|  |  |
| --- | --- |
|  | **INGENIERIA EN SISTEMAS DE INFORMACIÓN**  K2054  AÑO 2023 |
| ***Sintaxis y Semántica de los Lenguajes*** | |
| TP 1: Expresiones Regulares y ExpresionesRegulares Extendidas en Bash | |
| **DOCENTE TITULAR:** Ing. Pablo D. Mendez    **ALUMNOS:**  Manuel García Frigo  Legajo: 208.007-2  Mail Institucional: mgarciafrigo@frba.utn.edu.ar  Github: ManuelGarciaF  Matías Germán Rainhart  Legajo: 209.154-9  Mail Institucional: mrainhart@frba.utn.edu.ar  Github: matias-rainhart  **Link al Repositorio:** https://github.com/ManuelGarciaF/ssl-grupo-42 | |

**Punto 3**

**Variables** se las utiliza para almacenar valores. Para asignar valores se hace con el formato *NOMBRE = valor*. Para acceder a valores se usa el formato *$NOMBRE*.

Ejemplo:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

**Sentencias condicionales** permite tomar decisiones en función de condiciones específicas.

La estructura básica de una sentencia if en Bash es:

if [condición]; then

# Código a ejecutar si la condición es verdadera

fi

Dentro de la condición se utilizan distintos operadores de comparación, como, por ejemplo -lt (menor estricto), -ge (mayor o igual) ó -eq (igualdad).

Ejemplo:

Texto

Descripción generada automáticamente con confianza media

**Sentencias cíclicas** en Bash permiten repetir una serie de comandos varias veces. Los dos tipos comunes de ciclo son for y while.

El ciclo for se utiliza para iterar sobre una lista de elementos y realizar una serie de comandos para cada elemento de la lista. Su estructura es:

for elemento in elemento1 elemento2 elemento3; do

# Código a ejecutar para cada elemento

done

El ciclo while se utiliza para ejecutar un conjunto de comandos repetidamente mientras una condición se mantenga verdadera. Su estructura es:

while [condición]; do

# Código a ejecutar mientras la condición sea verdadera

done

Ejemplos de ambos ciclos:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Funciones o Subprogramas:** sirven para reutilizar un bloque de código. Para definir una función la estructura es:

nombre\_funcion() {

# Código de la función

}

Para llamar una función se debe escribir su nombre, seguido de sus argumentos, separados con un espacio

Ejemplo:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

**Casos de Prueba:**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

ACLARACIÓN: La salida de la ejecución también está disponible en salida.txt