

Nombre: _____ **Matrícula:** _____

Sección A

1. Dibuje un **diagrama de casos de uso** que modele el sistema que se presenta a continuación. Refine su caso de uso para mostrar al menos una relación <<extend>> y al menos una <<include>>. **[10%]**

John es un jardinero que gana dinero haciendo trabajos de jardinería para sus vecinos. John compra los materiales (fertilizante, tierra de sembrado, etc.) que necesita y cobra a sus clientes tanto por mano de obra como por materiales. Algunos clientes programan citas cuando quieren que se realice el trabajo y pagan cuando John lo completa. Algunos clientes establecen un horario (generalmente semanal) para el trabajo regular y pagan cuando John les envía una factura mensual por el trabajo realizado durante el mes.

2. Escriba el mínimo **código fuente** en Java para ilustrar el contraste entre una relación de agregación y otra de composición. **[10%]**
3. Considere la siguiente descripción simplificada de una universidad donde los profesores imparten módulos en los que estudiantes pueden registrarse.

Un profesor tiene un nombre, dirección, número de teléfono, dirección de correo electrónico y salario. Un estudiante también tiene un nombre, etc., pero no tiene salario. Un estudiante, sin embargo, tiene una nota promedio (de las notas finales de sus módulos). Un módulo tiene un nombre y un número. Cuando un estudiante está registrado en un módulo, se asientan las notas de esta inscripción; y se pueden obtener tanto el promedio actual como la nota final (si la hay). De un estudiante se puede obtener la lista de módulos en los que esté inscrito. Los profesores imparten módulos. Cada módulo tiene al menos uno y como máximo tres profesores. Hay dos tipos de módulos: pregrado y postgrado. De un módulo de pregrado los estudiantes no pueden retirarse. De uno de postgrado sí.

Dibuje un **diagrama de clases** para esta universidad. Agregue atributos y métodos cuando sea necesario. No es necesario incluir *getters* y *setters* de atributos. Modificadores de visibilidad (público, privado, etc.) no son requeridos. Declare cualquier asociación que considere necesaria. **[20%]**

4. Considere el caso de uso de **retiro de dinero** de una cuenta bancaria desde un cajero automático.

| Acción del actor | Respuesta del sistema |
|--|---|
| 1. Introducir tarjeta en el cajero | 2. Requerir clave |
| 3. Ingresar la clave | 4. Chequear que la cuenta sea válida y activa |
| | 5. Desplegar opciones |
| 6. Seleccionar la opción para retiro | 7. Desplegar cuentas asociadas a la tarjeta |
| 8. Escoger la cuenta desde la cual retirar | 9. Solicitar el monto a ser retirado |
| 10. Ingresar el monto a ser retirado | 11. Entregar el dinero |
| | 12. Preguntar si se requiere recibo impreso |
| 13. Ingresar decisión | 14. Entregar el dinero |
| | 15. Devolver la tarjeta al cliente |
| 16. Remover tarjeta del cajero | 17. Imprimir recibo (si se requiere) |
| | 18. Actualizar balance de la cuenta |

- a. Documente este caso de uso utilizando una plantilla con al menos seis elementos. **[06%]**
- b. Describa dos escenarios de este caso de uso. **[10%]**
- c. Elabore un diagrama de secuencia para el caso de uso. **[10%]**

Sección B

5. El servicio de transporte urbano de la ciudad requiere el diseño de una aplicación móvil para uso de la ciudadanía en general. La aplicación mostrará las paradas de buses en un mapa, las cuales son compartidas entre varias líneas de transporte. Los buses enviarán mensajes, en tiempo real, a la central de control del servicio de transporte. Los mensajes pueden ser simples o compuestos de varios mensajes simples. Entre los mensajes simples tenemos: (i) reporte de ubicación; (ii) desperfecto electromecánico; ó (iii) alerta por presión de neumáticos. Los mensajes serán enviados por las unidades de transporte en formato XML, pero el sistema central solo trabaja con datos en formato JSON. El servicio de transporte ha firmado un convenio con varias compañías que prestan servicios de remolque en los diferentes distritos de la ciudad. Cada compañía puede suscribirse para tomar conocimiento de emergencias y actuar según su capacidad de respuesta en ese momento.
- a. Identifique los patrones de diseño que pueda aplicar para implementar lo solicitado. Justifique su respuesta. **[09%]**
- b. Elabore un diagrama de clases aplicando los patrones de diseño que considere apropiados. Identifique herencias, multiplicidades, visibilidad de atributos y métodos. Indique, por medio de notación UML, si las entidades corresponden a interfaces, clases abstractas o concretas. Separe los patrones usados en paquetes. **[15%]**

Sección C

6. Considere el siguiente código:

```
1 class MathHelper{
2     public static boolean esPrimo(int numero) {
3         if (numero <= 1) {
4             return false;
5         }
6         if (numero <= 3) {
7             return true;
8         }
9         if (numero % 2 == 0 || numero % 3 == 0) {
10            return false;
11        }
12        for (int i = 5; i * i <= numero; i += 6) {
13            if (numero % i == 0 || numero % (i + 2) == 0) {
14                return false;
15            }
16        }
17        return true;
18    }
19 }
```

Para esta clase realice lo siguiente:

- a. Identifique 5 casos de prueba para esta clase. Para cada caso agregue una descripción de lo que se desea probar, los valores de entrada y los resultados esperados. Utilice una forma sencilla de especificación de caso de prueba; por ejemplo, una tabla con los siguientes encabezados: **[05%]**

| ID Caso de Prueba | Valores de Entrada | Resultado Esperado | Descripción |
|-------------------|--------------------|--------------------|-------------|
|-------------------|--------------------|--------------------|-------------|

- b. Implemente las 5 pruebas unitarias correspondientes. **[05%]**