los resultados.
- **Manejo de Errores:** Detección de divisiones por cero y manejo de errores de entrada.

2.3. Usuarios y Casos de Uso

- **Usuarios:** Cualquier persona que necesite realizar cálculos matemáticos sencillos.
- **Casos de Uso:** Ingresar números, seleccionar una operación y obtener el resultado.



3. Requisitos del Sistema

3.1. Requisitos Funcionales

- El sistema debe permitir al usuario ingresar dos números.
- El sistema debe permitir realizar operaciones de suma, resta, multiplicación y división.
- El sistema debe mostrar el resultado de la operación.
- El sistema debe manejar errores como división por cero.
- El sistema debe realizar raíz cuadrada.

3.2. Requisitos No Funcionales

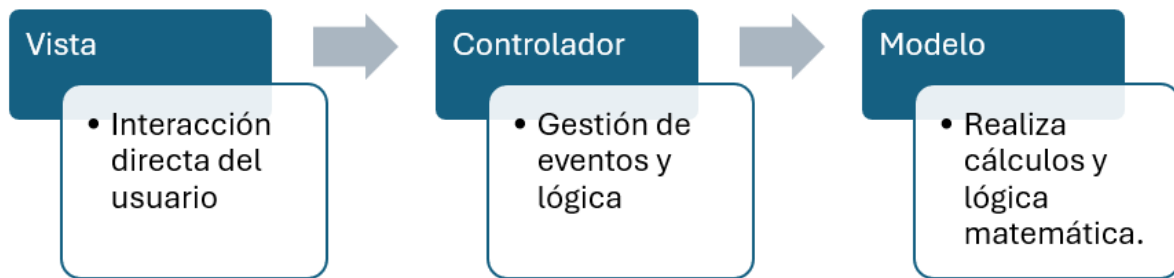
- **Usabilidad:** La interfaz debe ser intuitiva y fácil de usar.
- **Rendimiento:** La calculadora debe responder rápidamente a las entradas del usuario.
- **Portabilidad:** El sistema debe ser ejecutable en cualquier máquina que soporte Java SE.

4. Diseño de la Arquitectura

4.1. Diagrama de Componentes

El sistema se divide en los siguientes componentes:

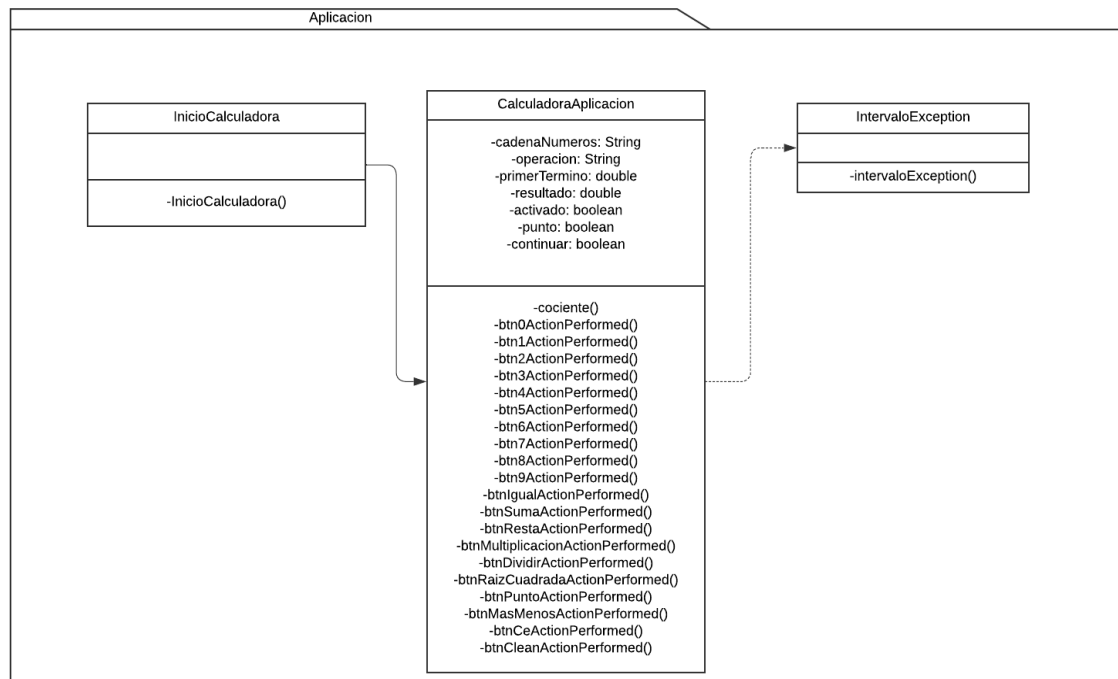
- **Controlador (Controller):** Gestiona la lógica de la aplicación y responde a las interacciones del usuario.
- **Vista (View):** Representa la interfaz gráfica de usuario (GUI), utilizando componentes de Swing.
- **Modelo (Model):** Encapsula los datos y la lógica de negocio, como la realización de cálculos



4.2. Diagrama de Clases

Las principales clases del sistema son:

- **CalculadoraAplicacion:** Gestiona la interfaz de usuario.
- **IntervaloExcepcion:** Gestiona la lógica de negocio (cálculos matemáticos).
- **CalculadorInicio:** Conecta la vista con el modelo y gestiona las interacciones del usuario.



4.3. Patrones de Diseño

- **MVC (Modelo-Vista-Controlador):** Separa la lógica de negocio, la interfaz de usuario y el controlador que las conecta.
- **Observador:** El controlador observa la vista para responder a eventos del usuario.

5. Implementación

5.1. Herramientas de Desarrollo

- **Lenguaje:** Java 8 o superior.
- **IDE:** NetBeans IDE 12.0 o superior.
- **Bibliotecas:** Swing para la interfaz gráfica.

5.2. Estrategia de Desarrollo

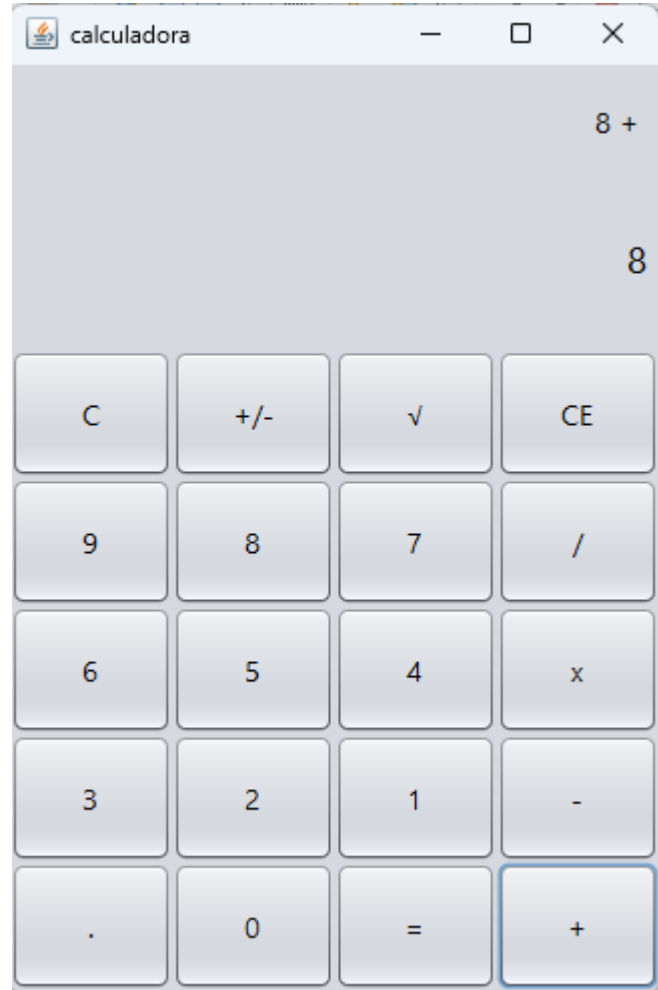
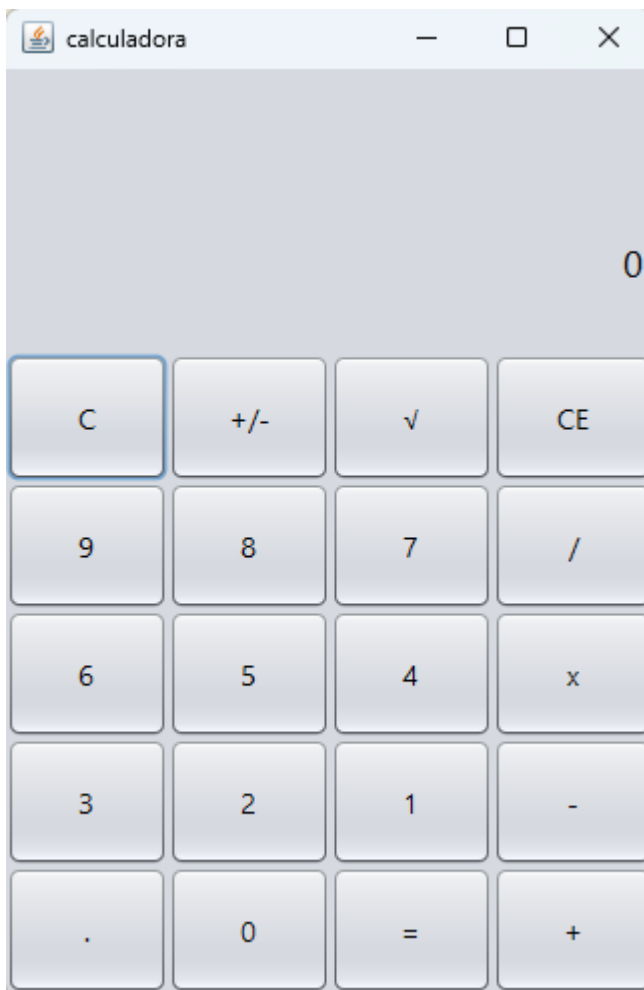
- **Iteración 1:** Implementar la interfaz gráfica (Vista).
- **Iteración 2:** Implementar la lógica de negocio (Modelo).
- **Iteración 3:** Conectar la vista y el modelo a través del controlador.
- **Iteración 4:** Realizar pruebas unitarias y de integración.

6. Pruebas

6.1. Plan de Pruebas

- **Pruebas Unitarias:** Verificar que cada operación en CalculadoraAplicacion funciona correctamente.
- **Pruebas de Integración:** Asegurar que la vista y el modelo interactúan correctamente a través del controlador.
- **Pruebas de Usuario:** Validar la usabilidad de la interfaz con usuarios reales.

6.2 Capturas de Pantalla



7. Despliegue

7.1. Requisitos del Entorno

- **Java Runtime Environment (JRE):** Versión 8 o superior.
- **Sistema Operativo:** Windows.

7.2. Proceso de Despliegue

- Compilar el código en NetBeans.

8. Mantenimiento

8.1. Gestión de Versiones

- Uso de Git para el control de versiones.

8.2. Plan de Mantenimiento

- Monitoreo de errores reportados por los usuarios.
- Actualización de la aplicación para corregir errores y mejorar funcionalidades.