

## EXAMEN PARCIAL 1 - POO

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta implementación de constructores.
- Aplicación adecuada de los conceptos asociados a la POO: Herencia, Polimorfismo, Composición y Agregación.
- Diseño UML y desarrollo en Nassi-Shneiderman

### Enunciado

Una empresa de software que se dedica a crear componentes para el desarrollo de aplicaciones nos pide desarrollar la lógica general del sistema de menús que utilizará de aquí en adelante. Como el prototipo se desarrollará para usar en consola se espera obtener algo similar a lo que se ve aquí debajo:

```
-- Menu Principal --  
1 - Opcion 1  
2 - Opcion 2  
3 - Acceso a Submenu 1  
X - Salir  
Ingrese la opción deseada:
```

Nos piden que creemos un prototipo de **Menu**. Este tendrá un título y una o más opciones de menú (de la clase **OpcionDeMenu**) que por ahora podrán ser solo de dos tipos: **OpcionAccesoAMenu** (que sirve para acceder a su menú secundario, dependiente del menú actual) y **OpcionDeSalida** que se usa simplemente como marca para cortar el ciclo de ejecución del menú (cuando se elige se sale del menú).

Tanto el Menu como sus ítems son *Mostrables* (se pueden mostrar) y *Ejecutables* (se pueden ejecutar).

**Menu** debe tener, además, una opción extra (de la clase **OpcionDeSalida**), la cual se creará con el menú y no debe estar junto a las opciones registradas sino guardarse aparte. Para agregarle opciones (de a una) tendrá un método **registrar(...)**.

Todas las clases de opciones de menú definidas o que se puedan definir en el futuro deben tener una descripción y un caracter que le servirá al usuario para elegir tal opción.

Los datos de pantalla de la clase **OpcionDeSalida** son fijos: su descripción es "SALIR" y el caracter asociado para su elección debe ser la 'X'. En cambio, cada instancia de **OpcionAccesoAMenu** necesita que se le pasen la descripción, el caracter de selección y el menú que recibe para usar cuando la opción se ejecute.

El método **ejecutar()** de **Menu** muestra todas sus opciones, más su opción de salida. Por último, se debe mostrar una leyenda que invite al usuario a ingresar una opción. Debe ignorarse cualquier ingreso inválido. La ejecución del método seguirá hasta que se elija la opción definida para salir.

### EXAMEN PARCIAL 1 - POO

**OpcionDeMenu** no implementará el método `ejecutar()`: serán sus clases derivadas las que lo hagan según lo que necesite hacer el programa.

- El método `ejecutar()` de **OpcionAccesoAMenu** simplemente debe ejecutar su menú secundario.
- El método `ejecutar()` de **OpcionDeSalida** no hace nada, debe implementarse pero por ahora quedará vacío.

**Se pide:**

- Realizar el diseño UML asociado a las clases descritas previamente.
- Desarrollar en Nassi-Shneiderman el constructor de la clase **OpcionAccesoAMenu**.
- Desarrollar en Nassi-Shneiderman el constructor de la clase **Menu**.
- Desarrollar en Nassi-Shneiderman los métodos `registrar(...)` y `ejecutar()` de **Menu** más aquellos métodos derivados de éstos que pudieran hacer falta para mostrar el menú y elegir y procesar sus acciones.

**EXAMEN PARCIAL 1 - POO**

Resolución (UML)

