



# Introducción a la Programación Orientada a Objetos

DCIC - UNS  
Primer Cuatrimestre de 2010  
PARCIAL Nº1



|           |          |                               |  |
|-----------|----------|-------------------------------|--|
| Apellido: |          | Nombre:                       |  |
| LU:       | Carrera: | ¿Cursa por primera vez IPOO?: |  |

## 1. Resuelva el siguiente problema:



La Liga de la Justicia (JL), es un equipo de superhéroes de cómics conformado por los principales personajes del ficticio Universo DC. La empresa Rockstar ha comenzado el diseño de un juego para la PlayStation3 sobre la JL. El objetivo principal de Rockstar es poder representar en el juego como los superhéroes combaten con supervillanos.

Para el diseño del juego se incluyeron las clases Superheroe y Villano, como se pueden apreciar en el diagrama de clases que se incluye. También se incluyó la clase Poder que se describirá a continuación; todo superhéroe o villano tiene un poder, el cual se presenta con un arreglo de  $n$  enteros. El rol que cumple el Poder en este juego es el de permitir definir quién ganará una batalla entre un superhéroe y un villano. También, todo superhéroe puede tener un compañero (por ejemplo Batman y Robin) y también puede tener un archienemigo (por ejemplo Superman y Lex Luthor), esto será importante al momento de definir el resultado de una batalla. Lo mismo ocurre con los villanos, todo villano puede tener un compañero y también puede tener un archienemigo. Una aclaración importante, si un superhéroe X es compañero de Y, entonces Y es compañero de X; esto no ocurre necesariamente con los supervillanos. También se incluyó en el diseño del juego la clase JL, que representa a la Liga de la Justicia. En ella se almacenarán todos los superhéroes y todos los villanos del juego.

| Superhéroe                                                                                                                                                                                                                                                                                      | Poder                                                                                                                                                                                                       | Villano                                                                                                                                                                                                                                                                                                                                                                                                                                                                | JL                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;&lt;atributos de instancia&gt;&gt; String nombreReal; String nombreSuper; Superhéroe compañero; Villano archienemigo; Poder suPoder;  -- constructores -- Poder(int n) {   &lt;&lt;consulta&gt;&gt;   obtenerPoder() : int[];   &lt;&lt;comando&gt;&gt;   nuevoPoder(int[] p); }</pre> | <pre>&lt;&lt;atributos de instancia&gt;&gt; int[] codigo;  -- constructores -- Poder(int n) {   &lt;&lt;consulta&gt;&gt;   obtenerPoder() : int[];   &lt;&lt;comando&gt;&gt;   nuevoPoder(int[] p); }</pre> | <pre>&lt;&lt;atributos de instancia&gt;&gt; String nombreReal; String nombreVil; Villano compañero; Superhéroe archienemigo; Poder suPoder;  -- constructores -- -- consultas -- obtenerNombreReal() : String; obtenerNombreVil() : String; obtenerCompañero() : Superhéroe; obtenerArchienemigo() : Villano; obtenerPoder() : Poder; -- comandos -- asociarCompañero(Superhéroe c); asociarArchienemigo(Villano v); asociarPoder(Poder p); batallar(Villano v);</pre> | <pre>&lt;&lt;atributos de instancia&gt;&gt; Superhéroe[] buenos; Villano[] malos;  -- constructores -- JL(int n, int m) {   &lt;&lt;consulta&gt;&gt;   obtenerHeroe(int i) : Superhéroe;   obtenerHeroe(int i) : Superhéroe;   obtenerVillano(int i) : Villano;   obtenerVillano(int i) : Villano;   &lt;&lt;comando&gt;&gt;   agregarHeroe(Superhéroe h);   agregarVillano(Villano v);   guerra(); }</pre> |

### Descripciones de operaciones:

- Batallar(Villano V)** (De la clase Superhéroe). Esta operación toma como parámetro de entrada un villano **V** y no retorna nada. Esta operación es un **comando**. La batalla entre un superhéroe y un villano se resuelve de la siguiente manera: Primero se obtiene el poder del superhéroe. Si el superhéroe tiene un compañero entonces se obtiene el poder del compañero. Si el superhéroe tiene un compañero entonces se creará un nuevo objeto instancia de la clase Poder. Este nuevo poder tendrá como longitud el máximo de las longitudes de los poderes anteriores (el poder del superhéroe y el de su compañero) y el contenido será la suma de los dos arreglos elemento a elemento (hasta donde sea posible, cuando no sea posible se copiará el elemento de mayor longitud). Este nuevo poder que hemos creado se llamará **PS**. Si el superhéroe no tiene compañero entonces **PS** es igual al poder del superhéroe. A continuación se realizan los mismos pasos pero para el villano, llamaremos al poder resultante de estos pasos **PV**. Si el villano **V** es el archienemigo del superhéroe entonces los arreglos en **PS** y en **PV** serán multiplicado por dos números aleatorios **a** y **b** respectivamente (los números aleatorios estarán entre 1 y 5). A continuación se debe calcular cuantos elementos en **PS** son mayores que sus contrapartidas en **PV** (por contrapartidas nos referimos a la misma posición en el otro arreglo) denominaremos a este resultado **nPS**. Lo mismo se hace para cada elemento en **PV** y se denomina **nPV**. Si **nPS > nPV**, entonces decimos que la batalla la ganó el superhéroe, si **nPS < nPV** entonces la batalla la ganó el villano y si son iguales es un empate. Si el superhéroe gana la batalla entonces el arreglo que se encuentra en el poder del villano será reducido a todos 1's. Si el villano gana la batalla, entonces el arreglo que se encuentra en el poder del superhéroe será reducido a todos 1's. Si hay empate, ningún poder se modifica.

- Batallar(Superhéroe S)** (De la clase Villano). Idem anterior.
- Guerra()** (De la clase JL). Esta operación hará batallar a cada superheroe con cada villano.
- JL(int n, m)** (De la clase JL). Este es el constructor de la clase JL, los parámetros **n** y **m** son las longitudes de los arreglos de superhéroes y villanos respectivamente.
- Poder(int n)** (De la clase Poder). El parámetro **n** representa la cantidad de enteros en el arreglo, el contenido del arreglo será generado en forma aleatoria.



**Introducción a la Programación Orientada a Objetos**  
DCIC - UNS  
Primer Cuatrimestre de 2010  
**PARCIAL Nº1**



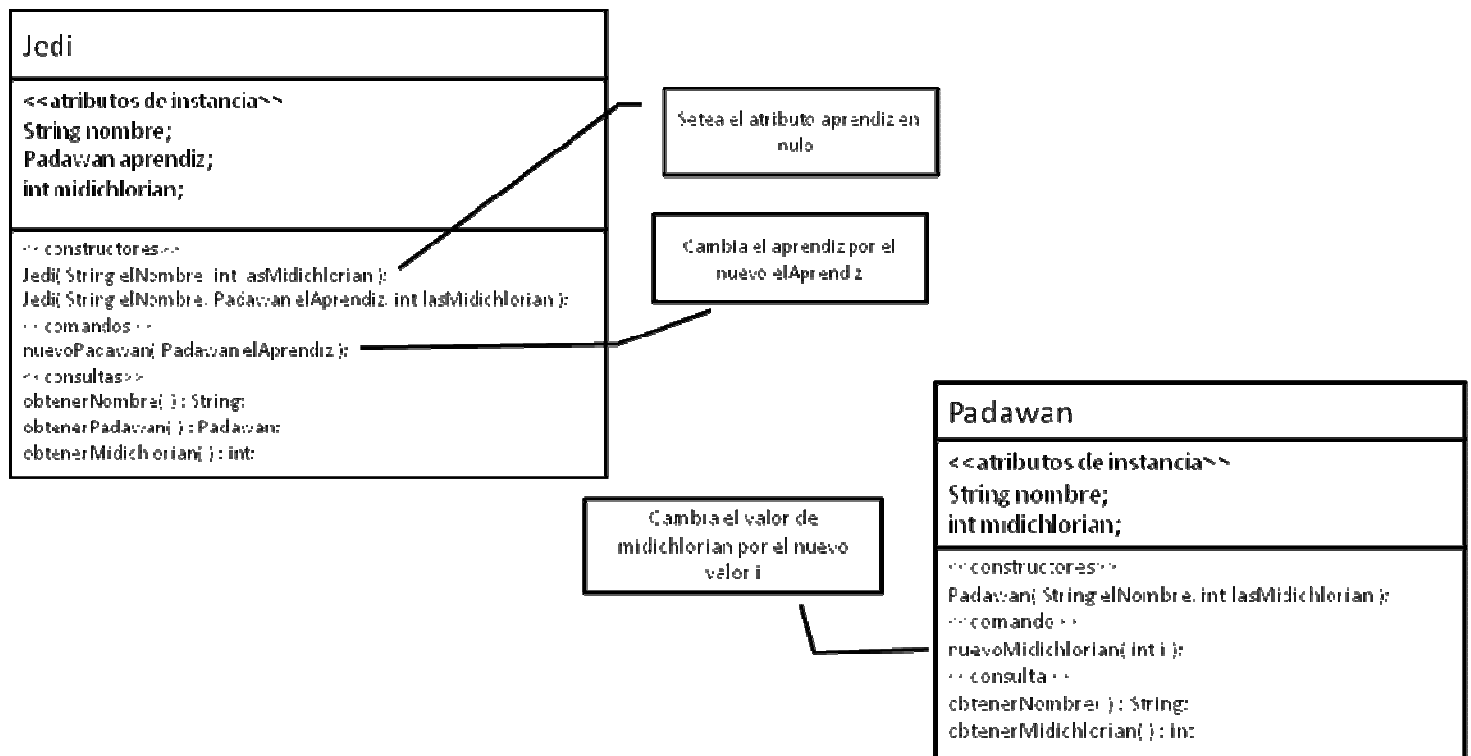
**Usted deberá:** Implementar las cuatro clases en Java con todas sus operaciones. Decidir cuáles son los constructores apropiados para Superhéroe y Villano e implementarlos.

**2. Resuelva el siguiente ejercicio:**

Dado un número  $n$  y tres dígitos  $x$ ,  $a$  y  $b$  se llama **mezcla** a la inserción del dígito  $x$  entre cada aparición (consecutiva) de los dígitos  $a$  y  $b$ . Si  $n=2357359$  y los dígitos son  $x=6$ ,  $a=3$  y  $b=5$ ; el resultado sería **236573659**.

- Escriba un planteo recursivo para realizar la **mezcla** de un número  $n$ , a partir de los dígitos  $x$ ,  $a$  y  $b$ .
- Escriba un método recursivo en Java para implementar el planteo recursivo anterior. El método deberá retornar el número mezclado.

**3. Considerando la descripción de clases que acompaña a este ejercicio, mostrar el diagrama de objetos resultante de la ejecución del código mostrado más abajo y luego contestar las preguntas justificando apropiadamente su respuesta.**



**Código**

```
1) Padawan p1 = new Padawan( "Anakin", 1000 );
2) Padawan p2 = new Padawan( "Kina", 500 );
3) Jedi j1 = new Jedi( "ObiWan", p1, 700);
4) Jedi j2 = new Jedi( "QuiGon", p2, 700);
5) p1 = p2
6) Padawan p3 = j2.obtenerPadawan( );
7) p3.nuevoMidichlorian( 1400 );
8) p2.copiar( p3 );
9) p3.copiar( p1 );
10) j1.nuevoPadawan( p1 );
11) p3 = j2.obtenerPadawan( );
```

**Preguntas**

- $j1 == j2$ ?
- $p1 == p2$ ?
- $j1.obtenerPadawan() == j2.obtenerPadawan()$ ?
- $j1.obtenerMidichlorian() < p3.obtenerMidichlorian()$ ?
- $j2 == p3$ ?
- $p1.igual(p2)$ ?
- ¿Existen objetos inaccesibles?