

## Práctica de laboratorio # 4

Puede desarrollar esta práctica en parejas o tríos. Procure trabajar con alguien diferente cada vez.

En este laboratorio el o la estudiante implementará el uso de herencia simple y polimorfismo para implementar una solución flexible, consistente y escalable, utilizando para ello el lenguaje C++.

### Problema único. Examen automático.

Suponga que usted trabaja para una institución educativa. Se ha determinado que a través del tiempo las preguntas se repiten en las evaluaciones, pero las y los profesores invierten demasiado tiempo año a año en la elaboración de exámenes. Entonces, se les ha contratado para implementar un sistema que genere exámenes automáticamente.



El sistema primero cuenta con un banco de preguntas. Entonces, **al generarse un examen nuevo**, solo se necesita la **cantidad de preguntas** que se quiere hacer, y la **cantidad de puntos totales que se requieren como mínimo**. El programa se encargará de tomar preguntas aleatorias de ese banco de preguntas, verificando que se cumplan ambas condiciones.

Las **preguntas de examen comparten ciertas características** que usted deberá determinar (como un enunciado y un puntaje asignado, entre otras), pero debe tomar en cuenta que estas pueden ser de varios tipos:

- **Selección única:** se dan 4 posibles respuestas y solo una es correcta.
- **Selección múltiple:** se dan 4 posibles respuestas, y puede haber una o más correctas.
- **Complete:** se da una oración incompleta y la persona debe contestar la palabra o frase que la termina.
- **Verdadero o falso:** se da una afirmación y la persona contesta si es cierta o no.
- [Agregue un nuevo tipo de pregunta que ustedes quieran implementar]

Haga un programa que **genera de forma automática un examen** con los parámetros especificados, y vaya mostrando las preguntas una a una para recibir las **respuestas**. Al final, deberá mostrar la **calificación** sobre 100.

Deberá implementar primero una jerarquía de clases que considere que todos esos tipos de pregunta siguen siendo especificaciones de pregunta, y que al final **TODAS-SON-PREGUNTAS**. Además, debe garantizarse que, si en un futuro se desean agregar más tipos de preguntas, estas cumplan los **requisitos mínimos** establecidos de su programa.

Sería bueno que las preguntas se carguen de un archivo de texto, pero no será implementado obligatoriamente para este laboratorio. Puede simplemente generar el banco de preguntas de forma directa en el código.

Haga un main que permita probar su programa. Si tiene supuestos, haga un README y adjúntelo.

Rubro	Peso
Uso de encapsulamiento POO – constructores, destructores, miembros de clase	10%
Definición de clase base virtual	15%
Definición de clases derivadas concretas	35%
Construcción de examen según parámetros	20%
Interacción y evaluación de examen	20%