



Problema resuelto usando pilas en Java

Una empresa quiere que su página web tenga integrado un sistema que simule el comportamiento del botón “Atrás” de los navegadores, que básicamente permite al usuario regresar a la página anterior.

Para ello, se utiliza una pila que guarda las páginas visitadas en orden. Cada vez que el usuario visita una nueva página, esta se guarda en la pila. Cuando escriba “Atrás”, se elimina la página actual y se muestra la anterior (la que está ahora en el tope de la pila).

Historias de usuario

Historia de usuario	1º	Título: Visitar páginas.
Descripción	Como:	Visitante de la página web.
	Quiero:	Agregar páginas que voy visitando en la página en un historial.
	Para:	Poder retroceder en caso de que así lo desee.
Criterios de aceptación	<ul style="list-style-type: none">● Cada URL ingresada representa una página nueva visitada y debe agregarse de la forma “www.{nombre_pagina}.com” (nombre_pagina es el nombre de la página sin espacios).● Las páginas se agregan en orden a una pila.● Al ingresar una página nueva, se debe mostrar en la consola “Página actual: {nombre_pagina}”, siendo nombre_pagina el mismo del de la URL.	

Historia de usuario	2º	Título: Retroceder en el historial.
Descripción	Como:	Visitante de la página web.

	Quiero:	Poder retroceder a través de mi historial de páginas.
	Para:	Entrar a páginas nuevas a través de una página previa.
Criterios de aceptación	<ul style="list-style-type: none"> Al escribir la palabra “ATRAS” el programa debe eliminar la última página registrada en el historial y presentar la anterior. En el caso de que la pila esté vacía, el programa debe mandar el mensaje “Sin historial disponible”. 	

Historia de usuario	3º	Título: Salir de la página.
Descripción	Como:	Visitante de la página web.
	Quiero:	Cerrar la página.
	Para:	Poder salir en cualquier punto.
Criterios de aceptación	<ul style="list-style-type: none"> Si el visitante escribe “SALIR”, el programa se debe cerrar sin mostrar ningún mensaje. 	

Formato de la entrada y salida:

Entrada	Salida
Al ingresar una página estará escrita de la forma “ www.{nombre pagina}.com ” como fue mencionado en los criterios de aceptación. Si se escribe “ATRAS”, debe eliminarse la última página visitada de la lista y la actual será la anterior ingresada a esa. En caso de escribir “SALIR” el programa debe finalizar.	Si el usuario ingresa una URL debe mostrar en consola “Pagina actual: nombre_pagina”. En caso de devolverse, debe mostrar el mismo mensaje de “Pagina actual: nombre_pagina”, pero con la página anterior a la eliminada.

Instrucciones para la calificación automática

- El nombre de la clase debe ser: Main
- El método principal debe ser public static void main(String[] args)
- Usar Scanner para leer las entradas desde consola.
- No se debe imprimir texto adicional fuera de lo indicado.

Ejemplos de entrada y de salida esperados.

Entrada	Salida esperada
www.google.com www.youtube.com ATRAS SALIR	Pagina actual: google Pagina actual: youtube Pagina actual: google
www.facebook.com ATRAS ATRAS SALIR	Pagina actual: facebook Sin historial disponible Sin historial disponible

Código en Java

```
import java.util.EmptyStackException;
import java.util.Scanner;
import java.util.Stack;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Stack<String> historial = new Stack<>();

        while (true) {
            String entrada = scanner.nextLine();

            if (entrada.equals("SALIR")) {
                break;
            }

            if (entrada.equals("ATRAS")) {
                if (!historial.isEmpty()) {
                    historial.pop();
                }
            } else {
                String nombre = entrada.substring(4, entrada.length() - 4);
                historial.push(nombre);
            }
        }
    }
}
```

```

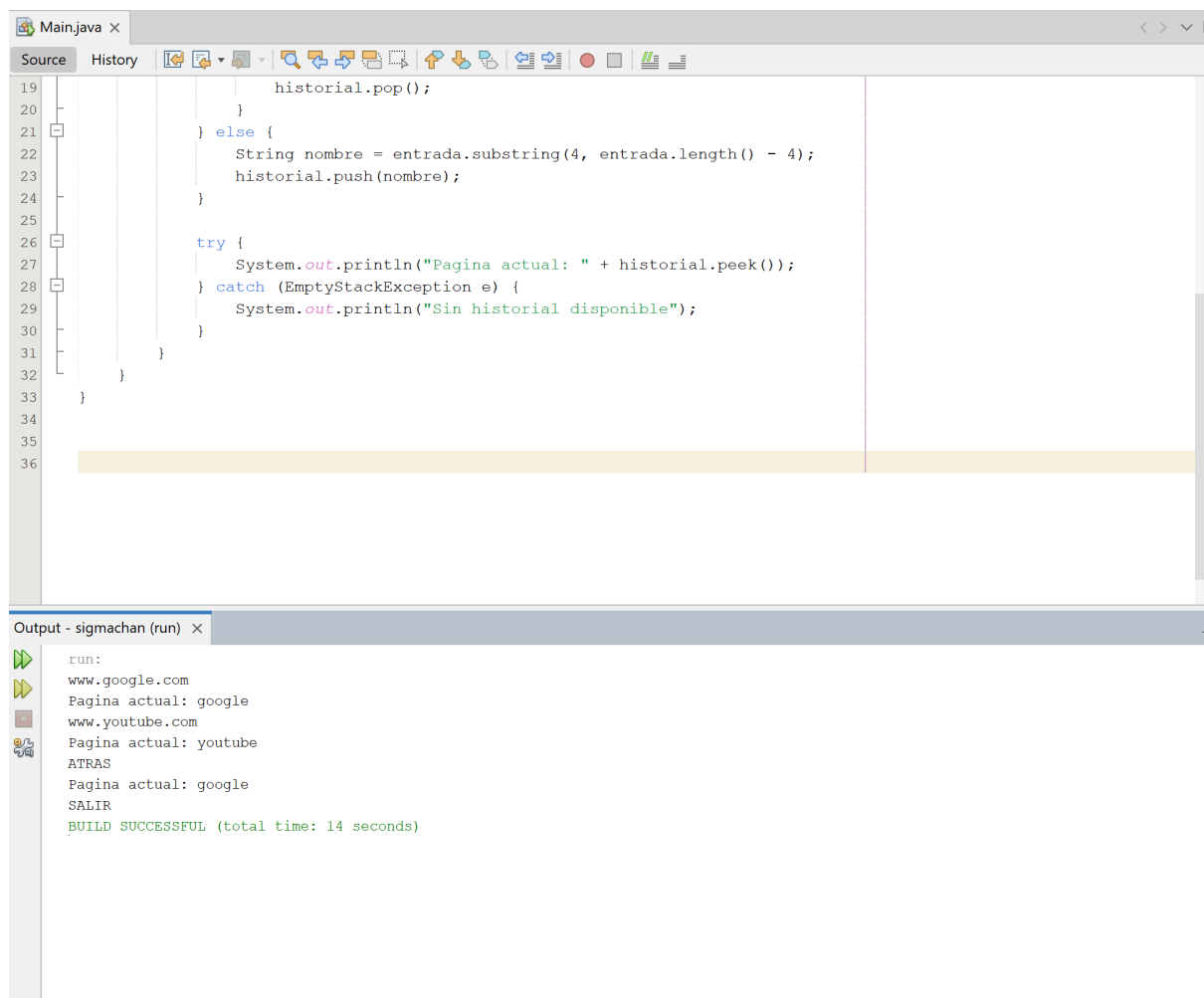
    }

    try {
        System.out.println("Pagina actual: " + historial.peek());
    } catch (EmptyStackException e) {
        System.out.println("Sin historial disponible");
    }
}
}
}
}

```

Evidencia de ejecución

Al ejecutar el código y usar el primer caso de ejemplo, se muestra lo siguiente



The screenshot shows an IDE window titled 'Main.java'. The code is as follows:

```

19         historial.pop();
20     }
21     } else {
22         String nombre = entrada.substring(4, entrada.length() - 4);
23         historial.push(nombre);
24     }
25
26     try {
27         System.out.println("Pagina actual: " + historial.peek());
28     } catch (EmptyStackException e) {
29         System.out.println("Sin historial disponible");
30     }
31 }
32 }
33 }
34 }
35
36

```

Below the code editor is the 'Output - sigmachan (run)' window, which displays the following execution results:

```

run:
www.google.com
Pagina actual: google
www.youtube.com
Pagina actual: youtube
ATRAS
Pagina actual: google
SALIR
BUILD SUCCESSFUL (total time: 14 seconds)

```