

# INTERFACES

## OBJETIVOS

---

Poner en practica el uso de interfaces en java.

Entender la utilidad de las interfaces en java.

Saber cuando y como utilizar las interfaces en java.

## ACTIVIDAD

---

1. En un centro médico se tiene:

### **Hospital**

Nombre.  
Dirección.  
Telefono.  
Fax.

### **Paciente**

IdPaciente  
DNI  
NombreApellido.  
Dirección  
Teléfono.

### **Personal**

DNI  
Dirección  
Teléfono.  
Cargo.



Desarrolle la Interfaz para Hospital, Paciente, Personal.  
Implemente las Interfaces y sus métodos.

Agrupe todo en un paquete de nombre **Hospital**

Generar el archivo **Hospital.jar** para poder distribuirlo a los usuarios finales.

## Resolución:

Creamos el proyecto Practice01 y dentro, el paquete hospital.

Creamos la interfaz Entity

```
package hospital;

public interface Entity {

    void setDirection(String direction);

    String getDirection();

    void setTelephone(String telephone);

    String getTelephone();

}
```

Creamos la clase Patient e implementamos la interfaz Entity

```
package hospital;

public class Patient implements Entity {

    private String id_patient, dni, name_lastname, direction, telephone;
```

```

@Override
public void setDirection(String direction) {
    this.direction = direction;
}

@Override
public String getDirection() {
    return direction;
}

@Override
public void setTelephone(String telephone) {
    this.telephone=telephone;
}

@Override

public String getTelephone() {
    return telephone;
}

public String getId_patient() {
    return id_patient;
}

public void setId_patient(String id_patient) {
    this.id_patient = id_patient;
}

public String getDni() {
    return dni;
}

public void setDni(String dni) {

```

```

        this.dni = dni;
    }

    public String getName_lastname() {
        return name_lastname;
    }

    public void setName_lastname(String name_lastname) {
        this.name_lastname = name_lastname;
    }
}

```

Creemos la clase Personal e implementamos la interfaz Entity

```

package hospital;

public class Personal implements Entity{

    private String dni,direction,telephone,postition;

    @Override
    public void setDirection(String direction) {
        this.direction=direction;
    }

    @Override
    public String getDirection() {
        return direction;
    }

    @Override
    public void setTelephone(String telephone) {
        this.telephone=telephone;
    }

    @Override

```

```

        public String getTelephone() {
            return telephone;
        }

        public String getDni() {
            return dni;
        }

        public void setDni(String dni) {
            this.dni = dni;
        }

        public String getPostition() {
            return postition;
        }

        public void setPostition(String postition) {
            this.postition = postition;
        }

    }

```

Creamos la clase Hospital e implementamos la interfaz Entity

```
package hospital;
```

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

```

public class Hospital implements Entity{

    private String name, direction,telephone,fax;
    public ArrayList<Patient> patients = new ArrayList();
    public ArrayList<Personal> personals = new ArrayList();

```

```
public void setPersonals(ArrayList<Personal> personals) {  
    this.personals = personals;  
}
```

```
@Override  
public void setDirection(String direction) {  
    this.direction=direction;  
}
```

```
@Override  
public String getDirection() {  
    return direction;  
}
```

```
@Override  
public void setTelephone(String telephone) {  
    this.telephone=telephone;  
}
```

```
@Override  
public String getTelephone() {  
    return telephone;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```

        public String getFax() {
            return fax;
        }

        public void setFax(String fax) {
            this.fax = fax;
        }

    }

```

Para realizar la prueba creamos la clase Main

```

package hospital;

public class Main {
    public static void main(String[] args) {
        Patient patient_one=new Patient();
        patient_one.setDirection("A.V Los Rosales 451");

        Hospital hospital_one=new Hospital();

        hospital_one.patients.add(patient_one);
        System.out.println(hospital_one.patients.get(0).getDirection());

    }
}

```

## RESULTADO

---

Creamos el archivo jar:

Cd NetBeansProjects

jar -cf Hospital.jar Practice01

Ejecutamos:

Java -classpath Hospital.jar Main.java

```
run:
A.V Los Rosales 451
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

**Repositorio GitHub**

**<https://github.com/Adrian18218217/SistemasDistribuidos>**

## CONCLUSIONES

---

En síntesis, es necesario utilizar interfaces cuando tenemos diferentes clases que requieren de métodos comunes, aunque con una diferente implementación, las interfaces nos permiten la herencia múltiple.

Se aprendió o recordó el uso de las interfaces en java.