



# PROGRA\_AMANDO

holamundo.co

# Agenda

Sobrecarga

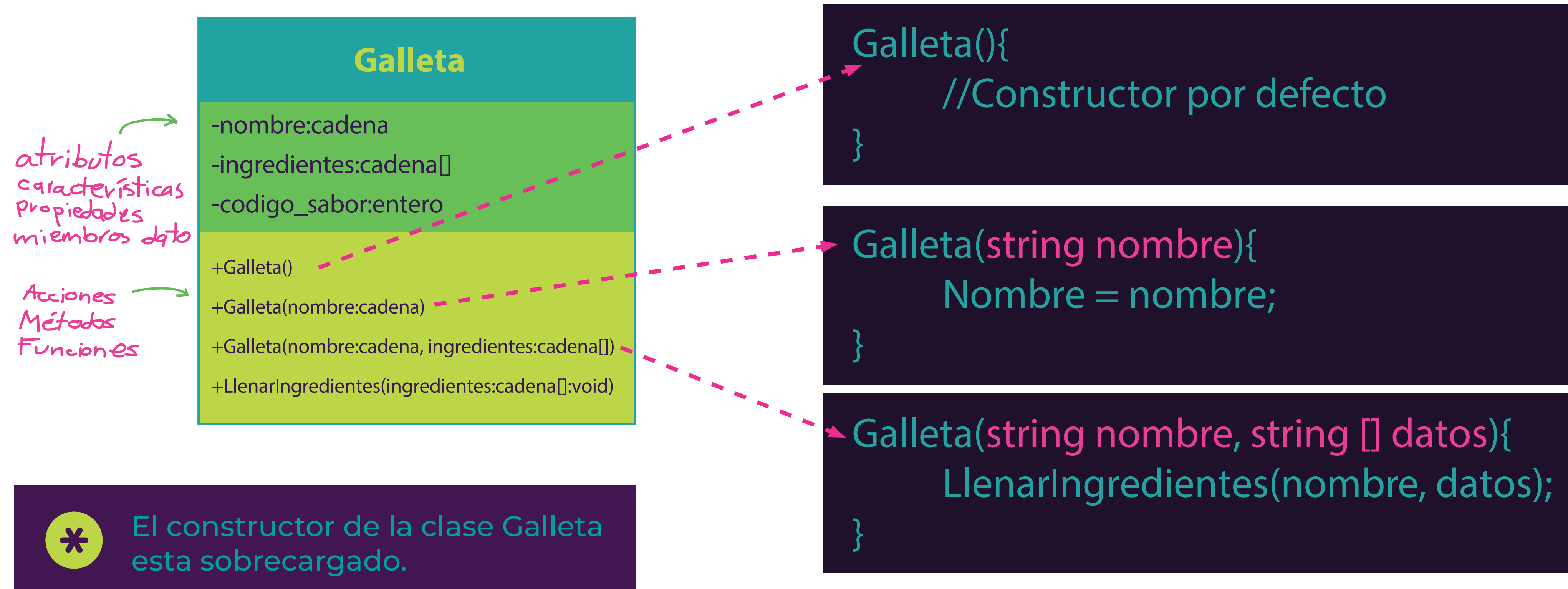
Sobre escritura

Ocultamiento

Clase Abstracta

# Sobrecarga

Tiene en cuenta la existencia de una acción o funcionalidad (método o constructor) que va a ser usada de diferentes maneras, pero con resultados similares. Esto con el fin de especializar su propósito, hacer más versátil la clase manteniendo la alta cohesión. Ya se han usado métodos similares, por ejemplo `writeLine`.



# Reglas para usar sobrecarga

**1**

Crear una función o un constructor de clase

**2**

Crear una nueva función o constructor con el mismo nombre

**3**

Cambiar el número de parámetros, variar el tipo o el orden de estos

**4**

Para invocar un método específico, se tiene en cuenta los argumentos que recibe el método sobrecargado

**5**

No hay limite de sobrecargas, desde que el compilador tenga una forma de diferenciar los métodos

**\***

NO se debe cambiar el nombre del método o el tipo de retorno

# Ejemplo de uso

Sobrecarga de constructores, observe como se pueden crear diferentes instancias de Galleta, con distintos valores iniciales, esto gracias a la sobrecarga.

```
Galleta alfajor = new Galleta("Alfajor");  
Galleta salada = new Galleta(false);  
Galleta pasabocas = new Galleta(20);
```

Se llaman diferentes constructores, pasando diferentes tipos de dato, como argumento

```
Galleta g = new Galleta("Sultana",false);
```

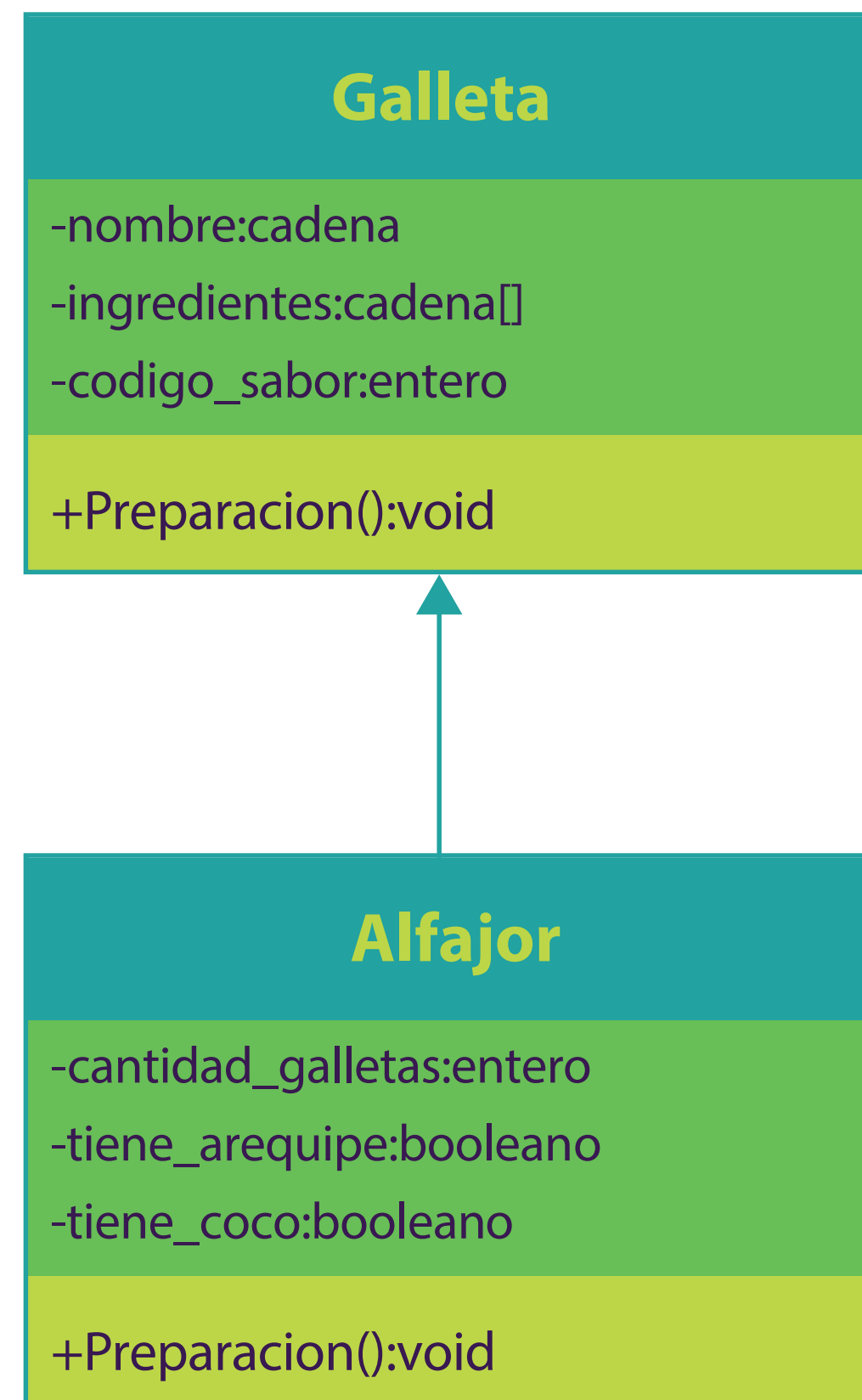
Se llama otro constructor, en este caso se varia el número de argumentos



Ejercicio: Escriba la implementación de los constructores del ejemplo

# Sobreescritura

Se presenta cuando hay herencia entre dos clases y es necesario redefinir la implementación de un método de la clase padre en la clase hija. Para que esto suceda, es necesario realizar una sobreescritura de un método heredado.



```
...  
public virtual void Preparacion(){  
    //Descripción genérica  
}  
...
```

```
...  
public override void Preparacion(){  
    //Descripción específica + toque secreto  
}  
...
```

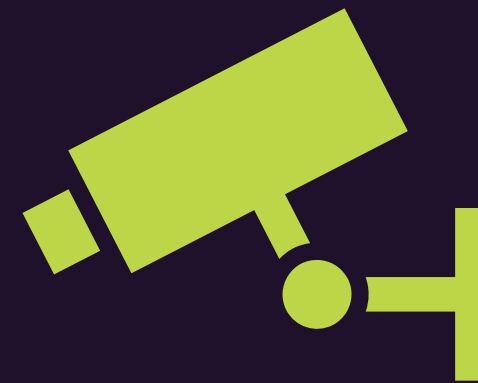


# Ocultamiento

```
...  
  
public void Preparacion(){  
    //Descripción genérica  
}  
  
...
```

Funcionalidad que permite re-definir la funcionalidad de un método si necesidad de usar sobreescritura, se usa la palabra reservada `new` como modificador de la firma del método que realiza el ocultamiento.

```
...  
  
public new void Preparacion(){  
    /*Descripción nueva, oculta funcionalidad  
    del padre */  
}  
  
...
```



Se puede usar para mantener el diseño de una clase padre, de tal forma que no es necesario marcar un método como virtual, para re-definir su funcionalidad.



# Clase abstracta

Cuando se habla en **“abstracto”** no se hace referencia a algo en particular, se habla entonces de una manera general, que reúne o expresa las diferentes partes que componen un concepto.

Las clases abstractas, mantienen la implementación de un concepto de manera muy general, para que sirva de punto de partida o referencia para explicar o representar otros conceptos.





# Las clases abstractas



Debe ser extendida



No puede ser instanciada directamente



Si tiene métodos abstractos, la clase hija debe realizar la implementación del método



Puede tener métodos con solo la firma y sin implementación



Debe ser marcada como abstracta (abstract)

# Las clases abstractas

Las clases abstractas también pueden tener métodos con implementación, para que sean usados por clases hijas.

```
abstract class Comida{  
    public abstract void ImprimirTablaNutricional();  
    public void Descripcion(){  
        Console.WriteLine("Descripcion general");  
    }  
}
```

```
class Program{  
    static void Main(string[] args){  
        Galleta g = new Galleta();  
        g.ImprimirTablaNutricional();  
        ((Comida)g).Descripcion();  
        Console.ReadKey();  
    }  
}
```

El método puede ser usado desde el objeto galleta, pero es necesario realizar un casting

# Ejercicios

- Sobrecargue 5 veces el constructor de una clase
- Sobreescrba dos métodos de una clase padre
- ¿Se puede sobrecargar y sobreescibir un método?
- ¿Cuántas sobrecargas puede tener un constructor?
- ¿Es posible asignar una instancia de una clase hija a una variable de tipo padre abstracta?
- ¿Cuales son las palabras claves de c sharp para realizar sobreescritura?
- Cree la clase abstracta Vehiculo, luego cree clases hijas que extiendan de la clase Padre, sobreescrba el método Descripcion, por último cree un array de Vehiculo y asigne instancias de las clases hijas y responda:
  - ¿Es posible crear un array de una clase abstracta?
  - ¿Es necesario tener una relación entra las clases hijas?
  - ¿Es posible acceder a métodos de la clase padre y de la clase hija?
  - ¿Es obligatorio sobreescibir los métodos de la clase padre?
- Cree una herencia de tres 5 niveles, sobreescrba un método en cada nivel. Luego cree instancias y use los métodos sobreescritos.
- Cree dos métodos según la invocación: `new Hijo1().Prueba(5);` `new Hijo2().Prueba(.4);`