



Pontificia Universidad
JAVERIANA
Colombia

ESTRUCTURAS DE DATOS

PROYECTO: ENTREGA 01

PRESENTADO POR:

SEBASTIÁN CAMILO HERNANDO MURCIA SIERRA

JUAN DIEGO ARIAS DURAN

DOCENTE:

ANDREA RUEDA

2024 - 1

ANÁLISIS DE ENTRADAS, SALIDAS Y CONDICIONES DE COMANDOS

Comando	Entrada	Salida	Condiciones
<u>inicializar</u>	<i>nombre_archivo</i>	“El diccionario <i>nombre_archivo</i> se ha inicializado correctamente”	- Archivo existe - Diccionario normal no haya sido inicializado - Palabras válidas
<u>iniciar inverso</u>	<i>nombre_archivo</i>	“El diccionario <i>nombre_archivo</i> se ha inicializado correctamente”	- Archivo existe - Diccionario inverso no haya sido inicializado - Palabras válidas
<u>puntaje</u>	<i>palabra</i>	“La <i>palabra</i> tiene un puntaje de <i>puntaje</i> ”	- Palabra existe - Palabra válida
<u>salir</u>	(No tiene entradas)	“Saliendo del programa...”	- No hay condiciones

DESCRIPCIÓN DE TADS

TAD Diccionario
<p>Conjunto mínimo de datos:</p> <ul style="list-style-type: none"> • palabras, vector de cadena de caracteres, representa las palabras en el diccionario. • palabrasInversas, vector de cadena de caracteres, representa las palabras del diccionario escritas al inverso.
<p>Comportamiento (operaciones):</p> <ul style="list-style-type: none"> • obtenerPalabras(), retorna las palabras. • verificarPalabra(palabra), recibe una palabra y valida que no contenga guiones, números y signos de puntuación. • agregarPalabra(palabra), recibe una palabra para agregarla al diccionario.

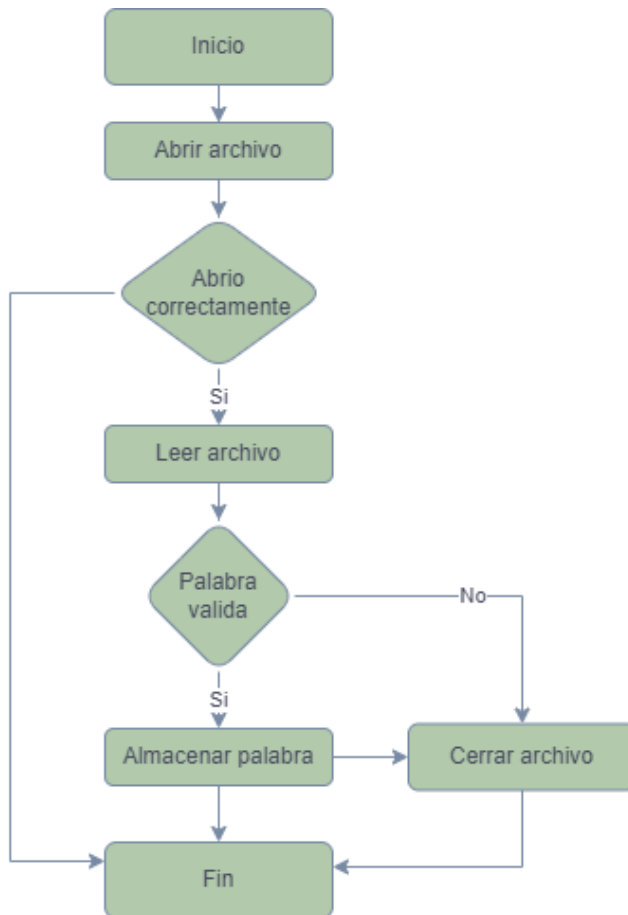
- `iniDiccionario(diccionario)`, recibe el diccionario y guarda las palabras que contiene en el vector llamado `palabras`.
- `iniDiccionarioInverso(diccionario)`, recibe el diccionario, invierte las palabras que contiene y las almacena en el vector llamado `palabrasInversas`.
- `puntuarPalabra(palabra)`, recibe una palabra, verifica que exista tanto en diccionario normal como inverso, posterior, calcula el puntaje de la palabra y retorna el puntaje.

- **Diagrama de relación entre TADs:**

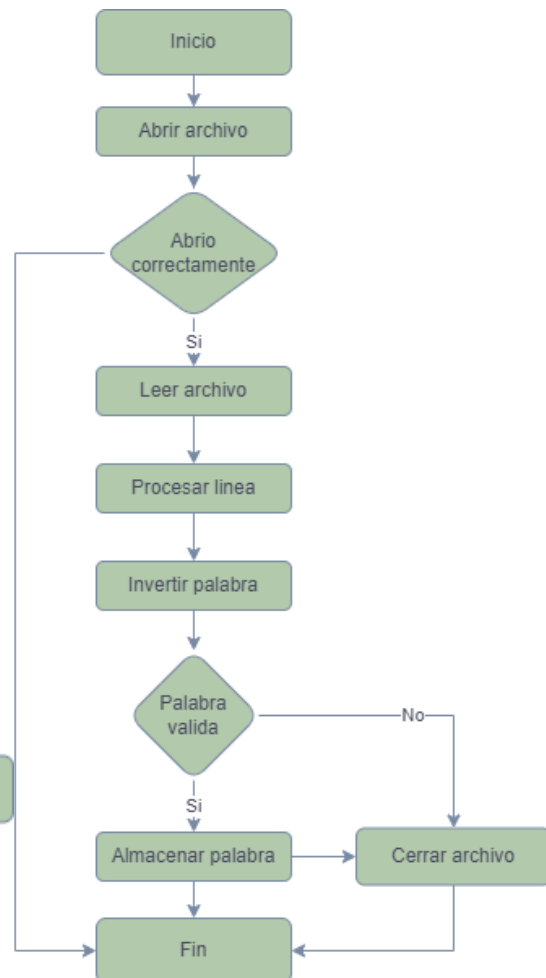
Diccionario
-palabras -palabrasInversas
+obtenerPalabras() +verificarPalabra(palabra) +agregarPalabra(palabra) +iniDiccionario(diccionario) +iniDiccionarioInverso(diccionario) +puntuarPalabra(palabra)

ESQUEMÁTICOS DE OPERACIONES

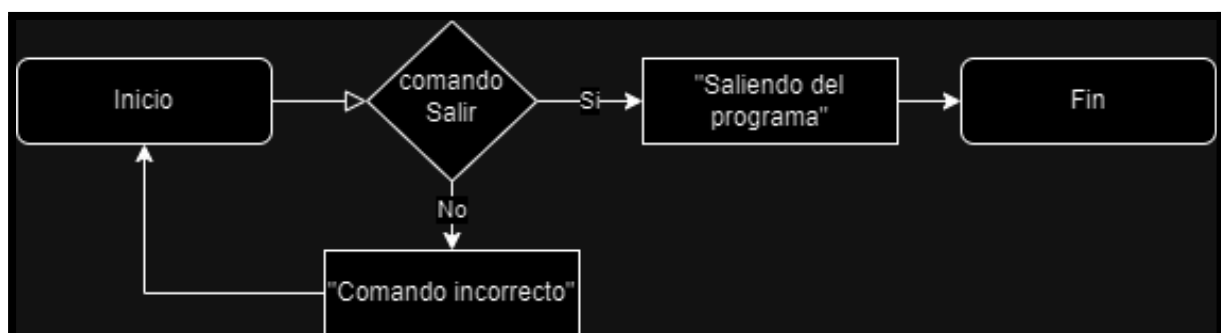
inicializar:



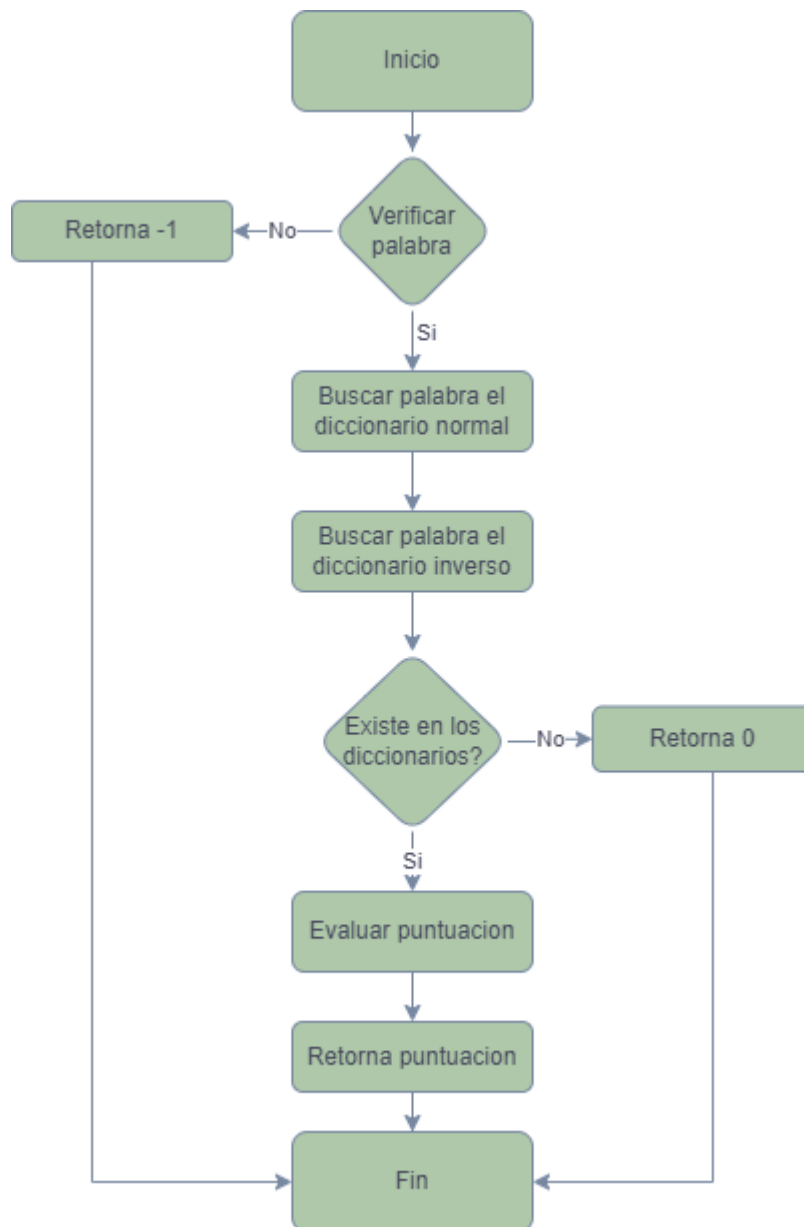
turno:



salir:



puntaje:



PLAN DE PRUEBAS

puntaje:

• Datos de entrada
<i>palabra</i> = abandon
Proceso
El comando permite conocer la puntuación que puede obtenerse con una <i>palabra</i> dada, de acuerdo a la tabla de puntuación. Sin embargo, el comando debe verificar que la palabra sea válida, es decir, que exista en el diccionario original o inverso, y que esté escrita con símbolos válidos.
Resultados
Se imprime en pantalla: “La palabra <i>abandon</i> tiene un puntaje de 10”

• Datos de entrada
<i>palabra</i> = wag
Proceso
El comando permite conocer la puntuación que puede obtenerse con una <i>palabra</i> dada, de acuerdo a la tabla de puntuación. Sin embargo, el comando debe verificar que la palabra sea válida, es decir, que exista en el diccionario original o inverso, y que esté escrita con símbolos válidos.
Resultados
Se imprime en pantalla: “La palabra <i>wag</i> tiene un puntaje de 7”

• Datos de entrada
<i>palabra</i> = zoo
Proceso
El comando permite conocer la puntuación que puede obtenerse con una <i>palabra</i> dada, de acuerdo a la tabla de puntuación. Sin embargo, el comando debe verificar que la palabra sea válida, es decir, que exista en el diccionario original o inverso, y que esté escrita con símbolos válidos.
Resultados
Se imprime en pantalla: “La palabra <i>zoo</i> tiene un puntaje de 12”

Tabla comparativa de resumen:

Caso	Entradas	Resultado esperado	Resultado obtenido
Prueba 1	abandon	“La palabra <i>abandon</i> tiene un puntaje de 10”	“La palabra <i>abandon</i> tiene un puntaje de 10”
Prueba 2	wag	“La palabra <i>wag</i> tiene un puntaje de 7”	“La palabra <i>wag</i> tiene un puntaje de 7”
Prueba 3	zoo	“La palabra <i>zoo</i> tiene un puntaje de 12”	“La palabra <i>zoo</i> tiene un puntaje de 12”