

REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES LAB 3 MAQUINA DE TURING**Requerimientos funcionales:**

1. Leer el carácter de una cabeza automática específica. Primero, se identifica la cabeza automática que leerá el carácter, con la lectura de un carácter que tendrá valores entre 0 y 2 (0 para C0, 1 para C1, y 2 para C2). Seguido de este carácter habrá un carácter con valor 0 (que indica la operación de lectura de carácter). Si la cabeza automática no se encuentra sobre ningún elemento, se retorna '#'. Si no, se retorna el carácter del elemento.
2. Agregar un elemento a la cinta de la máquina. Primero se identifica la cabeza automática que agregará el elemento, con la lectura de un carácter que tendrá valores entre 0 y 2 (0 para C0, 1 para C1, y 2 para C2). Seguido de este carácter habrá un carácter con valor 1 (que indica la operación de agregación de elemento). Por último, enseguida del anterior carácter estará el carácter a agregar (Una letra del alfabeto inglés, en mayúscula). La posición en la que será agregado el elemento depende de la cabeza seleccionada: C0 agrega en la primera posición, C1 agrega en la mitad y C2 agrega al final.
3. Remover un elemento de la cinta de la máquina. Primero se identifica la cabeza automática que removerá el elemento, con la lectura de un carácter que tendrá valores entre 0 y 2 (0 para C0, 1 para C1, y 2 para C2). Seguido de este carácter habrá un carácter con valor 2 (que indica la operación de remoción de elemento). Si se trata de remover un elemento cuando la cinta está vacía (No contiene ningún elemento), no ocurrirá nada.

Requerimientos no funcionales:

1. El input y output del programa deben darse por medio de archivos de texto.
2. Cada línea de output debe contener SOLAMENTE un carácter por línea (después de cada lectura de cabeza automática, se baja una línea).
3. Cada línea de input es una máquina de Turing diferente, por lo que, al pasar de una línea a otra, todos los elementos se eliminan.
4. Cada línea de input tiene llamados a los requerimientos funcionales, sin ningún tipo de separador entre ellos.
5. Cada línea de input debe ser válida (no se hacen llamados a cabezas ni operaciones inexistentes, ni se trata de agregar un carácter diferente del alfabeto inglés en mayúsculas).
6. El tiempo de operación debe ser calculado e impreso en consola.
7. Las cabezas deben de actualizar sus posiciones automáticamente después de cada operación de adición y remoción.

DISEÑO CASOS DE PRUEBA LAB 3 MAQUINA DE TURING**Configuración de los Escenarios**

Nombre	Clase	Escenario
setup1	Turing	Una máquina de Turing, con 8 elementos agregados: 1er elemento: A 2do elemento: U 3er elemento: R 4to elemento: O 5to elemento: I 6to elemento: D 7to elemento: L 8vo elemento: E C0 = firstCell = 1er elemento C1 = 4to elemento C2 = 8vo elemento
setup2	Turing	Una máquina de Turing, con 9 elementos agregados: 1er elemento: A 2do elemento: L 3er elemento: G 4to elemento: O 5to elemento: R 6to elemento: I 7to elemento: T 8vo elemento: H 9no elemento: M C0 = firstCell = 1er elemento

		C1 = 5to elemento C2 = 9no elemento
--	--	--

Diseño de Casos de Prueba

Objetivo de la Prueba: Comprobar el correcto funcionamiento del método addCell(letter:char,head:char):void, realizando el método bajo distintos parámetros: En una máquina sin elementos, en una maquina con un elemento, en una máquina con elementos pares y una máquina con elementos impares. Se revisarán las cabezas autómatas después de la realización del método a fin de observar si se han reubicado correctamente.				
Clase	Método	Escenario	Valores de Entrada	Resultado
TuringTest	addCell1	ninguno	letter = 'U' head = 0 letter = 'M' head = 1 letter = 'L' head = 2	True (se agregará a c0 sin elementos) True (se agregará a c1 sin elementos) True (se agregará a c2 sin elementos) En el primero, todas las cabezas apuntan al mismo elemento con la letra 'U' En el segundo, todas las cabezas apuntan al mismo elemento con la letra 'M' En el tercero, todas las cabezas apuntan al mismo elemento con la letra 'L'
TuringTest	addCellTest2	ninguno	letter = 'P' head = 0 letter = 'D' head = 1 letter = 'F' head = 2	True (se agregará a c0 con 1 elemento) True (se agregará a c1 con 1 elemento) True (se agregará a c2 con 1 elemento) En el primero, c0 y c1 apuntan al elemento con la letra 'P' y c2 al elemento con la letra 'C' En el segundo, c0 y c1 apuntan al elemento con la letra 'D' y c2 al elemento con la letra 'S'

				En el tercero, c0 y c1 apuntan al elemento con la letra 'W' y c2 al elemento con la letra 'F'
TuringTest	addCellTest3	setup1()	letter = 'S' head = 0 letter = 'A' head = 1 letter = 'D' head = 2	True (se agregará a c0 con 8 elementos) True (se agregará a c1 con 8 elementos) True (se agregará a c2 con 8 elementos) En el primero, c0 apunta a 'S', c1 apunta a 'O' y c2 apunta a 'E' En el segundo, c0 apunta a 'A', c1 apunta a 'A' y C2 apunta a 'E' En el tercero, c0 apunta a 'A', c1 apunta a 'I' y C2 apunta a 'D'
TuringTest	addCellTest4	setup2()	letter = 'N' head = 0 letter = 'B' head = 1 letter = 'A' head = 2	True (se agregará a c0 con 9 elementos) True (se agregará a c1 con 9 elementos) True (se agregará a c2 con 9 elementos) En el primero, c0 apunta a 'N', c1 apunta a 'O' y c2 apunta a 'M' En el segundo, c0 apunta a 'A', c1 apunta a 'B' y C2 apunta a 'M' En el tercero, c0 apunta a 'A', c1 apunta a 'R' y C2 apunta a 'A'

Objetivo de la Prueba: Comprobar el correcto funcionamiento del método `readCell(head:char):char`, realizando el método bajo distintos parámetros: En una máquina sin elementos, en una máquina con un elemento, en una máquina con elementos pares y una máquina con elementos impares. Se usará el nombre `letter` para expresar el return resultante del método.

Clase	Método	Escenario	Valores de Entrada	Resultado
TuringTest	readCellTest1	ninguno	head = 0 head = 1 head = 2	True (se leerá c0 sin elementos) True (se leerá c1 sin elementos) True (se leerá c2 sin elementos) letter debería ser igual a '#' en todos
TuringTest	readCellTest2	ninguno	head = 0 head = 1 head = 2	True (se leerá c0 con un elemento) True (se leerá c1 con un elemento) True (se leerá c2 con un elemento) letter debería ser igual a 'F' en todos
TuringTest	readCellTest3	setup1()	head = 0 head = 1 head = 2	True (se leerá c0 con 8 elementos) True (se leerá c1 con 8 elementos) True (se leerá c2 con 8 elementos) letter en c0 debería ser igual a 'A' letter en C1 debería ser igual a 'O' letter en C2 debería ser igual a 'E'
TuringTest	readCellTest4	setup2()	head = 0 head = 1 head = 2	True (se leerá c0 con 9 elementos) True (se leerá c1 con 9 elementos) True (se leerá c2 con 9 elementos) letter en c0 debería ser igual a 'A' letter en C1 debería ser igual a 'R' letter en C2 debería ser igual a 'M'

Objetivo de la Prueba: Comprobar el correcto funcionamiento del método `removeCell(head:char):void`, realizando el método bajo distintos parámetros: En una máquina sin elementos, en una máquina con un elemento, en una máquina con dos elementos, en una máquina con elementos pares y una máquina con elementos impares. Se revisarán las cabezas autómatas después de la realización del método a fin de observar si se han reubicado correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
TuringTest	removeCellTest1	ninguno	head = 0 head = 1 head = 2	True (se removerá c0 sin elementos) True (se removerá c1 sin elementos) True (se removerá c2 sin elementos) Todas las cabezas deberían apuntar a null
TuringTest	removeCellTest2	ninguno	head = 0 head = 1 head = 2	True (se removerá c0 con 1 elemento) True (se removerá c1 con 1 elemento) True (se removerá c2 con 1 elemento) Todas las cabezas deberían apuntar a null
TuringTest	removeCellTest3	ninguno	head = 0 head = 1 head = 2	True (se removerá c0 con 2 elementos) True (se removerá c1 con 2 elementos) True (se removerá c2 con 2 elementos) En el primero, c0, c1 y c2 apuntan a 'R' En el segundo, c0, c1 y c2 apuntan a 'R' En el tercero, c0, c1 y c2 apuntan a 'Q'

TuringTest	removeCellTest4	setup1()	<p>head = 0</p> <p>head = 1</p> <p>head = 2</p>	<p>True (se removerá c0 con 8 elementos)</p> <p>True (se removerá c1 con 8 elementos)</p> <p>True (se removerá c2 con 8 elementos)</p> <p>En el primero, c0 apunta a 'U', c1 apunta a 'I' y c2 apunta a 'E'</p> <p>En el segundo, c0 apunta a 'A', c1 apunta a 'I' y c2 apunta a 'E'</p> <p>En el tercero, c0 apunta a 'A', c1 apunta a 'O' y c2 apunta a 'L'</p>
TuringTest	RemoveCellTest5	setup2()	<p>head = 0</p> <p>head = 1</p> <p>head = 2</p>	<p>True (se removerá c0 con 9 elementos)</p> <p>True (se removerá c1 con 9 elementos)</p> <p>True (se removerá c2 con 9 elementos)</p> <p>En el primero, c0 apunta a 'L', c1 apunta a 'R' y c2 apunta a 'M'</p> <p>En el segundo, c0 apunta a 'A', c1 apunta a 'O' y c2 apunta a 'M'</p> <p>En el tercero, c0 apunta a 'A', c1 apunta a 'O' y c2 apunta a 'H'</p>