



Proyecto 2: Programación Orientada a Objetos

Usando Java desarrollar alguno de los siguientes proyectos, el proyecto puede ser realizado en equipo de hasta cuatro participantes. Para cada proyecto se requiere lo siguiente.

1. Hacer un diagrama de clases en UML que represente al sistema
2. Implementar el diagrama UML usando Java.
3. Las clases de java deberán incluir comentarios usando la notación de Java con la descripción de los métodos y secciones de código. Se deberá generar la documentación del proyecto
4. El programa a desarrollar deberá incluir herencia, polimorfismo, abstracción, encapsulamiento, además del uso de paquetes .
5. Desarrollar un reporte en PDF que incluya
 - el diagrama de clases y una explicación del mismo.
 - pruebas realizadas sobre la funcionalidad del código
6. Adjuntar empaquetado el código fuente y script de base de datos (si se requiere)
7. Adjuntar la documentación generada del proyecto.
8. Realizar una demostración del funcionamiento del proyecto en clase

Tema 1: Desarrollo de un sistema de emparejamiento de personas

Descripción: Un programa en Java que funcione como un sistema de citas básico, donde los usuarios ingresen sus características personales (intereses, edad, gustos) y el sistema les sugiera parejas compatibles basándose en criterios de afinidad.

Características Principales

1. **Interfaz Gráfica (Swing/JavaFX)**
 - a) Registro de Usuarios:
 - Formulario para ingresar: nombre, edad, género, ubicación, hobbies, etc.
 - Opción de subir una foto (opcional, usando **JFileChooser**).
 - b) Perfil de Usuario:
 - Muestra la información del usuario y sus matches.
 - c) Sistema de Matching:
 - Pantalla con posibles parejas compatibles (mostradas como "tarjetas" estilo Tinder).
 - Botones de "Me gusta" ♥ o "No me gusta" ✕.
2. **Lógica de Emparejamiento**
 - a) Algoritmo de Compatibilidad:
 - Calcula afinidad basada en:
 - Edad (rango compatible).
 - Intereses comunes (ej: música, deportes, etc.).
 - Ubicación (priorizar personas cercanas).
 - b) Lista de Matches:
 - Muestra los usuarios con mayor puntaje de compatibilidad.
3. **Persistencia:**
 - Guardar usuarios, hobbies y matches en SQLite/MySQL.

Tema 2: Editor Gráfico Simple en Java

Descripción: Un programa que permita dibujar figuras geométricas básicas (líneas, rectángulos, círculos, etc.) en un canvas, moverlas con el mouse y guardar/cargar los dibujos en una base de datos o archivo.

Características Principales

1. **Interfaz Gráfica (Swing/JavaFX):**
 - a) *Canvas* (área de dibujo) donde se plasmarán las figuras.



- Barra de herramientas con botones para seleccionar figuras (línea, rectángulo, círculo, etc.).
 - Selector de color.
 - Botones para guardar y cargar dibujos.
2. **Requerimientos funcionales**
 - a) Dibujar figuras arrastrando el mouse (ej: click + arrastrar para definir tamaño de un rectángulo).
 - b) Seleccionar figuras haciendo click sobre ellas y moverlas (drag and drop).
 - c) Cambiar color de relleno o borde (opcional avanzado).
 3. **Persistencia (Guardado/Carga):**
 - a) Guardar el dibujo actual en una base de datos (SQLite, MySQL) o en un archivo serializado (.bin, .json).
 - b) Cargar dibujos previos desde la base de datos/archivo.

Tema 3: Juego "Atrapa los Objetos"

Descripción: Un juego donde el jugador controla un personaje (ej: una canasta o un personaje 2D) que debe atrapar objetos que caen desde la parte superior de la pantalla. Cada objeto da puntos diferentes, y algunos restan vida.

Características Principales

1. **Interfaz Gráfica:**
 - a) Canvas: Área de juego donde caen los objetos y se mueve el jugador.
 - b) Controles: Teclas (← →) o mouse para mover al personaje.
 - c) HUD: Muestra puntuación, vidas y nivel actual.
 - d) Menú: Botones para iniciar/pausar/reiniciar el juego.
2. **Mecánicas del Juego:**
 - a) Objetos:
 - b) Beneficiosos: +10 puntos.
 - c) Dañinos: -1 vida.
 - d) Especiales: +50 puntos o power-up (ej: velocidad temporal).
 - e) Niveles: La velocidad de caída aumenta cada cierto tiempo.
 - f) Game Over: Si el jugador pierde todas las vidas.
3. **Persistencia de Datos:**
 - a) Guardar el récord de puntuación en una base de datos (SQLite) o archivo de texto.
 - b) Mostrar un "Top 10" de mejores jugadores.

Objeto a calificar	Porcentaje esperado	Porcentaje alcanzado
Reporte	20%	
Codigo fuente	30%	
Documentación	10%	
Demostración	40%	
Totales	100%	