

Proyecto: ADT Stack

Asignatura:	ICC735 Pruebas de Software
Profesor:	Ivan Zuñiga Valenzuela
Autor:	Camila Infante Benjamín Serrano
Documento	Versión 1.

Contenido

Tabla de contenido

Introducción.....	3
Descripción General.....	3
Objetivo del Proyecto.....	3
Metodología.....	3
Funcionalidades Principales.....	3
Elicitación.....	4
Stakeholders.....	4
Lista de requisitos.....	4
Restricciones de aplicación.....	4
Análisis.....	5
Historias de Usuario.....	5
Diccionario de datos (Glosario).....	6
Componentes principales.....	6
BackLog.....	7
Gestión de Requisitos.....	8
Sprint BackLog 1 (Básico).....	8
Sprint Backlog 2 (Errores y Mejoras).....	8
Estándares de Codificación acordados.....	9
Supuestos.....	9
Restricciones.....	9
Nomenclatura.....	10

Introducción

Descripción General

Un tipo de dato abstracto (TDA) es un modelo matemático, junto con varias operaciones definidas sobre ese modelo. Los TDA se representan en función de los tipos de datos y los operadores manejados por ese lenguaje.

Un stack es un tipo especial de la lista en que todas las inserciones y borrados toman lugar en un sólo lugar llamado tope.

Objetivo del Proyecto

Implementar el TDA Stack, que permita llevar a cabo el funcionamiento de una calculadora RPN. Esta calculadora es del tipo HP. La calculadora clásica es de tipo notación infija en que una expresión $(1-2) * (4+5) =$ se recorre en forma serial y requiere paréntesis.

Esta expresión en notación polaca inversa (RPN) es $12-45+*=$. El signo $=$ es opcional, los paréntesis no son necesarios.

Metodología

Se utilizará *extreme programming* como metodología de desarrollo

Funcionalidades Principales

Funcionalidad	Descripción
Ingresar	Ingresa un valor u operador al stack para calcular posteriormente
Calcular	Realiza operaciones de expresiones simples y complejas

Elicitación

Stakeholders

Las partes interesadas se identifican en la siguiente tabla

Cargo	Cod.	Descripción
Product owner	01	Persona que utilizará la calculadora RPN para realizar operaciones matemáticas.

Lista de requisitos

ID	Nombre	Descripción
01	Implementación Stack	El sistema debe implementar el TDA Stack con operaciones básicas.
02	Evaluación RPN	El sistema debe evaluar expresiones en notación polaca inversa (RPN).
03	Soporte Operadores	El sistema debe soportar operaciones aritméticas básicas (+, -, *, /)

Restricciones de aplicación

- La entrada de la calculadora RPN debe consistir solo en números y operadores válidos.
 - No se pueden realizar operaciones con menos de dos operadores.
 - El resultado final debe quedar en el tope del stack al finalizar la evaluación.
 - Los números deben ser reales.
 - Los operadores válidos son: suma (+), resta (-), multiplicación (*), y división (/).
 - La expresión debe seguir la notación polaca inversa.
-

Análisis

Historias de Usuario

A través de la siguiente tabla se muestran las historias de usuarios...

#HU	Prioridad	Nombre HU	Como	Quiero	Para	Criterio Aceptación
01	Alta	Ingreso de números	Usuario	Ingresar números	Realizar operaciones aritméticas	El sistema debe permitir ingresar números positivos o negativos
02	Alta	Operaciones aritméticas	Usuario	Ingresar operadores	Realizar operaciones aritméticas	El sistema debe permitir ingresar operadores básicos (+, -, *, /)
03	Alta	Sumar	Usuario	Sumar números	Obtener la suma	El sistema debe permitir que se devuelva el valor correcto de la suma
04	Alta	Restar	Usuario	Restar números	Obtener la diferencia	El sistema debe permitir que se devuelva el valor correcto de la resta
05	Alta	Multiplicar	Usuario	Multiplicar números	Obtener el producto	El sistema debe permitir que se devuelva el valor correcto de la multiplicación
06	Alta	Dividir	Usuario	Dividir números	Obtener cociente	La calculadora devuelve correctamente la división y maneja divisiones por 0
07	Alta	Manejo de errores	Usuario	Que la calculadora indique una entrada no válida	Corregir la operación y continuar	El sistema valida las entradas y no se detiene al recibir una entrada no válida
08	Media	Resultado final	Usuario	Ver el resultado final	Obtener el resultado	El resultado es visible por el usuario
09	Media	Evaluación expresiones complejas	Usuario	Evaluar expresiones complejas en RPN	Realizar operaciones complejas	La calculadora resuelve expresiones complejas correctamente
10	Media	Resultados decimales	Usuario	Ingresar números decimales	Operar números decimales	El sistema debe permitir y calcular correctamente operaciones que incluyan números decimales

Diccionario de datos (Glosario)

Término	Descripción
Extreme programming	Metodología de programación en parejas donde una persona supervisa a la otra en ciclos de trabajo intercalados y ambos trabajan el mismo código y se turnan los roles.
TDD	Test-Driven Development. Enfoque de desarrollo donde primero se escriben las pruebas antes de la implementación del código.
Stack (Pila)	Estructura de datos donde los elementos se apilan y desapilan siguiendo el principio LIFO (Last In, First Out) .
RPN	Reverse Polish Notation. Forma de notación matemática en la que los operadores se colocan después de los operandos, eliminando la necesidad de paréntesis.

Componentes principales

A través de la siguiente imagen se muestran los principales componentes identificados.

- **TDA Stack:** Implementación de una pila para almacenar operandos y resultados.
 - **Calculadora RPN:** Motor de evaluación de expresiones en notación polaca inversa.
 - **Interfaz de Usuario (CLI o GUI):** Permite la interacción con el usuario para ingresar expresiones y visualizar resultados.
 - **Gestor de Errores:** Valida las entradas y evita fallos en la ejecución.
-

BackLog

ID	Tipo Req. CO/FR/ NFR/DO	Ver sió n	Descripción	Autor	Prioridad A/M/B	Estabilidad F/SF/C	Estado C/A
01	FR	1.0	Implementar el TDA Stack con operaciones básicas	DevTeam	A	F	A
02	FR	1.0	Evaluar expresiones en notación polaca inversa (RPN)	DevTeam	A	F	A
03	FR	1.0	Soportar operaciones aritméticas básicas (+, -, *, /)	DevTeam	A	F	A
04	FR	1.0	Manejo de errores en entradas inválidas	DevTeam	A	F	A
05	FR	1.0	Manejo de división por cero	DevTeam	A	F	A
06	FR	1.0	Permitir números decimales en las operaciones	DevTeam	M	SF	A
07	FR	1.0	Mostrar el resultado final en el tope del stack	DevTeam	M	F	A

Gestión de Requisitos

Sprint BackLog 1 (Básico)

ID	Test ID	Descripción (condición de satisfacción)	Comentario del Test	UAT	QA	Prod	Otro	Estado
01	T01	El Stack permite ingresar y extraer datos	Se insertan y eliminan elementos de la pila correctamente	OK	OK	-	-	Abierto
02	T02	La calculadora evalúa expresiones simples en RPN	Se evalúa "3 4 +", resultado esperado: 7	OK	OK	-	-	Abierto
03	T03	La calculadora evalúa expresiones complejas	Se evalúa "5 1 2 + 4 * + 3 -", resultado esperado: 14	OK	OK	-	-	Abierto

Sprint Backlog 2 (Errores y Mejoras)

ID	Test ID	Descripción (condición de satisfacción)	Comentario del Test	UAT	QA	Prod	Otro	Estado
04	T04	Manejo de errores en entradas inválidas	Se introduce "3 4 &", debe mostrar error	OK	OK	-	-	Abierto
05	T05	Manejo de división por cero	Se evalúa "10 0 /", debe mostrar error	OK	OK	-	-	Abierto
06	T06	Permitir números decimales	Se evalúa "2.5 3.2 +", resultado esperado: 5.7	OK	OK	-	-	Abierto
07	T07	Mostrar el resultado final en el tope del stack	Se evalúa "4 5 +", el stack debe contener solo "9"	OK	OK	-	-	Abierto

Estándares de Codificación acordados

Lenguaje: Java

Framework de pruebas: JUnit

Principios: Clean Code, SOLID

Control de Versiones: GitHub

Supuestos

Se asume que el usuario ingresará expresiones en formato válido (se validarán errores).
La calculadora operará solo en modo consola en la primera versión.

Restricciones

No se permite usar bibliotecas externas para el Stack.
Se debe seguir TDD para el desarrollo.
Se debe desarrollar con metodología XP.

Nomenclatura

Símbolo	Significado
CO/FR/NFR/DO	Comercial/Functional Requirement/Non Functional Requirement/Dominio
A/M/B	Alta/Media/Baja
F/SF/C	Fijo/SemiFijo/Cambiante
C/A	Cerrado/Abierto para la columna Estado
OK/NOK	Valores para las columnas UAT, QA, Prod y Fail
