Taller 01: Teoría de la computación

PALABRAS

1. **Elaborar un programa que anexe, actualice y borre datos de una cadena de caracteres.**

```
import java.util.ArrayList;
import java.util.List;

public class EjemploListaNombres {
        private List<String> listaNombres;

        public EjemploListaNombres() {
                this.listaNombres = new ArrayList<>( );
        }

        public void ingresarNombres() {
                System.out.println("\nANEXAR DATOS");
                listaNombres.add("Leidy Gómez");
                listaNombres.add("Edwin Zuluaga");
                listaNombres.add("Julio Casanova");
                listaNombres.add("Daniel Narváez");
                listaNombres.add("Carolina Muelle");
        }

        public void imprimirlistaNombres() {
                for(String lst: listaNombres) {
                        System.out.println(lst);
                }
        }

        public void borrarElemento() {
                System.out.println("\nBORRAR DATOS");
                listaNombres.remove(3);
        }

        public void actualizarDato() {
                System.out.println("\nACTUALIZAR DATOS");
                listaNombres.set(2,"Fermín León");
        }
}
```

%%%

```
public class AppEjemploListaNombres {
        private EjemploListaNombres lista;

        public AppEjemploListaNombres() {
                this.lista = new EjemploListaNombres();
        }


        public void procesarLista() {
                lista.ingresarNombres();
                lista.imprimirlistaNombres();
                lista.actualizarDato();
                lista.imprimirlistaNombres();
                lista.borrarElemento();
                lista.imprimirlistaNombres();
        }

        public static void main(String[] args) {
                AppEjemploListaNombres app = new AppEjemploListaNombres();
                app.procesarLista();
        }
}
```

2. **A partir del alfabeto latino Σ = {a, b, c….z}. Generar palabras conformadas por tres letras tomadas de forma aleatoria del alfabeto.**

```
import java.util.Random;

public class Palabras {
    private char []alfabeto;

    public Palabras() {
        this.alfabeto = new char[ ]{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p',
                                    'q','r','s','t','u','v','w','x','y','z'};
    }

    public void formarPalabra( ) {
        Random rdm = new Random( );
        String palabra = "";
        int n = 0;
        for (int i = 1; i <= 3 ; i++) {
            n = rdm.nextInt(alfabeto.length);
            palabra+=" "+alfabeto[n];
        }
        System.out.println(""+palabra);
    }
}
```

%%

```
public class TestPalabra
{
        public static void main(String[ ] args) {
        Palabras app = new Palabras( );
        app.formarPalabra( );
    }
}
```

3. **Construir un programa que permita generar palabras que tengan por lo menos dos a.**

4. **Construir una palabra conformada por el alfabeto binario Σ= {0,1} de longitud aleatoria entre 1 y 5 dígitos y calcular su equivalencia en números decimales.**

5. **Teniendo en cuenta el alfabeto latino conformado por las letras {A…Z} (mayúsculas), {a…z} (minúsculas), los dígitos del {0…9}, construir de forma aleatoria una clave de un tamaño de 7 caracteres conformada por:**

    a. **El primer carácter en mayúscula.**
    b. **caracteres en minúscula.**
    c. **1 digito [0-9].**