

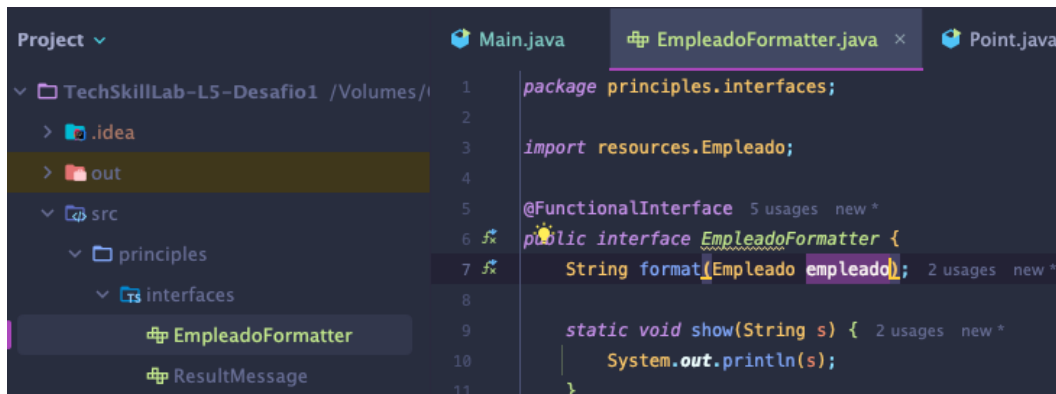
# Capturas Desafío 1

JULIETH TATIANA GARCIA ZULUAGA

Reporte de nómina y ficha personal:

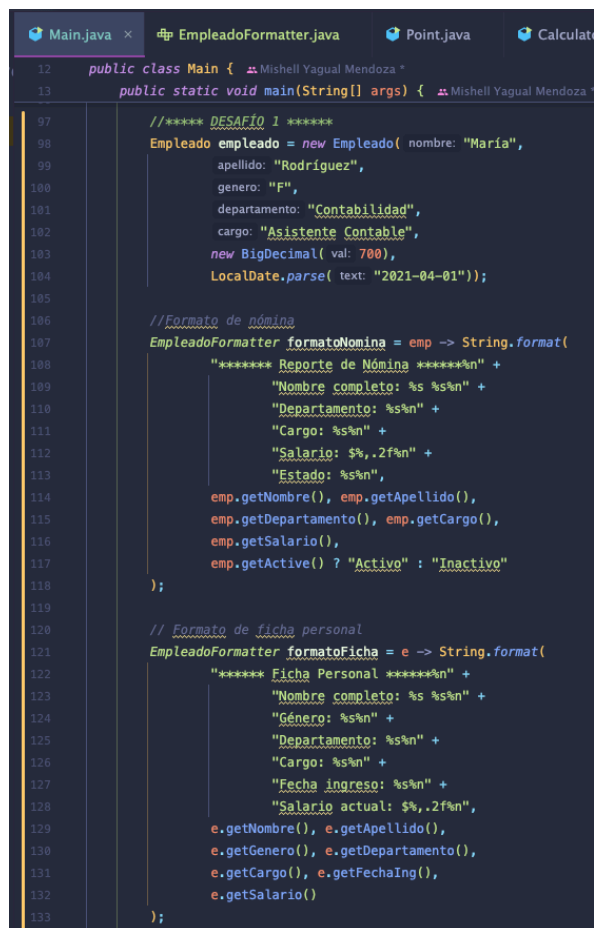
## Implementación

Se crea una interfaz EmpleadoFormatter:



```
Project ▾ Main.java EmpleadoFormatter.java Point.java
TechSkillLab-L5-Desafio1 /Volumes/I
  .idea
  out
  src
    principles
      interfaces
        EmpleadoFormatter
        ResultMessage
1 package principles.interfaces;
2
3 import resources.Empleado;
4
5 @FunctionalInterface 5 usages new *
6 public interface EmpleadoFormatter {
7     String format(Empleado empleado); 2 usages new *
8
9     static void show(String s) { 2 usages new *
10         System.out.println(s);
11     }
```

Se implementan en el Main:



```
Main.java EmpleadoFormatter.java Point.java Calculator
12 public class Main { Mishell Yagual Mendoza *
13     public static void main(String[] args) { Mishell Yagual Mendoza *
14
15         //***** DESAFIO 1 *****
16         Empleado empleado = new Empleado( nombre: "María",
17             apellido: "Rodríguez",
18             genero: "F",
19             departamento: "Contabilidad",
20             cargo: "Asistente Contable",
21             new BigDecimal( val: 700),
22             LocalDate.parse( text: "2021-04-01"));
23
24         //Formato de nómina
25         EmpleadoFormatter formatoNomina = emp -> String.format(
26             "***** Reporte de Nómina *****" +
27             "Nombre completo: %s %s\n" +
28             "Departamento: %s\n" +
29             "Cargo: %s\n" +
30             "Salario: $%,.2f\n" +
31             "Estado: %s\n",
32             emp.getNombre(), emp.getApellido(),
33             emp.getDepartamento(), emp.getCargo(),
34             emp.getSalario(),
35             emp.getActive() ? "Activo" : "Inactivo"
36         );
37
38         // Formato de ficha personal
39         EmpleadoFormatter formatoFicha = e -> String.format(
40             "***** Ficha Personal *****" +
41             "Nombre completo: %s %s\n" +
42             "Género: %s\n" +
43             "Departamento: %s\n" +
44             "Cargo: %s\n" +
45             "Fecha ingreso: %s\n" +
46             "Salario actual: $%,.2f\n",
47             e.getNombre(), e.getApellido(),
48             e.getGenero(), e.getDepartamento(),
49             e.getCargo(), e.getFechaIng(),
50             e.getSalario()
51         );
52     }
```

## Resultado

```
***** Reporte de Nómina *****  
Nombre completo: María Rodríguez  
Departamento: Contabilidad  
Cargo: Asistente Contable  
Salario: $700.00  
Estado: Activo
```

```
***** Ficha Personal *****  
Nombre completo: María Rodríguez  
Género: F  
Departamento: Contabilidad  
Cargo: Asistente Contable  
Fecha ingreso: 2021-04-01  
Salario actual: $700.00
```

**Predicate:** Filtrar empleados activos con salario menor a 700 USD:

## Implementación

```
//To-do: Filtrar empleados activos con salario menor a 700 USD.  
Predicate<Empleado> empActiveSalMenor700 = empleado ->  
    empleado.getActive() && empleado.getSalario().compareTo(new BigDecimal( val: "700")) < 0;
```

```
// PREDICATE
System.out.println("Predicate resultado");
System.out.println("Empleados activos con salario menor a 700 USD");

boolean encontrado = false;

for (Empleado e : empleados) {
    if (empActiveSalMenor700.test(e)) {
        System.out.println(e);
        encontrado = true;
    }
};

if (!encontrado) {
    System.out.println("No existen empleados activos con salario menor a 700 USD.");
}
```

## Resultado

```
Predicate resultado
Empleados activos con salario menor a 700 USD
No existen empleados activos con salario menor a 700 USD.
```

**Function:** Generar un mapa que almacene como clave un departamento y como valor el supervisor de dicho departamento.

## Implementación

```
//To-do: Generar un mapa que almacene como clave un departamento y como valor el supervisor de dicho departamento.
Function<List<Empleado>, Map<String, String>> mapDeptoSupervisor = l_empleados -> {
    Map<String, String> mapa = new HashMap<>();
    l_empleados.forEach(Empleado -> {
        if (Empleado.getCargo().contains("Supervis")) {
            mapa.put(Empleado.getDepartamento(), Empleado.getNombre() + ' ' + Empleado.getApellido());
        }
    });
    return mapa;
};
```

```
//FUNCTION
System.out.println("Function resultado");
System.out.println("Mapa departamento y valor supervisor");
Map<String, String> supervisoresxDepto = mapDeptoSupervisor.apply(empleados);
System.out.println(supervisoresxDepto);
```

## Resultado

```
Function resultado
Mapa departamento y valor supervisor
{Informática=Cecilia Marín, Contabilidad=Camilo López, Talento Humano=Estefanía Mendoza}
```

**Consumer:** Imprimir en consola un listado de empleados de un determinado departamento.

## Implementación

```
//To-do: Imprimir en consola un listado de empleados de un determinado departamento.  
Consumer<List<Empleado>> empInformatica = emps -> {  
    emps.forEach(empleado -> {  
        if(empleado.getDepartamento().contains("Informática")) {  
            System.out.println(empleado);  
        }  
    });  
};
```

```
//CONSUMER  
System.out.println("Consumer resultado");  
System.out.println("Lista de empleados del departamento de Informática");  
empInformatica.accept(empleados);
```

## Resultado

```
Consumer resultado  
Lista de empleados del departamento de Informática  
{nombre: 'Julían', apellido: 'Flores', genero: 'M', departamento: 'Informática', cargo: 'Soporte TI', salario: 800, fechaIng: 2023-11-01, fechaSal: null, isActive: true}  
  
{nombre: 'Camila', apellido: 'Mendoza', genero: 'F', departamento: 'Informática', cargo: 'Desarrollador UI/UX', salario: 1000, fechaIng: 2021-07-08, fechaSal: null, isActive: true}  
  
{nombre: 'Raúl', apellido: 'Echeverría', genero: 'M', departamento: 'Informática', cargo: 'Infraestructura TI', salario: 950, fechaIng: 2020-02-14, fechaSal: null, isActive: true}  
  
{nombre: 'Julie', apellido: 'Flores', genero: 'F', departamento: 'Informática', cargo: 'Desarrollador', salario: 1200, fechaIng: 2021-12-10, fechaSal: null, isActive: true}  
  
{nombre: 'José', apellido: 'Rodríguez', genero: 'M', departamento: 'Informática', cargo: 'Tester QA', salario: 1100, fechaIng: 2021-10-01, fechaSal: null, isActive: true}  
  
{nombre: 'Cecilia', apellido: 'Marín', genero: 'F', departamento: 'Informática', cargo: 'Supervisora TI', salario: 2000, fechaIng: 2020-04-21, fechaSal: null, isActive: true}  
  
{nombre: 'Edison', apellido: 'Cáceres', genero: 'M', departamento: 'Informática', cargo: 'Desarrollador TI', salario: 1300, fechaIng: 2023-07-07, fechaSal: null, isActive: true}
```

**Comparator:** Ordena a los empleados por su apellido en orden alfabético.

## Implementación

```
//To-do: Ordenar empleados por su apellido en orden alfabético
Comparator<Empleado> porApellido = Comparator.comparing(empleado -> empleado.getApellido());
```

```
//COMPARATOR
System.out.println("Comparator resultado");
System.out.println("Empleados ordenados por su apellido");
List<Empleado> empleados1 = new ArrayList<>(List.copyOf(empleados));
empleados1.sort(porApellido);
System.out.println(empleados1);
```

## Resultado

```
Comparator resultado
Empleados ordenados por su apellido
[{nombre: 'José', apellido: 'Albornoz', genero: 'M', departamento: 'Contabilidad', cargo: 'Asistente Contable', salario: 800, fechaIng: 2020-08-15, fechaSal: 2023-05-01, isActive: false}
, {nombre: 'Edison', apellido: 'Cáceres', genero: 'M', departamento: 'Informática', cargo: 'Desarrollador TI', salario: 1300, fechaIng: 2023-07-07, fechaSal: null, isActive: true}
, {nombre: 'Raúl', apellido: 'Echeverría', genero: 'M', departamento: 'Informática', cargo: 'Infraestructura TI', salario: 950, fechaIng: 2020-02-14, fechaSal: null, isActive: true}
, {nombre: 'Julián', apellido: 'Flores', genero: 'M', departamento: 'Informática', cargo: 'Soporte TI', salario: 800, fechaIng: 2023-11-01, fechaSal: null, isActive: true}
, {nombre: 'Julie', apellido: 'Flores', genero: 'F', departamento: 'Informática', cargo: 'Desarrollador', salario: 1200, fechaIng: 2021-12-10, fechaSal: null, isActive: true}
, {nombre: 'Carlos', apellido: 'Franco', genero: 'M', departamento: 'Talento Humano', cargo: 'Reclutador', salario: 650, fechaIng: 2023-01-07, fechaSal: 2024-12-09, isActive: false}
, {nombre: 'Juan', apellido: 'Gutierrez', genero: 'M', departamento: 'Talento Humano', cargo: 'Reclutador', salario: 500, fechaIng: 2023-03-11, fechaSal: 2024-04-01, isActive: false}
, {nombre: 'Esteban', apellido: 'Gutierrez', genero: 'M', departamento: 'Talento Humano', cargo: 'Reclutador', salario: 700, fechaIng: 2023-04-01, fechaSal: null, isActive: true}
, {nombre: 'Manuel', apellido: 'Játiva', genero: 'M', departamento: 'Contabilidad', cargo: 'Asistente Contable', salario: 850, fechaIng: 2023-06-03, fechaSal: null, isActive: true}
, {nombre: 'Camilo', apellido: 'López', genero: 'M', departamento: 'Contabilidad', cargo: 'Supervisor Contable', salario: 1500, fechaIng: 2020-04-11, fechaSal: null, isActive: true}
, {nombre: 'María', apellido: 'López', genero: 'F', departamento: 'Contabilidad', cargo: 'Asistente Contable', salario: 840, fechaIng: 2020-02-20, fechaSal: 2024-07-15, isActive: false}
, {nombre: 'Cecilia', apellido: 'Marín', genero: 'F', departamento: 'Informática', cargo: 'Supervisora TI', salario: 2000, fechaIng: 2020-04-21, fechaSal: null, isActive: true}
, {nombre: 'Camila', apellido: 'Mendez', genero: 'F', departamento: 'Contabilidad', cargo: 'Asistente Cuentas', salario: 860, fechaIng: 2020-10-01, fechaSal: null, isActive: true}
, {nombre: 'Camila', apellido: 'Mendoza', genero: 'F', departamento: 'Informática', cargo: 'Desarrollador UI/UX', salario: 1000, fechaIng: 2021-07-08, fechaSal: null, isActive: true}
, {nombre: 'Estefanía', apellido: 'Mendoza', genero: 'F', departamento: 'Talento Humano', cargo: 'Supervisora TM', salario: 1600, fechaIng: 2021-09-21, fechaSal: null, isActive: true}
, {nombre: 'Melissa', apellido: 'Morochó', genero: 'F', departamento: 'Contabilidad', cargo: 'Asistente Contable', salario: 820, fechaIng: 2022-05-22, fechaSal: 2023-07-09, isActive: false}
, {nombre: 'María', apellido: 'Rodríguez', genero: 'F', departamento: 'Contabilidad', cargo: 'Asistente Contable', salario: 700, fechaIng: 2021-04-01, fechaSal: null, isActive: true}
, {nombre: 'José', apellido: 'Rodríguez', genero: 'M', departamento: 'Informática', cargo: 'Tester QA', salario: 1100, fechaIng: 2021-10-01, fechaSal: null, isActive: true}
, {nombre: 'María', apellido: 'Silva', genero: 'F', departamento: 'Contabilidad', cargo: 'Asistente Contable', salario: 900, fechaIng: 2021-11-15, fechaSal: 2022-08-09, isActive: false}
}]
```