



ESTUDIANTE: HENRY JAVIER HUARACHI QUISPE

DOCENTE: LIC. WILLIAM BARRA

PRACTICA ESTRUCTURA DE DATOS

MANEJO DE CONCEPTOS

1. ¿A que se refiere cuando se habla de ESTRUCTURA DE DATOS?

- En el ámbito de la informática, las estructuras de datos son aquellas que nos permiten, como desarrolladores, organizar la información de manera eficiente, y en definitiva diseñar la solución correcta para un determinado problema.

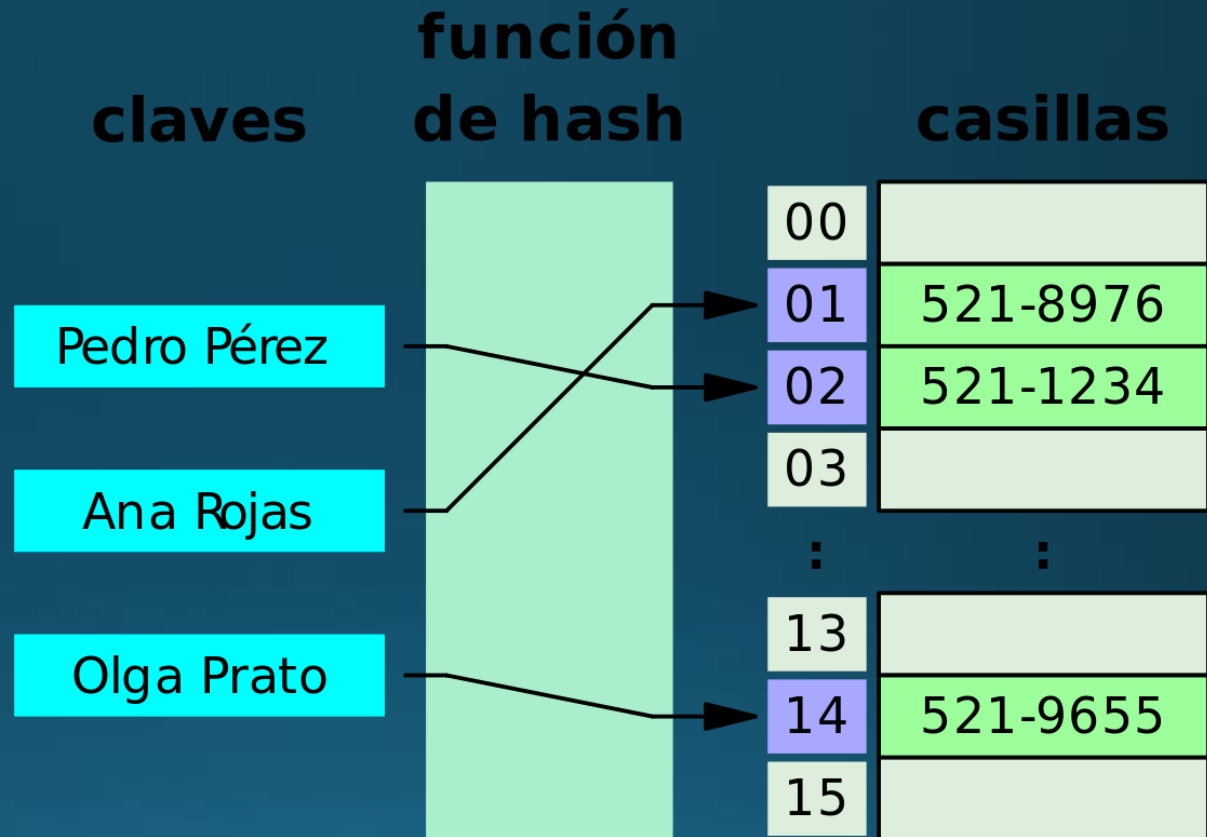
2. ¿Cuáles son los TIPOS DE ESTRUCTURA QUE EXISTE?

- Los tipos de estructura de datos son dos los cuales son:
 - Estructuras lineales
 - Estructuras no lineales

3. ¿Apoyándose en el link adjunto, explique, por qué son útiles las estructuras de datos?.

Nos son utiles porque nos brindan una bateria de herramientas.

Que nos ayudan a optimizar recursos mas que todo para internet de las cosas como Tambien para los que trabajan con bigdata



4. ¿Qué es una PILA?

es un tipo especial de **lista lineal** dentro de las **estructuras de datos dinámicas** que permite almacenar y recuperar datos.

PilaCliente		
m	PilaCliente (int)	
f	tope	int
f	max	int
f	pilita	Cliente []
m	mostrar()	void
m	insertar(Cliente)	void
m	eliminar()	Cliente
m	estaLleno()	boolean
m	numeroElementos()	int
m	estaVacia()	boolean
m	vaciar(PilaCliente)	void

5. ¿Qué es STACK en JAVA, una STACK será lo mismo que una PILA?











Es un stack (pila) es un objeto similar a una pila.

Si serian lo mismo.

PilaCliente		
m	PilaCliente (int)	
f	tope	int
f	max	int
f	pilita	Cliente []
m	mostrar()	void
m	insertar(Cliente)	void
m	eliminar()	Cliente
m	estaLleno()	boolean
m	numeroElementos()	int
m	estaVacia()	boolean
m	vaciar(PilaCliente)	void











6. ¿Qué es TOPE en una PILA?

El tope es un solo extremo de la pila donde se pueden poner elementos nuevos Como Tambien quitarlos el ultimo elemento será el primero en quitarse.

		PilaCliente	
		PilaCliente (int)	
		tope	int
		max	int
		pilita	Cliente []











7. ¿Qué es MAX en una PILA?

Es una variable auxiliar que se denomina como tope.

		PilaCliente	
		PilaCliente(int)	
		tope	int
		max	int
		pilita	Cliente []

8. ¿A que se refiere los métodos esVacia() y esLlena() en una PILA?

Devuelve verdadero si la pila está vacía, falso en caso contrario y funciona de la misma manera si la pila esta llena.

		PilaCliente	
		PilaCliente(int)	
		tope	int
		max	int
		pilita	Cliente []



```

public static void asignarDireccion(PilaCliente pila, String nuevaDireccion){
    Cliente elem = null;
    if(pila.estaVacia()){
        System.out.println("La pila esta vacia");
    } else {
        while(!pila.estaVacia()){
            elem = pila.eliminar();
            if(elem.getGenero().equals("Femenino")){
                elem.setDireccion(nuevaDireccion);
                elem.mostrarDatos();
            }
        }
    }
    pila.mostrar();
}

```

9. ¿Qué son los métodos estáticos en JAVA?

Un metodo estatico es un metodo que no crea ningun objeto previamente.

 `asignarDireccion (PilaCliente , String) void`

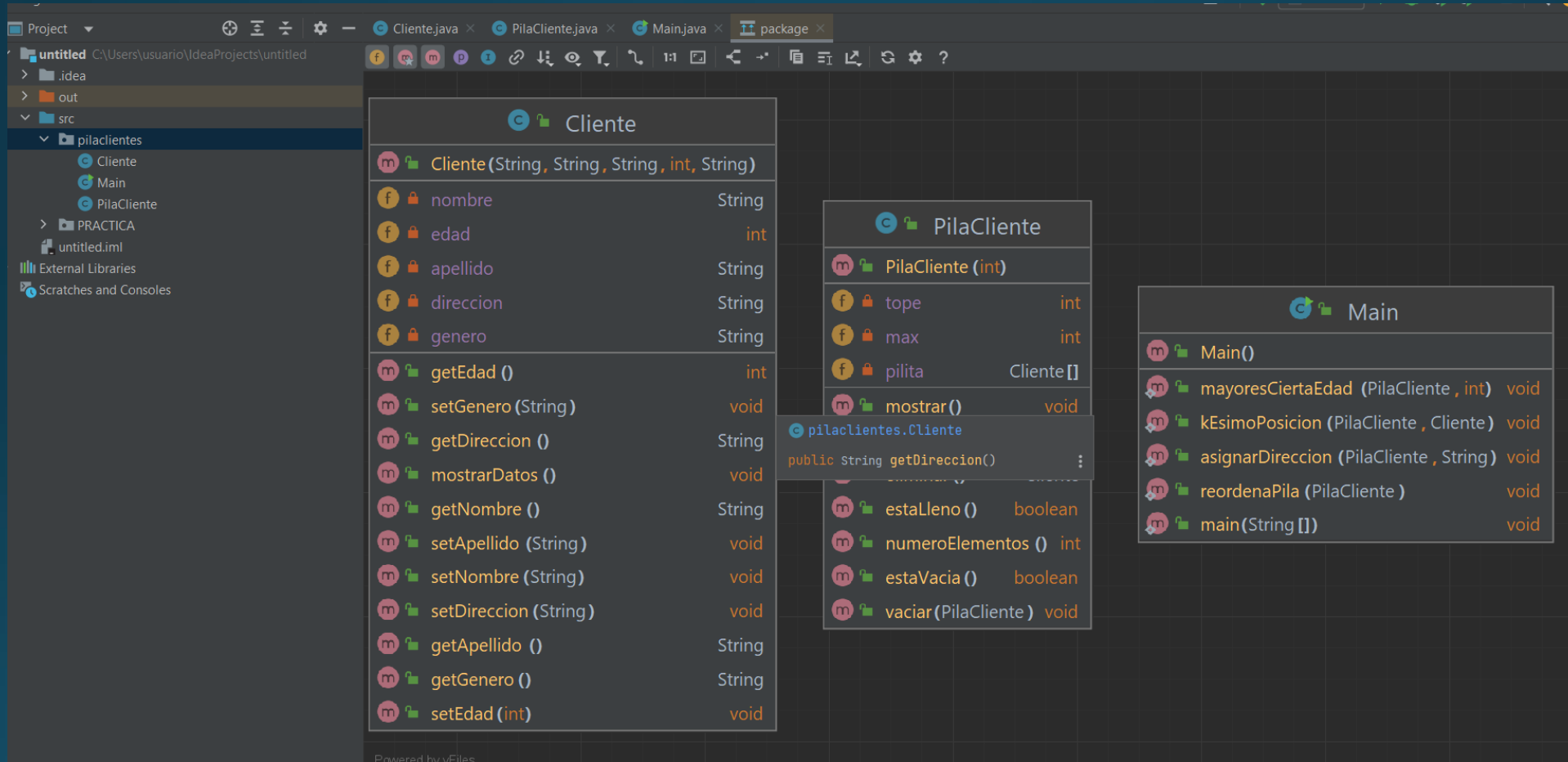
10. ¿A través de un gráfico, muestre los métodos mínimos que debería de tener una PILA?

Tendríamos

PilaDeLibros		
m	PilaDeLibros (int)	
f	max	int
f	tope	int
f	libro	libro []
m	esVacio ()	boolean
m	adicionar (libro)	void
m	vaciar (PilaDeLibros)	void
m	eliminar ()	libro
m	nroElementos ()	int
m	mostrar ()	void
m	esllena ()	boolean
m	llenar ()	void

Parte practica

11. Crear las clases necesarias para la PILA DE CLIENTES.



12.Determinar cuántos CLIENTES son mayores de 20 años.

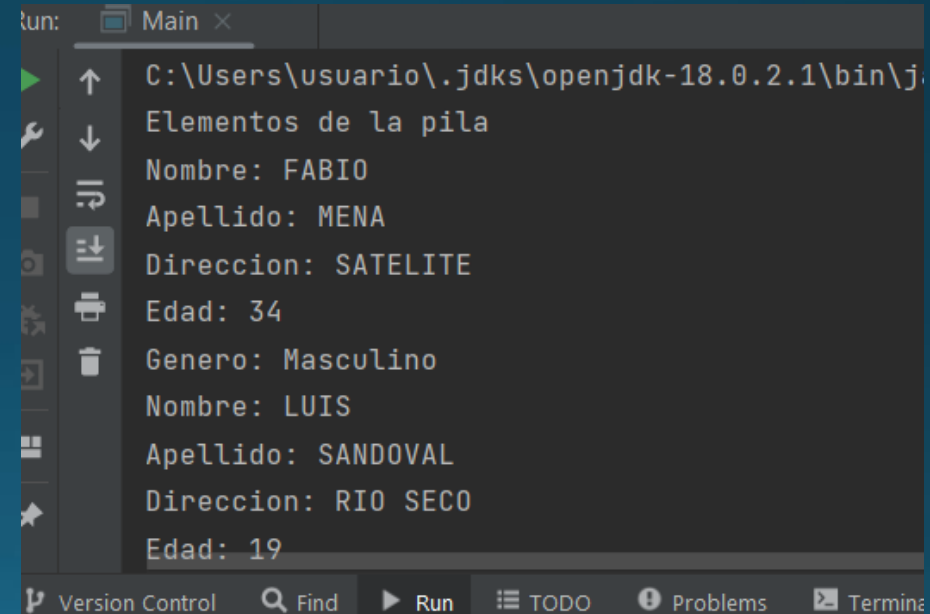
```
//metodos
//12.Determinar cuántos CLIENTES son mayores de 20 años.
// El método deberá llamarse mayoresCiertaEdad(Pila, edadMayor)
// El método debe ser creado en la clase MAIN como un método estático.
// El método recibe 2 parámetros
// -La Pila de Clientes
// -El valor de la edad.
1 usage
public static void mayoresCiertaEdad(PilaCliente pila, int edadMayor){
    int contador = 0;
    Cliente elem = null;
    if(pila.estaVacia()){
        System.out.println("La pila esta vacia");
    } else {
        while(!pila.estaVacia()){
            elem = pila.eliminar();
            if(elem.getEdad() > edadMayor){
                contador++;
            }
        }
    }
    System.out.println("La cantidad de clientes con mas de " + edadMayor + " son: " + contador);
}
```

```
C:\Users\usuario\.jdk\openjdk-18.0.2.1\bin\java.exe "
La cantidad de clientes con mas de 20 son: 3
```

```
Process finished with exit code 0
```

13. Mover el k-ésimo elemento al final de la pila.

```
//13.Mover el k-ésimo elemento al final de la pila.  
// El método deberá llamarse moverElemento(Pila, valorTope)  
// El método debe ser creado en la clase MAIN como un método estático.  
// El método recibe 2 parámetros  
// -La Pila de Clientes  
// -El valor de valorTope  
  
public static void kEsimoPosicion(PilaCliente pila, Cliente valorTope){  
    PilaCliente aux= new PilaCliente( max: 10);  
    Cliente elem = null;  
    while(!pila.estaVacia()){  
        elem = pila.eliminar();  
        if(elem != valorTope){  
            aux.insertar(elem);  
        }  
    }  
    pila.vaciar(aux);  
    pila.insertar(valorTope);  
    pila.mostrar();  
}
```



```
run: Main x  
C:\Users\usuario\.jdk\openjdk-18.0.2.1\bin\j.  
Elementos de la pila  
Nombre: FABIO  
Apellido: MENA  
Direccion: SATELITE  
Edad: 34  
Genero: Masculino  
Nombre: LUIS  
Apellido: SANDOVAL  
Direccion: RIO SECO  
Edad: 19  
Version Control Find Run TODO Problems Termina
```

14. Cambiar la dirección de algunos CLIENTES de la PILA

```
//14. Cambiar la dirección de algunos CLIENTES de la PILA.
// El método deberá llamarse asignarDireccion(Pila, nuevaDireccion)
// El método debe ser creado en la clase MAIN como un método estático.
// El método recibe 2 parámetros
// -La Pila de Clientes
// -La nueva dirección
//Cambiar la dirección del cliente siempre y cuando el género sea FEMENINO.

1 usage
public static void asignarDireccion(PilaCliente pila, String nuevaDireccion){
    Cliente elem = null;
    if(pila.estaVacia()){
        System.out.println("La pila esta vacia");
    } else {
        while(!pila.estaVacia()){
            elem = pila.eliminar();
            if(elem.getGenero().equals("Femenino")){
                elem.setDireccion(nuevaDireccion);
                elem.mostrarDatos();
            }
        }
    }
    pila.mostrar();
}
```

```
Main X
Nombre: FANELA
Apellido: PLATA
Direccion: villa dolores
Edad: 45
Genero: Femenino
Nombre: SECILIA
Apellido: CONDORI
Direccion: villa dolores
Edad: 23
Genero: Femenino
La pila esta vacia
```

15.Mover ÍTEMS de la PILA.

```
//15.Mover ÍTEMS de la PILA.  
//El método deberá llamarse reordenaPila(Pila)  
//El método debe ser creado en la clase MAIN como un método estático.  
//El método recibe 1 parámetro  
// -La Pila de Clientes  
//El método debe reordenar la pila de tal manera que los clientes de género FEMENINO queden al final de la pila.  
  
public static void reordenaPila(PilaCliente pila){  
    PilaCliente aux= new PilaCliente( max: 10);  
    Cliente valorExtraidoPila=null;  
    while(!pila.estaVacia()){  
        valorExtraidoPila = pila.eliminar();  
        if(valorExtraidoPila.getGenero().equals("Femenino")){  
            valorExtraidoPila.mostrarDatos();}  
        else{  
            aux.insertar(valorExtraidoPila);  
        }  
    }  
    pila.vaciar(aux);  
    pila.mostrar();  
}
```

Run: Main x

Nombre: PAMELA
Apellido: PLATA
Direccion: SANTA ROSA
Edad: 45
Genero: Femenino
Nombre: SECILIA
Apellido: CONDORI
Direccion: 12 DE OCTUBRE
Edad: 23
Genero: Femenino
Elementos de la pila

Structure

Version Control Find Run TODO

Build completed successfully in 2 sec, 17 ms (moment)

Escribe aquí para buscar