1. **[2p]** Dado el siguiente código Java,

   a) Dibujar un diagrama de clases que muestre todas las clases/interfaces involucradas (salvo `Object`) y sus relaciones.

   b) ¿Hay comportamiento polimórfico en el código? Indicar dónde y explicar brevemente.

LectorCSV.java

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.util.List;
import java.util.Arrays;

public class LectorCSV {
  private BufferedReader entrada;

  public LectorCSV(BufferedReader entrada) {
    this.entrada = entrada;
  }

  public List<String> lerFila() throws IOException {
    String linea = entrada.readLine();
    if (linea == null) {
      return null;
    }
    String[] valores = linea.split(",");
    return Arrays.asList(valores);
  }
}
```

Main.java

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.List;

public class Main {
  public static void main(String[] args) throws
  IOException {
    BufferedReader br = new BufferedReader(new
    FileReader("datos.csv"));
    LectorCSV lector = new LectorCSV(br);

    List<String> fila;
    while ((fila = lector.lerFila()) != null) {
      System.out.println(fila);
    }

    br.close();
  }
}
```

Nota: las definiciones de las clases e interfaces usadas de la biblioteca estándar son las siguientes:

```java
public interface List<E> { ... }
public class BufferedReader extends Reader {
    public BufferedReader(Reader in) { ... }
    String readLine() { ... }
    ...
}
public class FileReader extends InputStreamReader { ... }
public class InputStreamReader extends Reader { ... }
public abstract class Reader { ... }
```

Criterios de corrección:

- En el diagrama de clases verificar principalmente que usen bien las flechas de relaciones (composición, generalización, etc). No es tan importante la diferencia entre composición y agregación. Al menos deberían figurar las clases `LectorCSV`, `Main`, `BufferedReader`, `FileReader`, `InputStreamReader` y `Reader`.
- Prestar especial atención a la relación entre `BufferedReader` y `Reader`, que es doble (herencia y composición/agregación).
- No es necesario que escriban los nombres de atributos y métodos.
- En la pregunta b) hay varias respuestas posibles. Verificar sobre todo que usen los términos correctos en la justificación. Por ejemplo, estaría mal si ponen algo como "sí, porque `lerFila` es una función que devuelve una clase..." (es un **método** y devuelve una **instancia de una interfaz**).

2. **[1p]** Considerar el siguiente código. Identificar un principio de diseño que se viola, indicar cuál es y justificar. Proponer una solución para evitar la violación del principio (no es necesario escribir el código).

```java
public class Factura {
  private String empresa;
  private List<Item> items;

  public Factura(String empresa, List<Item> items) {
    this.empresa = empresa;
    this.items = items;
  }

  public void agregarItem(Item item) {
    items.add(item);
  }

  public void exportar(String ruta, String formato) {
    OutputStream archivo = new FileOutputStream(ruta);
    if (formato.equals("PDF")) {
      exportarPDF(archivo);
    } else if (formato.equals("HTML")) {
      exportarHTML(archivo);
    } else if (formato.equals("TXT")) {
      exportarTXT(archivo);
    }
  }

  private void exportarPDF(OutputStream archivo)
  { ... }
  private void exportarHTML(OutputStream archivo)
  { ... }
  private void exportarTXT(OutputStream archivo)
  { ... }
}
```

Criterios de corrección:

- Verificar que identifiquen correctamente el principio de diseño violado (OCP o SRP pero si justifican bien puede ser otro).
- De nuevo, verificar que usen los términos correctos en la justificación y en la solución propuesta.

3. **[2p]** Dada la siguiente base de datos Prolog, dibujar el árbol de búsqueda e indicar el resultado de la consulta:

   ?- `lo_quiero(X).`

```
es_nuevo(iphone_13).
es_nuevo(galaxy_s21).
es_potente(iphone_13).
es_potente(pixel_6).
es_barato(galaxy_s21).
es_barato(pixel_6).

lo_quiero(X) :- es_nuevo(X), es_potente(X), es_barato(X).
```

Criterios de corrección:

- Arbol de busqueda:

```
?- lo_quiero(X).
    |
    +-[X = X]- ?- es_nuevo(X), es_potente(X), es_barato(X).
               |
               +-[X = iphone_13]- ?- es_potente(iphone_13), es_barato(iphone_13).
               |                  |
               |                  +-- ?- es_barato(iphone_13). -- FALLA
               |
               +-[X = galaxy_s21]- ?- es_potente(galaxy_s21), es_barato(galaxy_s21). -- FALLA
```

- El resultado de la consulta es que no hay soluciones (falla).

4. **[2p]** Dado el siguiente código Clojure,
   a) ¿Cuál o cuáles funciones (con nombre o anónimas) son puras/impuras? Justificar.
   b) ¿Cuál o cuáles funciones (con nombre o anónimas) son de orden superior? Justificar.
   c) ¿La salida del programa es deterministica? Justificar.

```
1  (defn con-registro [f]
2    (let [log (atom [])]
3      [log
4       (fn [x]
5         (swap! log conj x)
6         (f x))]))
7
```

```
8  (let [[log cuad] (con-registro
9                    (fn [x]
10                     (* x x)))
11       a (future (cuad 10))
12       b (future (cuad 20))
13       resultado (+ @a @b)]
14   (println "El resultado es:" resultado)
15   (println "La función fue invocada" (count @log)
     "veces.")
16   (println "Log:" @log))
```

Criterios de corrección:

- `con-registro` es **pura** y de **orden superior** (recibe una función, devuelve una función)
- La `fn [x]` en la línea 4 es **impura** (invoca a swap! que es impura)
- La `fn [x]` en la línea 9 es **pura** (solo hace una multiplicación)
- La salida del programa **no es determinística** (el orden de ejecución de los futures no está garantizado, por lo que el contenido de `log` puede variar entre ejecuciones).

5. **[2p]** Dada la siguiente expresión lambda,

   a) Identificar las variables libres y ligadas.

   b) Mostrar los pasos de reducción usando orden normal (β-redex más externa desde la izquierda) hasta llegar a la forma normal (si existe). Aplicar reducciones α según sea necesario para evitar conflictos de nombres.

   `(λx f.f x) (λq.f) (λa.a a)`

   Criterios de corrección: https://projectultimatum.org/cgi-bin/lambda?t=(%CE%BBx.%CE%BBf.f%20x)%20(%CE%BBq.f)%20(%CE%BBa.a%20a)&r=&m=normal%20order

   - No es necesario que dibujen el árbol, solo identificar las variables libres y ligadas.
   - El resultado en forma normal es `f`.

6. **[1p]** Dado el siguiente código JavaFX:

   a) Dibujar el árbol correspondiente al Scene Graph.

   b) Explicar brevemente el funcionamiento del programa. ¿Qué sucede cuando el usuario interactúa con la interfaz? ¿Cómo se logra este comportamiento?

```java
public class App extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Label etiqueta = new Label("Ingrese un color en formato #RRGGBB:");
        TextField color = new TextField("#ff0000");

        Rectangle rectangulo = new Rectangle(300, 300);
        rectangulo.fillProperty().bind(Bindings.createObjectBinding(
                () -> {
                    try {
                        return Color.web(color.getText());
                    } catch (IllegalArgumentException e) {
                        return Color.RED;
                    }
                },
                color.textProperty()
        ));

        stage.setScene(new Scene(new VBox(
                new HBox(etiqueta, color),
                rectangulo
        )));
        stage.show();
    }
}
```

Criterios de corrección:

```
Stage
  |
  +-- Scene
        |
        +-- VBox
              |
              +-- HBox
              |     |
              |     +-- Label
              |     +-- TextField
              |
              +-- Rectangle
```

Verificar que usen los términos correctos en la explicación:

- Cuando el usuario ingresa un color en el TextField, el rectángulo cambia su color de relleno.
- Esto se logra mediante un binding entre la propiedad `text` del TextField y la propiedad `fill` del Rectangle, que automáticamente invoca la función anónima cuando el usuario cambia el texto.
- Si el texto ingresado no es un color válido, el rectángulo se rellena con rojo por defecto.