

# Directorio en java /Repaso a POO

## Funcionamiento en general .

El código necesita para funcionar las librería java.util.Arrays.Ya que esté programa maneja el directorio mediante arreglos.

Al ejecutar el programa lo primero que nos pide es el tamaño de la agenda que vamos a manejar, luego nos muestra las seis opciones para el usuario

### 1. Agregar una Persona

1.1Alumno

1.2 Coordinador

1.3 Profesor

### 2. Editar una persona

1.1Alumno

1.2 Coordinador

1.3 Profesor

### 3. Borrar una persona

3.1 Por nombre

3.2 Por correo

### 4. Buscar una persona

4.1 Búsqueda por nombre

4.2 Búsqueda por nombre y rol

4.3 Búsqueda por correo

### 5. ver la agenda

### 6. Salir

## Clases :

Persona Esta el **clase padre de Alumno,Director,Coordinadores** ,es decir en esta clase se declaran las **características/parámetros** que tienen en común las clases hijas ,y cada clase hija tiene sus **características/parámetros** únicos .Aquí se encuentran en orden (como aparecen en el código)los métodos (-) para:

- Imprimir arreglos tipo String

- Imprimir arreglos tipo int

-Metodo to String para imprimir contacto

Alumno Es **clase hija de Persona**. Aquí tenemos las características propias de la clase alumno y su estructura(Parámetros requeridos para ser un Alumno)

Coordinador Es **clase hija de Persona**. Aquí tenemos las características propias de la clase Coordinador y su estructura(Parámetros requeridos para ser un Coordinador).

Profesor Es **clase hija de Persona**. Aquí tenemos las características propias de la clase Profesor y su estructura(Parámetros requeridos para ser un Profesor).

Comparacarrera Aquí construimos el argumento con el que comparara bubble sort.

Comparacorreo Aquí construimos el argumento con el que comparara bubble sort.

Comparatipo Aquí construimos el argumento con el que comparara bubble sort.

Directorio Aquí se encuentran en orden (como aparecen en el código)los métodos (-) ,extras (\*)para :

- Mostrar la información completa de un contacto, con solo proporcionar su nombre (independientemente que sea un alumno, profesor o coordinador).

- Mostrar la información completa de un contacto, dado su nombre y rol (alumno, profesor o director).

- Buscar a una persona con su correo.

- Insertar datos de un nuevo contacto (alumno, profesor o coordinador).

- Eliminar los datos de un contacto, a partir del nombre.

- Borrar personas ingresando su correo.

- Actualizar la información de alguno de los contactos, a partir del nombre.

- Imprimir arreglo.

\*Creación y uso de Bubble Sort

- Mostrar solo alumnos o solo profesores, según la elección del usuario. Los contactos se deben mostrar ordenados de forma ascendente.

- Mostrar a todos los contactos con correo electrónico. La información se mostrará agrupada por categoría.

- Mostrar a todos los contactos de una carrera en particular. La información se mostrará agrupada por categoría (alumno, profesor o coordinador).

\*Estructura de las cuentas con carreras de los profesores

\*Estructura de las cuentas con carrera de los coordinadores

-Mostrar a todos los contactos con un sueldo determinado. La información se mostrará agrupada por categoría (profesor o coordinador).

\*Estructura de las cuentas de sueldos de profesores

-Metodo toString

**MiDirectorio**(Clase Main) Aquí se hace el llamado a los métodos creados siendo llamados desde su respectiva Clase.

Tambien aqui esta definido el menú.

## **Funcionamiento del menú.**

Nuestro menú funciona mediante los ciclos switch ,if,else if ,else ,teniendo desde el caso 1 hasta el caso 6 (SWITCH Menú/opciones), entró de cada caso tiene diferentes sub casos (IF,ELSE IF ,ELSE Submenús/subopciones.)

Repitiéndose las opciones una y otra vez hasta que el usuario decida salir.