

Universidad de Costa Rica

Seguridad de Sistemas Computacionales

CI0143 - I Semestre 2025

Calculadora con operandos en texto

Desarrollo y seguridad de una aplicación de software

Fecha entrega: 12 abril 2024 (Parte I)

Prof. Ricardo Villalón Fonseca

Estudiantes:

Alessandro Merino López

Tonny Ortiz Salazar

### 1. Link de Github al proyecto:

<https://github.com/TonnyOrtiz/CalculadoraTextoSeguridad>

### 2. Identificación y Vista General

La entidad de interés es la aplicación de calculadora de texto que interpreta expresiones matemáticas escritas en lenguaje natural con el propósito de evaluar la seguridad de la misma en dos versiones distintas. La primera versión será sin estructurar y la segunda será estructurada utilizando el modelo de Vista-Controlador. Se hará una comparación entre ambas y se identificarán ya sean ventajas o desventajas entre ellas.

### 3. Partes Interesadas

Parte Interesada	Rol	Intereses clave
Estudiantes	Equipo de desarrollo	lograr identificar la ventajas de utilizar seguridad estructural sobre no utilizarla.
Profesor	Supervisor académico	Que los estudiantes comprendan las implicaciones y los beneficios en la aplicación de la seguridad estructural en sistemas de software.

### 4. Perspectivas de las Partes Interesadas

Perspectiva	Descripción
Estudiantes	Analiza, diseña, desarrolla y prueba el sistema, a su vez que analiza el problema mientras aprende sobre los beneficios de la seguridad estructural.
Profesor	Analiza si el sistema promueve el aprendizaje de conceptos como seguridad estructural y buenas prácticas de diseño.

## 5. Preocupaciones

Preocupación	Parte interesada	Justificación
Precisión de los cálculos	Estudiantes	Resultados correctos
Modelo estructurado	Estudiantes, Profesor	Seguridad estructural
Aprendizaje	Profesor	Comprensión de la seguridad estructural

## 6. Aspectos

Aspecto	Descripción	Preocupación relacionada
Funcional	Analizar y calcular expresiones matemáticas	Interpretación y precisión en los cálculos
Estructural	Modelo Vista-Controlador con varios módulos	Mantenibilidad y extensibilidad
Visual	Mostrar los resultados de los cálculos	Claridad

## 7. Puntos de Vista Arquitectónicos

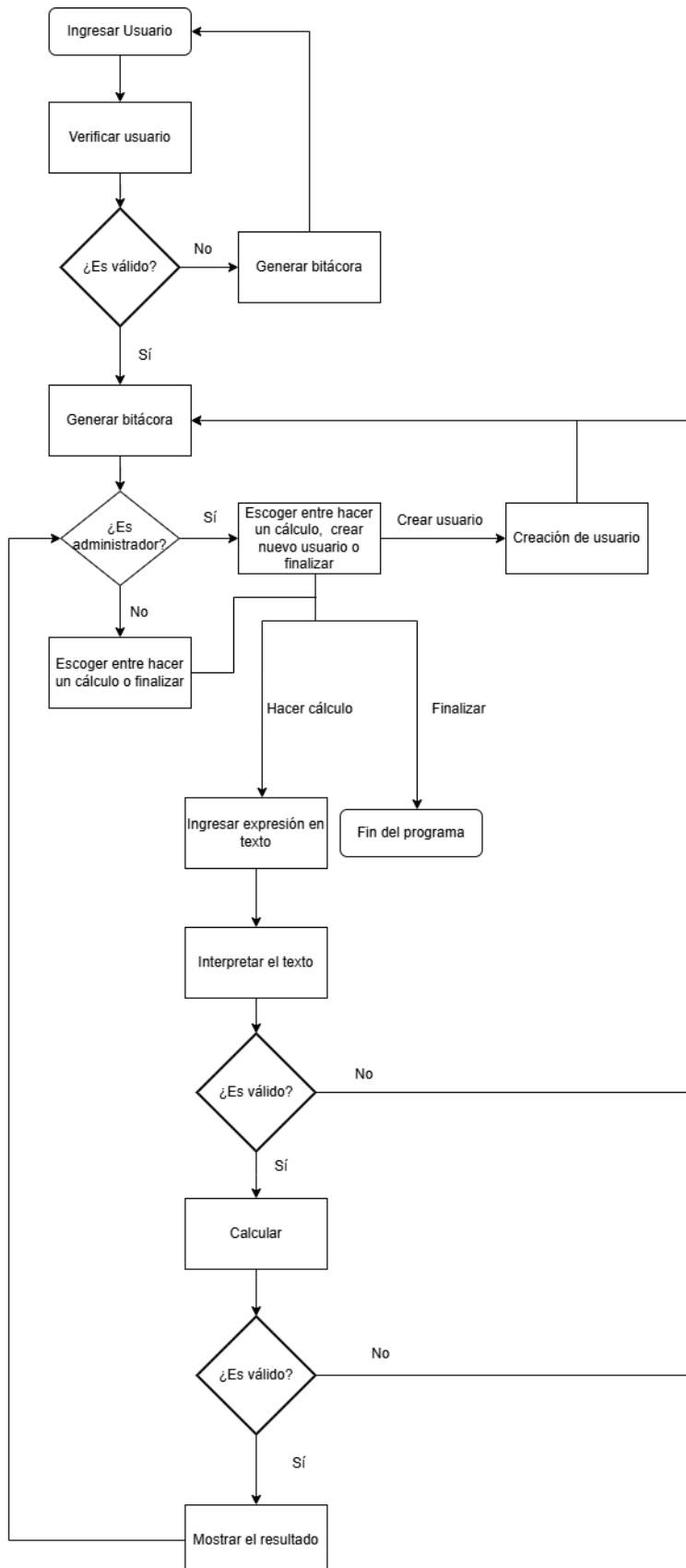
Punto de vista	Propósito	Aspectos	Tipo de modelo
Vista funcional	Mostrar cómo se realiza el procesamiento del texto	Funcional	Diagrama de flujo
Vista de componentes	Mostrar los módulos del sistema	Estructural	Diagrama de componentes

## 8. Vistas Arquitectónicas

Nombre de la vista	Vista aplicada	Contenido
Procesamiento lógico	Vista funcional	Describe cómo el texto se transforma en un resultado.
Componentes internos	Vista de componentes	Muestra los módulos que componen la aplicación

## 9. Componentes de Vista

### Diagrama de flujo



## 10. Correspondencias

Elementos relacionados	Tipo de correspondencia	Método de verificación
Entrada de texto ↔ Módulo de análisis semántico	Trazabilidad	Revisión del flujo desde entrada hasta resultado
Módulo parser ↔ Controlador (MVC)	Composición	Diagrama de componentes
Cálculo ↔ Vista de resultados	Dependencia funcional	Pruebas funcionales
Vista funcional ↔ Preocupación "Precisión"	Consistencia	Evaluación de resultados esperados
Componentes estructurados ↔ Seguridad estructural	Seguridad	Comparación entre versiones estructurada y no estructurada

## 11. Decisiones y Razonamiento

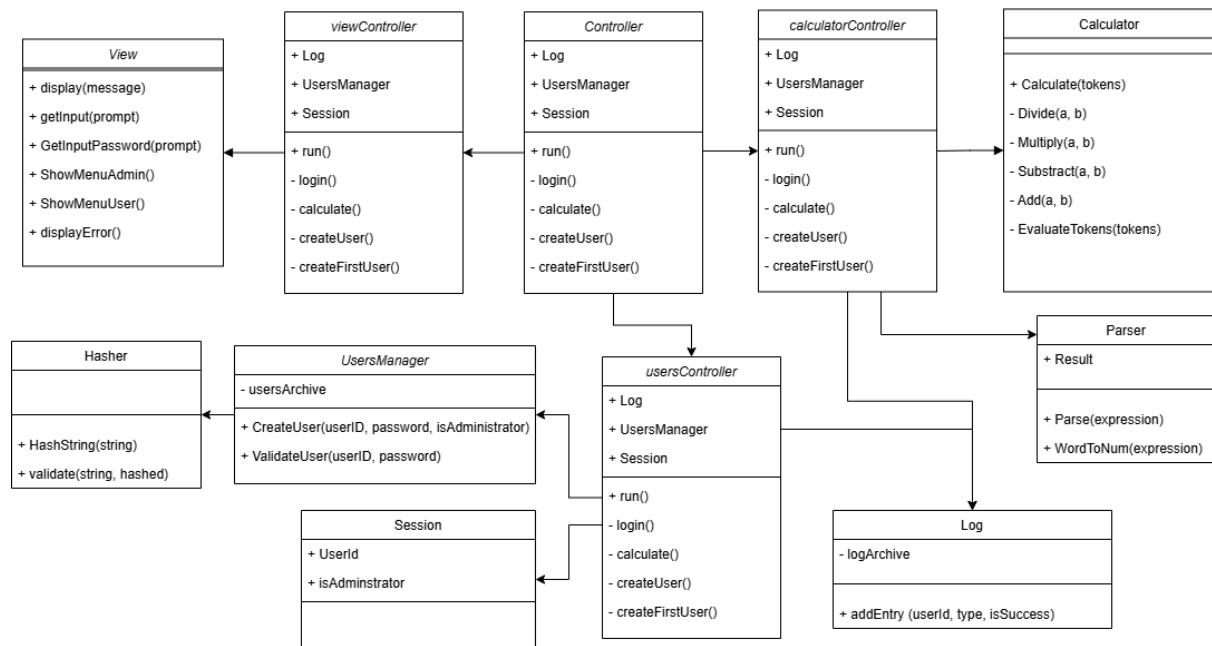
Decisión	Razonamiento	Impacto
Utilizar MVC	Facilita la separación de responsabilidades y permite observar el efecto de la estructura sobre la seguridad y mantenibilidad	Mejora la claridad del diseño, facilita futuras extensiones y el control del flujo de datos
Parser independiente de la calculadora	Especializar al parser en convertir expresiones en texto a tokens para que la calculadora solo se preocupe por calcular.	Separación de responsabilidades y mejora la mantenibilidad.

## 12. Comparación entre ambas versiones

Aspecto	Estructurada	No estructurada
Mantenibilidad	Al estar todo bien modularizado y las responsabilidades asignadas a las clases que les corresponde es sencillo comprender el código lo que facilita hacer modificaciones como nuevos métodos de seguridad.	Al tener todo el código en un mismo archivo main se vuelve más complicado llevar el hilo del programa, lo que complica comprenderlo y a la misma vez hacer modificaciones.
El manejo de la sesión	La clase Session se encarga de llevar los datos de la sesión como los permisos y el usuario por lo que con tener esta clase protegida disminuye el riesgo de que estos datos sean modificados por terceros.	Al tener variables globales para los datos de la sesión es una vulnerabilidad ya que podría prestarse para la escala de privilegios.
Facilidad de realizar pruebas	Como cada funcionalidad está separada por componente es muy sencillo probarlas.	Debido a que toda la funcionalidad está en un mismo archivo se vuelve difícil probar funciones individualmente.
Superficie de ataque	Dado que el sistema se separa en diversos archivos se aíslan los problemas de una clase de las demás.	A consecuencia de que todo el código está en un mismo archivo la superficie de ataque es absurdamente grande lo que significa que un solo ataque ya comprometió a todo el sistema.

## 13. Diagramas

### Diagrama de componentes UML de la versión Estructurada.



### Diagrama de version no Estructurada

No posee diagrama porque se ha hecho todo en el mismo script del main