**Trabajo práctico nro. 12**

|  |  |  |
| --- | --- | --- |
|  | **Asignatura:** | |
|  | |
| **Cursado:** | **Horas** **semanales**: |
|  | **Horas semestrales:** |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

* **Tema:**

**Expresiones Regulares**

* **Enunciados:**

**Empezaremos por unos ejercicios básicos de programas Java con estructura secuencial, es decir, en estos programas no hay instrucciones condicionales ni repetitivas. En la mayoría de ellos las operaciones a realizar son: lectura de datos por teclado, realizar alguna operación con esos datos y mostrar resultados por pantalla.**

**Recomendaciones: Leer la teoría, investigar, consultar si existen dudas y realizar el trabajo en clases. La entrega de este trabajo practico es individual.**

* **Investigación de Expresiones Regulares**:
* Investiga y comprende los conceptos básicos de las expresiones regulares, incluyendo caracteres especiales, cuantificadores y grupos.
* Explora ejemplos de patrones comunes de expresiones regulares utilizados en la validación de datos.
* **Creación de un Conjunto de Datos**:
* Crea un conjunto de datos ficticio que contenga información variada, como direcciones de correo electrónico, números de teléfono, fechas, códigos postales, etc.
* **Validación de Datos**:
* Diseña un programa en un lenguaje de programación (Java) que utilice expresiones regulares para validar y verificar los datos en tu conjunto de datos. Por ejemplo, puedes verificar si una dirección de correo electrónico es válida o si un número de teléfono sigue un formato específico.
* Validar entrada de datos en un arraylist y aplicarlo.
* **Informe de Investigación**:
* Prepara un informe que desarrolle el tema sobre las expresiones regulares y cómo se aplicaron en la validación de datos.
* Incluye ejemplos de patrones de expresiones regulares utilizados en tu programa y cómo ayudaron en la validación de los datos.
* Comprensión de los conceptos de expresiones regulares.
* Diseño y desarrollo del programa de validación de datos.
* Claridad y organización del informe de investigación.
* **Ejercicios**:
* Comprobar si el String *cadena* no acaba con un dígito.

String regex1 = ".\*\\D$";

* Comprobar si el String *cadena* no empieza por un dígito.

String regex2 = "^\\D.\*";

* Comprobar si el String *cadena* está formado por un mínimo de 5 letras mayúsculas o minúsculas y un máximo de 10.

String regex3 = "^[a-zA-Z]{5,10}$";

* Validar el ingreso de un DNI (xx-xxx-xxx).

String regex4 = "^\\d{2}-\\d{3}-\\d{3}$";

* Guardar en un arraylist una serie de mails, pero antes de ser guardados deben cumplir con las condiciones de las expresiones regulares.

ArrayList<String> listaCorreos = new ArrayList<>();

String emailRegex = "^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$";

// Asumiendo que tienes un array de correos electrónicos llamado emails

for (String email : emails) {

if (email.matches(emailRegex)) {

listaCorreos.add(email);

}

}

* Comprobar si el String *cadena* contiene “abc”.

String regex6 = ".\*abc.\*";