



Programación y Laboratorio II

Clase 13

Interfaces

Interfaces

- ¿Qué es una interfaz?
- Usos y Generalidades

Implementación de interfaces

- Definición e Implementación
- Interfaces Explícitas



01.

Interfaces



¿Qué es una **Interfaz**?

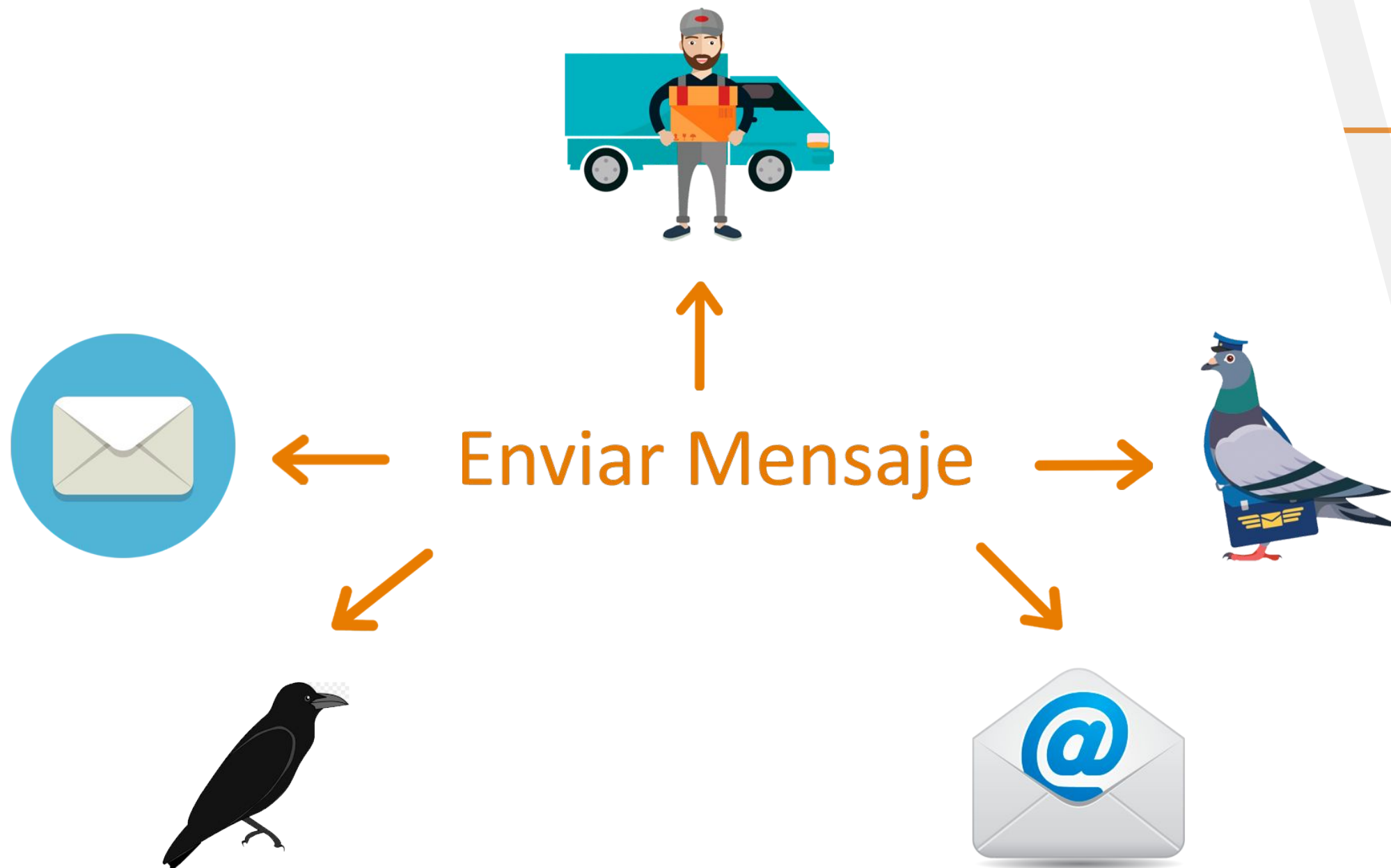
Una interfaz es un medio común que nos permite agrupar **funcionalidades** que una clase luego podría implementar.

Se puede decir que las interfaces establecen un **contrato** en el cual las clases que implementan la interfaz están **obligadas** a implementar sus funcionalidad



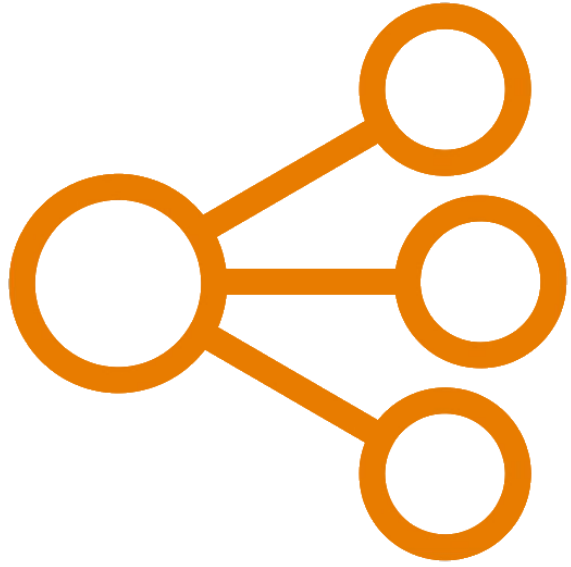
Interfaces

Como las interfaces nos permiten tener la funcionalidad separada de la implementación puedes tener distintas implementaciones en diferentes clases sin una relación fuerte pero con la misma funcionalidad.



Usos y Generalidades

- Nos permiten definir métodos, propiedades, indexadores y eventos.
- Una interfaz no permite definir atributos ni constructores.
- Una clase puede implementar varias interfaces
- Todas las funcionalidades definidas en una interfaz son públicas sin posibilidad de modificación



02.

Implementación de interfaces

Definicion e Implementacion

```
public interface IMensaje
{
    string EnviarMensaje();
}
```

- Para definir una interfaz declaramos su modificador de visibilidad seguido de la palabra reservada **interface** y su nombre, por convención este siempre comienza con la letra I mayúscula.
- Los métodos o propiedades definidos no llevan modificar de visibilidad Y por defecto van a ser siempre públicos.

Para implementar una interfaz lo único que debemos hacer es emplear el operador ":" seguido de la interfaz.

```
public class Email:IMensaje
{
    public string EnviarMensaje()
    {
        return "Ya falta mucho para implementar? :)";
    }
}
```

Si queremos implementar una interfaz en una clase derivada. la clase base siempre va a ir primero y luego irá la interfaz o interfaces separadas por una coma ",".

```
public class Carta :Papel,IMensaje
{
    public string EnviarMensaje()
    {
        return "Esta es una declaracion de guerra";
    }
}
```

Usos y Generalidades

- Si una clase implementa más de una interfaz que contiene métodos con la misma definición, la llamada a ese método implementa el mismo método y el método específico de la interfaz.
- Usando la implementación explícita de una interfaz podemos decirle al compilador el método de que interfaz vamos a implementar y vamos a poder proveer diferente funcionalidad a los métodos de estas distintas interfaces.
- Del lado de la interfaz no vamos a ver reflejado ningún cambio ya que los cambios son en las implementaciones de estos métodos.
- Otra razón para utilizar interfaces explícitas es ocultar la implementación de los elementos de la interfaz para que no sean fácilmente accesibles.

Interfaces Explícitas

Para lograr esto colocamos delante del nombre del método el nombre de la interfaz que estamos implementando y no le indicamos el modificador de visibilidad.

```
public class Cuervo :IMensaje, IEncriptado
{
    public string EnviarMensaje()
    {
        return "Llego el invierno";
    }

    string IEncriptado.EnviarMensaje()
    {
        return "Jon Snow es el verdadero rey";
    }
}
```

← Forma Implícita

← Forma Explícita

```
static void Main(string[] args)
{
    Cuervo cuervo = new Cuervo("Bloodraven");

    //Casteamos al tipo de interfaz
    ((IEncriptado)cuervo).EnviarMensaje()
}
```

Para convocar el método o propiedad implementado explícitamente debemos castear nuestro objeto al tipo de la interfaz.

Ejercicios



- I01 - Cartuchera
- I02 - Control de aduana

https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/

Tarea



- C01 - La centralita episodio 4

https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/