UNIVERSIDAD DE ALCALÁ, GRADO DE INGENIERÍ INFORMÁTICA

ASIGNATURA : INGENIERÍA DEL SOFTWARE

PRÁCTICA DE LABORATORIO: PECL2 (2025)

Ejercicio 1

Implementar un conjunto de clases Java que representen entidades de una **agenda digital inteligente**. Las clases deben permitir gestionar:

- a) Contactos personales (nombre, teléfono, correo electrónico, dirección postal),
- b) Citas (fecha, hora, lugar, personas involucradas),
- c) Tareas pendientes (descripción, prioridad, estado).

A partir de estas clases base, se debe implementar mediante **herencia** un segundo conjunto de clases que represente:

- a) Contactos profesionales (con empresa y cargo),
- b) Citas de trabajo (incluyendo agenda de temas y duración estimada),
- c) Tareas con seguimiento (con fecha límite y responsables asignados).

Cada clase debe contener **métodos que gestionen y validen sus datos**, por ejemplo: detección de citas en conflicto, comprobación de correos válidos, ordenación por prioridad, etc.

Todas las clases deben incorporar **pruebas unitarias con JUnit** que verifiquen la lógica del sistema.

Se usará **NetBeans** o un IDE Java equivalente con soporte para JUnit.

Ejercicio 2

Calcular la complejidad ciclomática y determinar los caminos básicos del siguiente fragmento de código Java (proporcionado al final del documento).

Ejercicio 3

Aplicar **Ingeniería Inversa** para documentar el diseño de alto nivel del código del apéndice. Después, realizar pruebas de **caja blanca** (centradas en el control de flujo del código) y **caja negra** (basadas en la entrada y salida del sistema).

DOCUMENTACIÓN A ENTREGAR

Un archivo comprimido `.zip` con los siguientes elementos:

Ejercicio 1

- Documento PDF con:
  - a) Código fuente de las clases implementadas.
  - b) Código y resultados de las pruebas unitarias con JUnit.
  - c) Capturas de pantalla del entorno de desarrollo (opcional).
  - d) Proyecto completo del IDE (carpeta del proyecto).

Ejercicio 2

- Documento PDF con:
  - a) Cálculo de complejidad ciclomática.
  - b) Caminos básicos del código.

Ejercicio 3

- Documento PDF con:
  - a) Diseño de alto nivel mediante ingeniería inversa.
  - b) Resultados y justificación de pruebas de caja blanca.
  - c) Resultados y justificación de pruebas de caja negra.

Entrega

- Archivo `.zip` con nombre formado por los apellidos del alumno (Ejemplo: `LopezGonzalez.zip`).
- Cada documento PDF debe seguir este nombre seguido del número de ejercicio (Ejemplo: `LopezGonzalez_Ejercicio1.pdf`).
- No se admitirán modificaciones tras la entrega.

CRITERIOS DE VALORACIÓN

Se valorará:

- Claridad y estructura de los documentos.
- Precisión y corrección del código.
- Coherencia y justificación de las pruebas.
- Aplicación de principios de diseño y reutilización.

APÉNDICE – Fragmento de Código para los Ejercicios 2 y 3

```
public int gestionInventario(int stock, int pedidos, boolean urgente,
boolean proveedorActivo) {
    int resultado = 0;

    if (stock < 10 && pedidos > 0) {
        resultado += pedidos;
        if (urgente) {
            resultado += 5; // margen de urgencia
        }
    } else if (stock >= 10) {
        resultado -= 2; // pequeña penalización por sobrealmacenaje
    }

    if (!proveedorActivo) {
        resultado = -1; // error en la gestión
    }

    return resultado;
}
```

Universidad de Alcalá - Curso 2025