

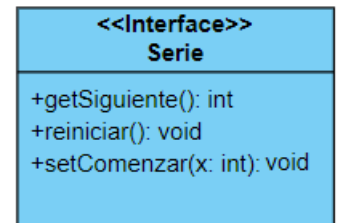
# Ejercicios de Interfaces en P00



## Relación de ejercicios

**Ejercicio 1.** Implementa una interfaz llamada *Serie*. Dicha interfaz debe definir los siguientes métodos. Recuerda que la interfaz no va a tener código dentro de los métodos (cuerpo del método). Serán las clases que implementen dicha interfaz la que tendrán el código necesario.

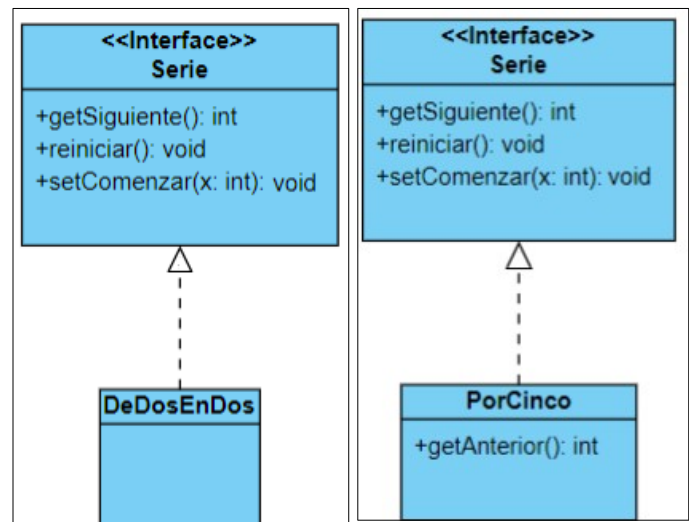
- `getSiguiente()`: devolverá un entero con el siguiente número de la serie en función de la clase que lo implemente.
- `reiniciar()`: permitirá reiniciar el valor de la serie.
- `setComenzar(int x)`: recibirá un entero y establecerá el valor inicial de la serie al valor del parámetro pasado.



**Ejercicio 2.** Implementa una clase llamada *DeDosEnDos* que implemente la interfaz *Serie*. Esta clase deberá devolver números de 2 en 2. Por ejemplo: 0, 2, 4, 6, 8, etcétera.

**Ejercicio 3.** Implementa una clase llamada *PorCinco* que implemente la interfaz *Serie*. Esta clase deberá devolver números multiplicados por 5. Por ejemplo: 5, 25, 125, etcétera. Además, tendrá un método llamado `getAnterior()` que devolverá el número

anterior de la serie. Por ejemplo, si actualmente la serie tiene el valor 125, deberá devolver el 25. El diagrama puede que esté incompleto. Toma las decisiones que creas conveniente para implementar la solución al ejercicio.



**Ejercicio 4.** Implementa un programa que genere los primeros 10 números de cada una de las series anteriores y los muestre por pantalla. Además, una vez concluida la serie *PorCinco* (cuando se hayan mostrado los 10 valores) se debe mostrar el valor anterior de nuevo (el noveno generado).

**Ejercicio 5.** Implementa una interfaz llamada `Animal` que tenga dos métodos: un método llamado `onomatopeya()` que devuelva una cadena de texto; otro método, llamado `getEspecie()`, que devuelva la especie del animal. Para ello, implementa un enumerado, llamado `AnimalEspecie`, que tenga los siguientes valores: Pavo, Caballo, Pájaro, Ganso, Gallina, Gallo, Perro, Vaca y Gato.

**Ejercicio 6.** Crea una clase, por cada una de las especies anteriores de animales, que implemente la interfaz `Animal`. Todas las clases tienen que tener un atributo de tipo `AnimalEspecie` que *setee* su valor en el constructor sin parámetros.

**Ejercicio 7.** Implementa una aplicación, llamada `SonidosAnimalesApp`, que tenga las siguientes características y/o funcionalidades:

- Debe tener un diccionario de clave `AnimalEspecie` y de valor la interfaz `Animal`.
- Cada animal, de los anteriores, se debe incluir en el diccionario con su correspondiente clase.
- Debe existir un método que permita conocer las onomatopeyas de todos los animales. Se indica un ejemplo de salida por pantalla.

```
La onomatopeya del animal 'GANSO' es 'On, on'
La onomatopeya del animal 'GATO' es 'Miau'
La onomatopeya del animal 'VACA' es 'Muu, muu'
La onomatopeya del animal 'CABALLO' es 'Hiii'
La onomatopeya del animal 'GALLINA' es 'Co, co, co'
La onomatopeya del animal 'PERRO' es 'Guau, guau'
La onomatopeya del animal 'PAVO' es 'Glu, glu'
La onomatopeya del animal 'PAJARO' es 'Pío, pío'
La onomatopeya del animal 'GALLO' es 'Quiquiriquí'
```

**Ejercicio 8.** Implementa un programa para la gestión del precio de alquiler de una serie de pisos. Puedes reaprovechar las clases implementadas en otros temas y/o ejercicios. Vamos a hacer uso también de la lógica de las dos clases de inmobiliarias que hemos creado. Posteriormente, piensa cómo podríamos hacer uso de interfaces para mejorar un programa que nos muestre el precio de 4 pisos en función de la lógica de cada una de las inmobiliarias.

Piso
-numPortal: int
-numPlanta: int
-letraVivienda: LetraVivienda
-m2: short
-mobiliario: Mobiliario
-piscina: boolean
+getMensajeDerechoPiscina(): String
+getMensajeMobiliario(): String
+descripcion(): String

**Ejercicio 9.** Implementa lo necesario para que, haciendo uso de la interfaz creada en este [ejercicio](#), el usuario pueda ver las primeras 10 ocurrencias de los números del 1 al 20. Por ejemplo, la serie del 1 sería: 1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|; la del 2 sería: 2|4|6|8|10|12|14|16|18|20|22|24|26|28|30|32|34|36|38|40|; y así sucesivamente. Se indica la salida que debe mostrar al usuario por pantalla.

```
Serie del 1: 1|2|3|4|5|6|7|8|9|10|
Serie del 2: 2|4|6|8|10|12|14|16|18|20|
Serie del 3: 3|6|9|12|15|18|21|24|27|30|
Serie del 4: 4|8|12|16|20|24|28|32|36|40|
Serie del 5: 5|10|15|20|25|30|35|40|45|50|
Serie del 6: 6|12|18|24|30|36|42|48|54|60|
Serie del 7: 7|14|21|28|35|42|49|56|63|70|
Serie del 8: 8|16|24|32|40|48|56|64|72|80|
Serie del 9: 9|18|27|36|45|54|63|72|81|90|
Serie del 10: 10|20|30|40|50|60|70|80|90|100|
Serie del 11: 11|22|33|44|55|66|77|88|99|110|
Serie del 12: 12|24|36|48|60|72|84|96|108|120|
Serie del 13: 13|26|39|52|65|78|91|104|117|130|
Serie del 14: 14|28|42|56|70|84|98|112|126|140|
Serie del 15: 15|30|45|60|75|90|105|120|135|150|
Serie del 16: 16|32|48|64|80|96|112|128|144|160|
Serie del 17: 17|34|51|68|85|102|119|136|153|170|
Serie del 18: 18|36|54|72|90|108|126|144|162|180|
Serie del 19: 19|38|57|76|95|114|133|152|171|190|
Serie del 20: 20|40|60|80|100|120|140|160|180|200|
```