

[Página Principal](#) / [Mis cursos](#) / [Instituto de Tecnología ORT](#) / [Analista de Sistemas Almagro y Belgrano](#) / [Programación](#)  
/ [Cursada del 2° Cuatrimestre 2022](#) / [Materias del 2°Cuatrimestre \(1°2°\)](#) / [TP1-2022-2](#) / [Evaluaciones Generales](#) / [1°Parcial Ambas Sedes](#)

## Pregunta 1

Sin responder aún

Puntúa como 10,00

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 definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reusable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación de miembros de instancia y de clase.
- Correcta aplicación de herencia y polimorfismo, incluyendo interfaces.
- Correcta aplicación conceptual de las relaciones entre clases.
- Implementación de POO en Java desde Eclipse.
- Correcta importación y exportación de proyectos desde Eclipse.

Antes de comenzar descargá el [proyecto adjunto](#), importalo desde Eclipse y renombralo con el siguiente formato:

**TP1-PAR1-SEDE-CURSO-APELLIDO-Nombre**

Ejemplos:

**TP1-PAR1-BE-11A-GONZALEZ-Benito**

**TP1-PAR1-YA-11A-HARA-Jaime**

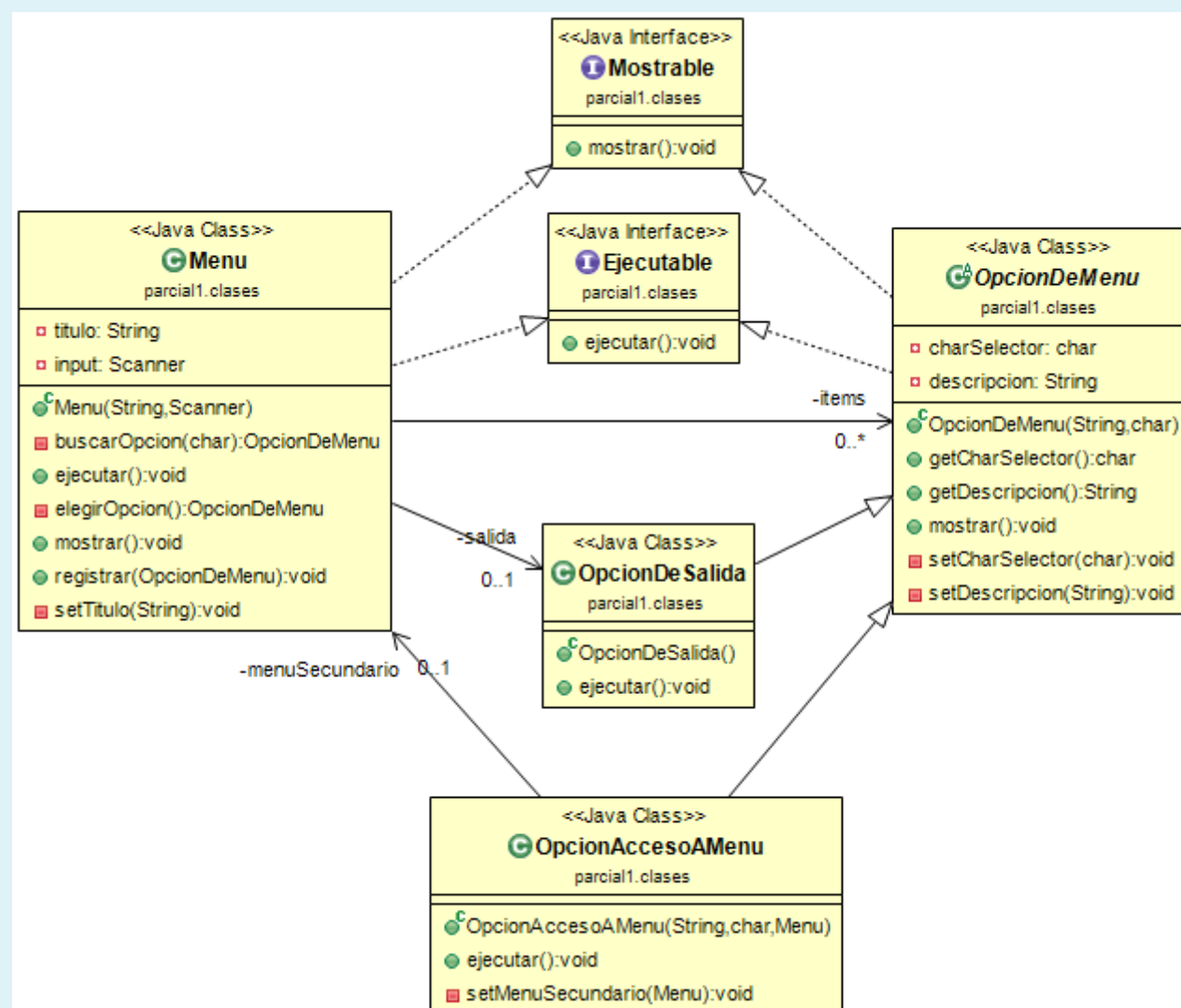
**Este proyecto deberá ser exportado al entregar con este nombre. Tené en cuenta que esto también forma parte del examen y se califica.**

#### NOTA IMPORTANTE:

- No te quedes con dudas, preguntá lo que haga falta.
- Al entregar el examen asegurate de finalizar la entrega. y confirmá la recepción con tu profesor.
- No se aceptarán exámenes por fuera de la página de entrega oficial de la plataforma.

## Enunciado

Una empresa de software que se dedica a crear componentes gráficos para el desarrollo de aplicaciones nos había pedido desarrollar el “corazón” del sistema de menús que utilizará. Este menú está implementado para consola y cuenta con una clase para el menú y varias clases distintas que heredan de una clase abstracta común para implementar varios tipos de opciones de menú. Aceptaron nuestro prototipo (cuyo diseño se puede ver aquí debajo) pero nos pidieron ciertos cambios.



## Nos piden

- Escribir la interfaz **Activable**, que tiene declarado los métodos `activar()`, `desactivar()` y `estaActivada():boolean`.
- Crear la clase **OpcionOnOff** que extiende **OpcionDeMenu** e implementa **Activable**.
  - Esta clase tiene dos constructores: El primero recibe como parámetros la descripción y el caracter de selección. El segundo, además, recibe un booleano que indica el estado inicial de la acción (si está activada o no). Por defecto, cuando no se recibe, el valor de activada debe ser falso.

- valor de activada debe ser *true*.
- El método `mostrar()` de **OpcionOnOff** debe escribir, antes de la descripción de la opción:
    - “[x]”, cuando la opción está activa;
    - “[ ]” (con un espacio en blanco en vez de la ‘x’) si no está activa.
  - El método `ejecutar()` de **OpcionOnOff** invierte el estado de activación (pasa de activa a inactiva y viceversa).
  - Implementar la clase **MenuDeTareas** a partir de la clase **Menu**. Los cambios respecto a esta segunda clase son:
    - Su método `registrar()` solamente recibe objetos de la clase **OpcionOnOff**.
    - Implementa el método `getTotalTareasCumplidas():int` que devuelve la cantidad de opciones del menú que están marcadas como activas.
  - Implementar la clase **OpcionTareasCumplidas** que extiende **OpcionDeMenu**.
    - Además de la descripción y el caracter de selección recibe un **MenuDeTareas**.
    - Su método `ejecutar()` mostrará la cantidad de opciones marcadas como cumplidas (activadas) en la instancia de **MenuDeTareas** recibida en el constructor.
  - Descomentar las líneas comentadas del main y probar.

La ejecución del programa completo, activando la primera de las opciones sin marcar mostraría lo siguiente:

```
Menu Principal
L - Listar Tareas
C - Contar Tareas Cumplidas
X - Salir
Ingrese la opcion deseada
c
Cantidad de tareas cumplidas: 0
Menu Principal
L - Listar Tareas
C - Contar Tareas Cumplidas
X - Salir
Ingrese la opcion deseada
l
Lista De Tareas
[ ] I - Escribir Interfaz Activable
[ ] O - Desarrollar Clase OpcionOnOff
[ ] C - Desarrollar Clase OpcionTareasCumplidas
[ ] M - Extender MenuDeTareas desde Menu
[ ] T - Testear Programa
[ ] E - Exportar Proyecto
[ ] U - Entregar Proyecto Exportado
X - Salir
Ingrese la opcion deseada
I
Lista De Tareas
[x] I - Escribir Interfaz Activable
[ ] O - Desarrollar Clase OpcionOnOff
[ ] C - Desarrollar Clase OpcionTareasCumplidas
[ ] M - Extender MenuDeTareas desde Menu
[ ] T - Testear Programa
[ ] E - Exportar Proyecto
[ ] U - Entregar Proyecto Exportado
X - Salir
Ingrese la opcion deseada
x
Menu Principal
L - Listar Tareas
C - Contar Tareas Cumplidas
X - Salir
Ingrese la opcion deseada
c
Cantidad de tareas cumplidas: 1
Menu Principal
L - Listar Tareas
C - Contar Tareas Cumplidas
X - Salir
Ingrese la opcion deseada
x
Fin del programa
```

**Cuando termines y/o antes de que expire el tiempo del examen exportá el proyecto tal como se pide al principio. Luego subí el archivo generado al Aula Virtual como entrega del examen. Asegurate de haber adjuntado lo correcto y finalizá el examen**

Tamaño máximo de archivo: 100MB, número máximo de archivos: 1



Ir a... [2022 1C - TP1 - Simulacro 2 Examen](#) - [Formato Seguro SEB - Sala Curso Individual](#) ►