

PRUEBAS

Pruebas Cesar

Configuración de los Escenarios

Nombre	Clase	Escenario
setup1	CesarTest	Un objeto de la clase Cesar con track= 1 y direction= "R"
setup2	CesarTest	Un objeto de la clase Cesar con track= 1 y direction= "L"
setup3	CesarTest	Un objeto de la clase Cesar con track= 54 y direction= "R"
setup4	CesarTest	Un objeto de la clase Cesar con track= 54 y direction= "L"
setup5	CesarTest	Un objeto de la clase Cesar con track= 100 y direction= "R"
Setup6	CesarTest	Un objeto de la clase Cesar con track= 100 y direction= "L"

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método encrypt de la clase Cesar funcione correctamente, encriptando un texto según el track y la direction dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Cesar	encrypt	setup1	txt = "a"	El mensaje encriptado es "b"
Cesar	encrypt	setup1	txt = "z"	El mensaje encriptado es "A"
Cesar	encrypt	setup1	txt = "Z"	El mensaje encriptado es "a"
Cesar	encrypt	setup1	txt = "hola como estas"	El mensaje encriptado es "ipmb dpnp ftubt"
Cesar	encrypt	setup1	txt = "Hola, ¿como estas?"	El mensaje encriptado es "ipmb, ¿dpnp ftubt?"
Cesar	encrypt	setup2	txt = "a"	El mensaje encriptado es "Z"
Cesar	encrypt	setup2	txt = "z"	El mensaje encriptado es "y"
Cesar	encrypt	setup2	txt = "Z"	El mensaje encriptado es "Y"
Cesar	encrypt	setup2	txt = "hola como estas"	El mensaje encriptado es "gñkZ bñlñ drsZr"
Cesar	encrypt	setup2	txt = "Hola, ¿como estas?"	El mensaje encriptado es "GñkZ, ¿bñlñ drsZr?"
Cesar	encrypt	setup3	txt = "a"	El mensaje encriptado es "a"
Cesar	encrypt	setup3	txt = "z"	El mensaje encriptado es "z"
Cesar	encrypt	setup3	txt = "Z"	El mensaje encriptado es "Z"
Cesar	encrypt	setup3	txt = "hola como estas"	El mensaje encriptado es "hola como estas"
Cesar	encrypt	setup3	txt = "Hola, ¿como estas?"	El mensaje encriptado es "Hola, ¿como estas?"
Cesar	encrypt	setup4	txt = "a"	El mensaje encriptado es "a"
Cesar	encrypt	setup4	txt = "z"	El mensaje encriptado es "z"
Cesar	encrypt	setup4	txt = "Z"	El mensaje encriptado es "Z"
Cesar	encrypt	setup4	txt = "hola como estas"	El mensaje encriptado es "hola como estas"

Cesar	encrypt	setup4	txt = "Hola, ¿como estas?"	El mensaje encriptado es "Hola, ¿como estas?"
Cesar	encrypt	setup5	txt = "a"	El mensaje encriptado es "S"
Cesar	encrypt	setup5	txt = "z"	El mensaje encriptado es "r"
Cesar	encrypt	setup5	txt = "Z"	El mensaje encriptado es "R"
Cesar	encrypt	setup5	txt = "hola como estas"	El mensaje encriptado es "ZhdS Uheh WlmSI"
Cesar	encrypt	setup5	txt = "Hola, ¿como estas?"	El mensaje encriptado es "zhdS, ¿Uheh WlmSI?"
Cesar	encrypt	setup6	txt = "a"	El mensaje encriptado es "i"
Cesar	encrypt	setup6	txt = "z"	El mensaje encriptado es "H"
Cesar	encrypt	setup6	txt = "Z"	El mensaje encriptado es "h"
Cesar	encrypt	setup6	txt = "hola como estas"	El mensaje encriptado es "owsi kwtw mABiA"
Cesar	encrypt	setup6	txt = "Hola, ¿como estas?"	El mensaje encriptado es "Owsi, ¿kwtw mABiA?"

Objetivo de la Prueba: Verificar que el método decrypt de la clase Cesar funcione correctamente, desencriptando un texto según el track y la direction dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Cesar	decrypt	setup1	txt = "b"	El mensaje desencriptado es "a"
Cesar	decrypt	setup1	txt = "A"	El mensaje desencriptado es "z"
Cesar	decrypt	setup1	txt = "a"	El mensaje desencriptado es "Z"
Cesar	decrypt	setup1	txt = "ipmb dpnp ftubt"	El mensaje desencriptado es "hola como estas"
Cesar	decrypt	setup1	txt = "ipmb, ¿dpnp ftubt?"	El mensaje desencriptado es "Hola, ¿como estas?"
Cesar	decrypt	setup2	txt = "Z"	El mensaje desencriptado es "a"
Cesar	decrypt	setup2	txt = "y"	El mensaje desencriptado es "z"
Cesar	decrypt	setup2	txt = "Y"	El mensaje desencriptado es "Z"
Cesar	decrypt	setup2	txt = "gñkZ bñlñ drsZr"	El mensaje desencriptado es "hola como estas"
Cesar	decrypt	setup2	txt = "GñkZ, ¿bñlñ drsZr?"	El mensaje desencriptado es "Hola, ¿como estas?"
Cesar	decrypt	setup3	txt = "a"	El mensaje desencriptado es "a"
Cesar	decrypt	setup3	txt = "z"	El mensaje desencriptado es "z"
Cesar	decrypt	setup3	txt = "Z"	El mensaje desencriptado es "Z"
Cesar	decrypt	setup3	txt = "hola como estas"	El mensaje desencriptado es "hola como estas"
Cesar	decrypt	setup3	txt = "Hola, ¿como estas?"	El mensaje desencriptado es "Hola, ¿como estas?"
Cesar	decrypt	setup4	txt = "a"	El mensaje desencriptado es "a"
Cesar	decrypt	setup4	txt = "z"	El mensaje desencriptado es "z"
Cesar	decrypt	setup4	txt = "Z"	El mensaje desencriptado es "Z"

Cesar	decrypt	setup4	txt = "hola como estas"	El mensaje descriptado es "hola como estas"
Cesar	decrypt	setup4	txt = "Hola, ¿como estas?"	El mensaje descriptado es "Hola, ¿como estas?"
Cesar	decrypt	setup5	txt = "S"	El mensaje descriptado es "a"
Cesar	decrypt	setup5	txt = "r"	El mensaje descriptado es "z"
Cesar	decrypt	setup5	txt = "R"	El mensaje descriptado es "Z"
Cesar	decrypt	setup5	txt = "ZhdS Uheh WImSI"	El mensaje descriptado es "hola como estas"
Cesar	decrypt	setup5	txt = "zhdS, ¿Uheh WImSI"	El mensaje descriptado es "Hola, ¿como estas?"
Cesar	decrypt	setup6	txt = "i"	El mensaje descriptado es "a"
Cesar	decrypt	setup6	txt = "H"	El mensaje descriptado es "z"
Cesar	decrypt	setup6	txt = "h"	El mensaje descriptado es "Z"
Cesar	txt = "i"	setup6	txt = "owsi kwtw mABiA"	El mensaje descriptado es "hola como estas"
Cesar	decrypt	setup6	txt = "Owsi, ¿kwtw mABiA?"	El mensaje descriptado es "Hola, ¿como estas?"

Pruebas Atbash

Configuración de los Escenarios

Nombre	Clase	Escenario
setup1	AtbashTest	Un objeto de la clase Atbash

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método encrypt de la clase Cesar funcione correctamente, encriptando un texto según el track y la direction dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Atbash	encrypt	setup1	txt = "a"	El mensaje encriptado es "z"
Atbash	encrypt	setup1	txt = "n"	El mensaje encriptado es "n"
Atbash	encrypt	setup1	txt = "A"	El mensaje encriptado es "Z"
Atbash	encrypt	setup1	txt = "algoritmos"	El mensaje encriptado es "zotlirgñlh"
Atbash	encrypt	setup1	txt = "¡Programacion!"	El mensaje encriptado es "¡Kiltizñzxrln!"

Objetivo de la Prueba: Verificar que el método decrypt de la clase Cesar funcione correctamente, desencriptando un texto según el track y la direction dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Atbash	decrypt	setup1	txt = "z"	El mensaje desencriptado es "a"
Atbash	decrypt	setup1	txt = "n"	El mensaje desencriptado es "n"
Atbash	decrypt	setup1	txt = "Z"	El mensaje desencriptado es "A"
Atbash	decrypt	setup1	txt = "zotlirgñlh"	El mensaje desencriptado es "algoritmos"
Atbash	decrypt	setup1	txt = "¡Kiltizñzxrln!"	El mensaje desencriptado es "¡Programacion!"

Pruebas Vigenere

Configuración de los Escenarios

Nombre	Clase	Escenario
setup1	VigenereTest	Un objeto de la clase Vigenere con key= "louis"
setup2	VigenereTest	Un objeto de la clase Vigenere con key= "fishclown"
setup3	VigenereTest	Un objeto de la clase Vigenere con key= "redhair"
setup4	VigenereTest	Un objeto de la clase Vigenere con key= "reggaeton"

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método encrypt de la clase Vigenere funcione correctamente, encriptando según la key dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	encrypt	setup1	message = "niño"	El mensaje encriptado es "ywxí"
Vigenere	encrypt	setup2	message = "valentina"	El mensaje encriptado es "aidlpewjn"
Vigenere	encrypt	setup3	message = "alejandro"	El mensaje encriptado es "rphqavuis"
Vigenere	encrypt	setup4	message = "jose luis"	El mensaje encriptado es "asyk lybg"

Objetivo de la Prueba: Verificar que el método decrypt de la clase Vigenere funcione correctamente, desencriptando según la key dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	decrypt	setup4	message = "asyk lybg"	El mensaje encriptado es "jose luis"
Vigenere	decrypt	setup3	message = "rphqavuis"	El mensaje encriptado es "alejandro"
Vigenere	decrypt	setup2	message = "aidlpewjn"	El mensaje encriptado es "valentina"

Objetivo de la Prueba: Verificar que el método recursiveSearchFileEncryptedLetter de la clase Vigenere funcione correctamente, buscando la posición de una letra según la fila de la key dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	recursiveSearchFileEncryptedLetter	setup1	fileKey = 10 currentColumn = 1 currentChar = 'n'	La posición de la letra es 5
Vigenere	recursiveSearchFileEncryptedLetter	setup1	fileKey = 23 currentColumn = 1 currentChar = 'y'	La posición de la letra es 3
Vigenere	recursiveSearchFileEncryptedLetter	setup1	fileKey = 23 currentColumn = 1 currentChar = 'a'	La posición de la letra es 5
Vigenere	recursiveSearchFileEncryptedLetter	setup1	fileKey = 26 currentColumn = 1 currentChar = 'y'	La posición de la letra es 26

Vigenere	recursiveSearchFileEncriptedLetter	setup1	fileKey = 11 currentColumn = 1 currentChar = 'w'	La posición de la letra es 13
----------	------------------------------------	--------	--	-------------------------------

Objetivo de la Prueba: Verificar que el método recursiveSearchPlaneLetter de la clase Vigenere funcione correctamente, buscando la posición de una letra del texto a encriptar en la primera fila de la matriz

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	recursiveSearchPlaneLetter	setup1	currentColumn = 0 currentChar = 'a'	La posición de la letra es 1
Vigenere	recursiveSearchPlaneLetter	setup1	currentColumn = 0 currentChar = 'm'	La posición de la letra es 13
Vigenere	recursiveSearchPlaneLetter	setup1	currentColumn = 0 currentChar = 'l'	La posición de la letra es 12
Vigenere	recursiveSearchPlaneLetter	setup1	currentColumn = 0 currentChar = 'z'	La posición de la letra es 26

Objetivo de la Prueba: Verificar que el método recursiveSearchKeyLetter de la clase Vigenere funcione correctamente, buscando la posición de una letra de la Key en la primera columna de la matriz

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	recursiveSearchKeyLetter	setup1	currentRow = 0 currentChar = 'g'	La posición de la letra es 7
Vigenere	recursiveSearchKeyLetter	setup1	currentRow = 0 currentChar = 'j'	La posición de la letra es 10
Vigenere	recursiveSearchKeyLetter	setup1	currentRow = 0 currentChar = 'p'	La posición de la letra es 16
Vigenere	recursiveSearchKeyLetter	setup1	currentRow = 0 currentChar = 'z'	La posición de la letra es 26

Objetivo de la Prueba: Verificar que el método getVigenereMatrix de la clase Vigenere funcione correctamente, buscando una letra según la posición dada en la matriz

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	getVigenereMatrix	setup1	column = 1 file = 1	La letra en la posición es a
Vigenere	getVigenereMatrix	setup1	column = 7 file = 6	La letra en la posición es l
Vigenere	getVigenereMatrix	setup1	column = 15 file = 17	La letra en la posición es e

Objetivo de la Prueba: Verificar que el método filled de la clase Vigenere funcione correctamente, confirmando que se ha llenado bien la matriz dándole una posición

Clase	Método	Escenario	Valores de Entrada	Resultado
Vigenere	filled	setup1	column = 4 file = 4	La letra en la posición es g
Vigenere	filled	setup1	column = 7 file = 4	La letra en la posición es j
Vigenere	filled	setup1	column = 6 file = 6	La letra en la posición es k

Pruebas AES

Configuración de los Escenarios

Nombre	Clase	Escenario
setup1	AESTest	Un objeto de la clase AES con encryptionKey = "algoritmos"

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método encrypt de la clase AES funcione correctamente, encriptando según la key dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
AES	encrypt	setup1	txt = "holaaa"	El mensaje encriptado es "elixxx"

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método decrypt de la clase AES funcione correctamente, encriptando según la key dada.

Clase	Método	Escenario	Valores de Entrada	Resultado
AES	decrypt	setup1	txt = "Algoritmos y programación dos"	El mensaje encriptado es "xidlofqjlp v moldoxjxzflk alp"

Pruebas RouteList

Configuración de los Escenarios

Nombre	Clase	Escenario
setup1	RouteList	Crea un objeto de la clase RouteList con listas vacías.
setup2	RouteList	Crea un objeto de la clase RouteList con 6 listas. (Route, type, text) "data/encrypt1.txt", true, "hola mundo" "data/encrypt2.txt", true, "chao mundo" "data/encrypt3.txt", true, "punto" "data/decrypt1.txt", false, "hola mundo" "data/decrypt2.txt", false, "chao mundo" "data/decrypt3.txt", false, "punto"

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método add de la clase RouteList funcione correctamente, añadiendo las listas correctamente.				
Clase	Método	Escenario	Valores de Entrada	Resultado
RouteList	add	Setup1	"data/encrypt1.txt", true, "hola mundo" "data/encrypt2.txt", true, "chao mundo" "data/encrypt3.txt", true, "punto" "data/decrypt1.txt", false, "hola mundo" "data/decrypt2.txt", false, "chao mundo" "data/decrypt3.txt", false, "punto"	Se han añadido correctamente

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método search de la clase RouteList funcione correctamente, según las listas creadas en cada caso.				
Clase	Método	Escenario	Valores de Entrada	Resultado
RouteList	search	setUp2	route="data/encrypt1.txt "	El txt = "hola mundo"
RouteList	search	setUp2	route="data/encrypt2.txt "	El txt = "chao mundo"
RouteList	search	setUp2	route="data/encrypt3.txt "	El txt = "punto"
RouteList	search	setUp2	route="data/decrypt1.txt "	El txt = "hola mundo"
RouteList	search	setUp2	route="data/decrypt2.txt"	El txt = "chao mundo"
RouteList	search	setUp2	route="data/decrypt3.txt"	El txt = "punto"
RouteList	search	setUp2	route="data/decrypt4.txt"	El txt = null, ya que no existe esa route

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método delete de la clase RouteList funcione correctamente, eliminando un nodo según el parámetro.

Clase	Método	Escenario	Valores de Entrada	Resultado
RouteList	delete	Setup2	Route= "data/encrypt1.txt"	Se ha eliminado correctamente
RouteList	delete	Setup2	Route= "data/encrypt3.txt"	Se ha eliminado correctamente
RouteList	delete	Setup2	Route= "data/decrypt3.txt"	Se ha eliminado correctamente