

**FACULTAD DE INGENIERÍAS Y ARQUITECTURA**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**MISCELANIA DE EJERCICIOS DE RECUPERACIÓN PARA EL PROYECTO DE POO**

|  |  |  |  |
| --- | --- | --- | --- |
| **CARRERA** |  |  | **CICLO** |
| Ciencias de la computación |  |  | **2** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Autores:**

Pedro Daniel Irene Robalino

TABLA DE CONTENIDO

[PROPUESTAS PARA EL PROYECTO 3](#_Toc204671023)

[1. TEMA DE PROYECTOS NRO. 1.: Sistema Integral de Gestión de Talleres Artísticos Comunitarios – Loja 2025 3](#_Toc204671024)

[2. TEMA DE PROYECTOS NRO. 2.: Sistema de Gestión de Proyectos de Reforestación Comunitaria - Loja Verde 2025. 4](#_Toc204671025)

[3. TEMA DE PROYECTOS NRO. 3.: Aplicación de Seguimiento de Actividades Deportivas Universitarias UTPL. 5](#_Toc204671026)

[4. TEMA DE PROYECTOS NRO. 4.: Sistema de Reservas y Control de Visitas a Museos y Centros Culturales de Loja. 6](#_Toc204671027)

[5. TEMA DE PROYECTOS NRO. 5.: Plataforma de Tutorías Académicas Personalizadas para Estudiantes de Primer Ciclo. 7](#_Toc204671028)

[6. TEMA DE PROYECTOS NRO. 6.: Simulador de Comercio Local y Ferias Artesanales en los Barrios Lojanos. 8](#_Toc204671029)

[7. TEMA DE PROYECTOS NRO. 7.: Sistema de Monitoreo de Biodiversidad en el Parque Nacional Podocarpus de Loja. 9](#_Toc204671030)

## **PROPUESTAS PARA EL PROYECTO**

## **TEMA DE PROYECTOS NRO. 1.: Sistema Integral de Gestión de Talleres Artísticos Comunitarios – Loja 2025**

El Ministerio de Cultura y Patrimonio, en coordinación con el Gobierno Autónomo Descentralizado de Loja, impulsa el desarrollo de talleres artísticos comunitarios permanentes como parte de su programa cultural para el año 2025. Estos talleres se enfocan en diversas áreas como danza, música, pintura, escultura, teatro y escritura creativa, y están dirigidos a distintos grupos etarios y sociales. Cada taller está dirigido por un facilitador certificado y cuenta con una planificación de sesiones, materiales requeridos, número máximo de participantes y nivel de dificultad.

Los participantes se registran en los talleres y pueden acceder a recursos pedagógicos, horarios, evaluaciones de progreso y actividades complementarias. Algunas actividades requieren evaluaciones prácticas con criterios definidos y puntajes ponderados. El sistema debe permitir además la gestión de inscripciones, seguimiento de participantes, asignación de facilitadores, programación de sesiones y generación de reportes de impacto cultural y social.

Para efectos de control institucional y de transparencia, toda la información debe persistirse en una base de datos relacional y además permitir la carga inicial de datos mediante archivos CSV. El sistema debe respetar una arquitectura basada en el patrón arquitectónico MVC, permitir consultas por nombre de participante, tipo de taller, nivel de avance y generar reportes organizados en consola. Se espera que el sistema aplique correctamente los pilares fundamentales de la Programación Orientada a Objetos: abstracción, encapsulamiento, herencia y polimorfismo, así como el uso adecuado de arreglos dinámicos, manejo de excepciones y principios SOLID para garantizar su escalabilidad.

**Requerimientos funcionales**:

1. Permitir el registro y gestión de múltiples talleres artísticos, considerando tipo de taller, duración, materiales necesarios y nivel de dificultad.
2. Registrar participantes y su asignación a talleres disponibles, considerando restricciones de edad y cupo.
3. Asociar facilitadores a uno o varios talleres, con la posibilidad de evaluación periódica de su desempeño.
4. Registrar las sesiones planificadas y realizar el seguimiento de la asistencia y avance de los participantes.
5. Implementar evaluaciones prácticas con criterios y puntajes por taller, incluyendo cálculo de promedios y asignación de niveles de progreso.
6. Importar información inicial desde archivos CSV para talleres, facilitadores y materiales.
7. Almacenar todos los datos en una base de datos relacional.
8. Aplicar excepciones para manejo de errores en el registro de datos, asignaciones, cargas de archivos y consultas.
9. Mostrar reportes finales por consola que incluyan: listado de participantes por taller, talleres por facilitador, evaluaciones realizadas y ranking de avance.
10. Seguir la arquitectura MVC y estructurar el sistema usando herencia, polimorfismo, encapsulamiento y abstracción.
11. Aplicar relaciones de composición y agregación entre clases cuando sea necesario.
12. Diseñar el UML detallado, incluyendo nombres de clases, atributos, métodos, constructores, relaciones, cardinalidades, y tipos de clases.

## **TEMA DE PROYECTOS NRO. 2.: Sistema de Gestión de Proyectos de Reforestación Comunitaria - Loja Verde 2025.**

En el marco de la iniciativa “**Loja Verde 2025**”, impulsada por gobiernos locales y colectivos ambientales, se lanza una campaña integral de reforestación en sectores urbanos, rurales y periurbanos de Loja. Se desea implementar un sistema de información para planificar, registrar y monitorear los distintos proyectos de reforestación comunitaria organizados por parroquias y barrios.

Cada proyecto de reforestación tiene un objetivo ecológico, una meta de árboles a sembrar y una duración definida. Existen varios tipos de proyectos según el ecosistema de intervención: urbano, ribereño, rural agrícola o forestal protegido. Los responsables comunitarios lideran estos proyectos y coordinan equipos de voluntarios que participan en actividades de siembra, monitoreo y cuidado. **Nota**: cada tipo de proyecto tiene sus datos diferenciadores.

Cada voluntario puede participar en varios proyectos, registrando el número de árboles sembrados por jornada. El sistema debe calcular el total de árboles sembrados por proyecto, por voluntario y el cumplimiento porcentual de la meta ecológica. Asimismo, debe permitir generar informes por tipo de proyecto, parroquia, o grado de éxito.

Adicionalmente, el sistema debe permitir importar datos de registros voluntarios desde archivos CSV enviados por los coordinadores barriales, guardar toda la información en una base de datos relaciones sql, y presentar reportes detallados en consola, manejando cualquier tipo de inconsistencia o error en los datos importados.

**Requerimientos funcionales**

1. Registrar múltiples proyectos de reforestación con metas, tipo, duración y responsable comunitario.
2. Gestionar equipos de voluntarios, con la posibilidad de participar en múltiples proyectos.
3. Registrar jornadas de siembra y árboles sembrados por voluntario.
4. Calcular el avance porcentual del proyecto y el promedio de árboles sembrados por tipo de proyecto dado sus atributos diferenciadores (urbano: áreas verdes, ribereño: número de ríos, rural agrícola: hectáreas y propietario, o forestal protegido: hectáreas).
5. Permitir la importación de datos desde archivos CSV estructurados por jornada y voluntario.
6. Implementar estructuras dinámicas de objetos para representar proyectos, voluntarios y jornadas.
7. Persistir todos los datos en una base de datos.
8. Aplicar el patrón MVC para separar la lógica del sistema, la interfaz de usuario (consola) y el acceso a datos.
9. Gestionar errores como duplicidad de registros, datos inválidos o errores de carga.
10. Generar informes detallados por parroquia, responsable, grado de avance y tipo de proyecto.

## **TEMA DE PROYECTOS NRO. 3.: Aplicación de Seguimiento de Actividades Deportivas Universitarias UTPL.**

La Universidad Técnica Particular de Loja desea implementar una solución informática para registrar, gestionar y analizar las actividades deportivas universitarias desarrolladas por los estudiantes de pregrado en sus diferentes ciclos académicos. Las actividades pueden corresponder a disciplinas como fútbol, atletismo, natación, entre otras, y pueden organizarse en eventos como torneos, campeonatos internos, entrenamientos o jornadas de preparación física.

El sistema debe permitir registrar participantes, los entrenadores responsables y detalles específicos de cada actividad. Se desea que el sistema también sea capaz de generar reportes de rendimiento físico por tipo de actividad, detectar atletas destacados, calcular puntajes promedio por competencias y llevar un seguimiento de asistencia. Las métricas de evaluación y seguimiento pueden variar en función de la actividad deportiva.

Los entrenadores podrán planificar y registrar sesiones según su área, establecer criterios personalizados de evaluación del desempeño y hacer seguimiento del progreso de cada estudiante. Además, debe preverse el uso del sistema por parte de personal administrativo, para generar informes organizativos e históricos.

Se espera que el sistema esté diseñado aplicando un modelo robusto de Programación Orientada a Objetos, incluyendo una jerarquía de entidades, relaciones claras entre clases, encapsulamiento de datos, comportamientos polimórficos y principios SOLID.

**Requerimientos funcionales**

1. Permitir registrar múltiples tipos de actividades deportivas con criterios específicos según la disciplina.
2. Almacenar y gestionar datos de estudiantes, entrenadores y administrativos vinculados a las actividades.
3. Gestionar sesiones o jornadas deportivas con detalle de fechas, duración y participantes.
4. Incluir procesos de evaluación del desempeño físico individual según la actividad practicada.
5. Calcular automáticamente puntajes o indicadores de desempeño con fórmulas variables por tipo de disciplina. Por ejemplo *(solo es sugerible, analice, adapte los modelos matemáticos según la pertinencia)*:
   * 1. Fútbol: rendimiento = (goles \* 3 + asistencias \* 2) / partidosJugados
     2. Atletismo: rendimiento = (tiempoInicial - tiempoFinal) / tiempoInicial \* 100
     3. Natación: rendimiento = metrosRecorridos / tiempo \* eficienciaFactor
6. Generar reportes por estudiante, por actividad o por evento, según filtros definidos por el usuario.
7. Incorporar estructuras dinámicas para manejar múltiples registros de actividades y participantes.
8. Cargar datos históricos desde archivos CSV y permitir la exportación de reportes.
9. Implementar persistencia en una base de datos (SQLite o MySQL).
10. Manejar excepciones en la carga de archivos, inserciones inválidas y operaciones de base de datos.
11. Aplicar el patrón MVC para la organización del sistema.
12. Aplicar principios SOLID, especialmente para separar lógica de negocio, acceso a datos y presentación.

## **TEMA DE PROYECTOS NRO. 4.: Sistema de Reservas y Control de Visitas a Museos y Centros Culturales de Loja.**

La Dirección de Cultura y Patrimonio de Loja desea implementar un sistema automatizado que permita gestionar las reservas y control de visitas de ciudadanos, turistas y grupos educativos a los principales museos y centros culturales de la ciudad. El sistema debe considerar las particularidades de cada tipo de visitante y las características únicas de cada espacio cultural.

Cada centro cultural cuenta con espacios de exhibición, actividades programadas y limitaciones de aforo. Algunos eventos requieren reserva anticipada y validación del acceso en puntos de control. Las visitas pueden ser individuales, guiadas o grupales, cada una con procesos y reglas específicas de registro y seguimiento.

El sistema debe poder procesar distintos tipos de usuarios, registrar múltiples visitas, generar reportes de asistencia por tipo de evento, e incluir un módulo que permita calcular indicadores como la ocupación promedio, cancelaciones o visitantes recurrentes. Los reportes deben adaptarse a las características del visitante y del centro visitado.

**Requerimientos funcionales**

1. Permitir registrar museos y centros culturales con su respectiva capacidad máxima, ubicación, horarios y tipos de eventos culturales asociados.
2. Gestionar distintos tipos de visitantes:
   1. Individuales (pueden reservar en línea y cancelar visitas)
   2. Educativos (grupos escolares o universitarios con guías designados)
   3. Turísticos (acceso a visitas guiadas y actividades opcionales)
3. Cada visita debe estar asociada a una fecha, un espacio cultural y un visitante.
4. Calcular la ocupación de cada espacio mediante: ocupacion = (visitasConfirmadas / aforoMaximo) \* 100
5. Calcular un indicador de repetición por visitante: frecuenciaVisita = visitasRealizadas / mesesActivos
6. Procesar las reservas automáticamente según el tipo de usuario y evento.
7. Implementar estructuras dinámicas para almacenar reservas, espacios, actividades y usuarios.
8. Aplicar sobrecarga de métodos para el cálculo de indicadores y envío de confirmaciones.
9. Persistir los datos en una base de datos relacional.
10. Leer información de visitantes o espacios culturales desde archivos CSV.
11. Manejar excepciones para operaciones como reservas inválidas, visitas fuera de horario o duplicadas.
12. Seguir el patrón arquitectónico MVC.
13. Aplicar principios SOLID en la distribución de responsabilidades entre clases y capas del sistema.

## **TEMA DE PROYECTOS NRO. 5.: Plataforma de Tutorías Académicas Personalizadas para Estudiantes de Primer Ciclo.**

La Universidad desea implementar una plataforma para gestionar tutorías académicas dirigidas a estudiantes de primer ciclo de distintas carreras. El sistema debe permitir a los tutores planificar sesiones, asignar materiales, evaluar progresos y adaptar estrategias según el tipo de estudiante. La plataforma debe también diferenciar los procesos de tutorías presenciales, virtuales y grupales, cada una con requerimientos específicos y mecanismos de registro distintos.

Los estudiantes pueden acceder al historial de sus sesiones, recomendaciones académicas, y visualizar estadísticas de progreso. Cada tutoría puede incluir recursos de apoyo, actividades asignadas y observaciones personalizadas. Los materiales de tutoría pueden incluir enlaces, documentos o tareas prácticas, y deben almacenarse y gestionarse para su reutilización futura. La plataforma debe ser capaz de cargar y guardar datos de tutorías desde archivos CSV y persistirlos en una base de datos relacional.

Se requiere que el sistema pueda calcular métricas relevantes para los tutores y los estudiantes como:

porcentajeAvance = (sesionesCumplidas / sesionesTotales) \* 100

nivelRiesgo = (sesionesCanceladas / sesionesProgramadas) \* 10 + ausencias

Cada tipo de tutoría (presencial, virtual, grupal) debe implementar métodos de evaluación y adaptación diferenciados. Además, debe haber un módulo para cargar desde CSV la planificación semestral de tutorías y generar reportes al final del ciclo. El sistema debe aplicar excepciones para control de errores como solapamiento de sesiones, recursos no encontrados o estudiantes no asignados.

**Requerimientos funcionales**

1. Registrar a los tutores, estudiantes y los tipos de tutoría a desarrollar.
2. Gestionar sesiones de tutoría con información como fecha, duración, tipo y materiales usados.
3. Registrar los diferentes tipos de tutorías (presencial, virtual, grupal) con sus respectivos métodos y procesos diferenciadores.
4. Implementar métodos de análisis polimórficos para el cálculo de avance académico y nivel de riesgo.
5. Almacenar y gestionar materiales didácticos de cada sesión de tutoría.
6. Permitir la planificación y edición de agendas de tutorías dinámicamente.
7. Implementar agregación entre tutorías y los recursos, así como composición para las sesiones programadas.
8. Aplicar encapsulamiento y abstracción en el diseño de clases y acceso a la información.
9. Utilizar arreglos dinámicos para almacenar sesiones, recursos y estudiantes.
10. Aplicar principios SOLID para separar responsabilidades y evitar dependencia de clases concretas.
11. Persistir la información en una base de datos y cargar información desde archivos CSV.
12. Manejar excepciones relacionadas con operaciones inválidas, como colisiones de horarios o datos mal formateados.
13. Mostrar resultados en consola con claridad, organización y detalles sobre avances y estadísticas.
14. Respetar el patrón arquitectónico MVC, separando la lógica de presentación, negocio y persistencia.
15. Documentar el modelado UML con detalles de clases, atributos, relaciones, cardinalidades y jerarquías.

## **TEMA DE PROYECTOS NRO. 6.: Simulador de Comercio Local y Ferias Artesanales en los Barrios Lojanos.**

Con el objetivo de promover la economía popular en los barrios de Loja, se plantea la creación de un simulador que permita gestionar y proyectar el funcionamiento de pequeños comercios y ferias artesanales. Cada punto de venta local y feria artesanal contará con características particulares como su ubicación, especialidad comercial, y días de actividad. Dependiendo del tipo de evento (feria itinerante, mercado fijo o tienda local), se deben administrar los recursos, ventas, atención a clientes, logística de productos y su rentabilidad.

El sistema debe diferenciar los procesos según el tipo de actividad económica: por ejemplo, las ferias itinerantes deben calcular la logística de transporte y la instalación temporal; los mercados fijos requieren administración de espacios y rotación de productos; y las tiendas locales mantienen inventarios permanentes con atención personalizada. Se desea estimar el rendimiento económico de cada modalidad a través de métricas tales como:

• gananciaNeta = ingresosTotales - (costosFijos + costosVariables)

• eficiencia = productosVendidos / productosDisponibles \* 100

El simulador debe permitir registrar vendedores, productos, jornadas de actividad, analizar resultados y generar reportes personalizados. Además, debe integrarse con una base de datos para guardar registros históricos y permitir cargar agendas desde archivos CSV. Es esencial aplicar el manejo adecuado de errores (como duplicación de eventos, productos inexistentes o conflictos de horario) y mostrar reportes claros y detallados en consola.

**Requerimientos Funcionales**

1. Registrar puntos de comercio con datos generales, tipo de actividad, ubicación y horario.
2. Gestionar ferias y eventos diferenciados según su tipo con comportamientos particulares.
3. Registrar productos, sus características, stock disponible y costos asociados.
4. Simular actividades comerciales en una agenda y registrar el detalle de las ventas realizadas.
5. Implementar métodos diferenciadores para calcular rentabilidad y eficiencia según el tipo de actividad.
6. Asociar recursos logísticos y materiales a los eventos o comercios según sus necesidades.
7. Modelar estructuras de datos para almacenar registros de ventas, productos y actividades en arreglos dinámicos.
8. Persistir datos en una base de datos relacional y permitir la carga de información desde archivos CSV.
9. Aplicar principios SOLID para dividir responsabilidades, evitar acoplamientos innecesarios y facilitar la extensibilidad.
10. Manejar excepciones comunes como productos agotados, datos inconsistentes o registros duplicados.
11. Utilizar una estructura MVC para separar claramente la lógica de presentación, negocio y acceso a datos.
12. Generar reportes claros y detallados de desempeño económico y logístico en la consola.
13. Documentar un diagrama de clases UML completo con atributos, métodos, constructores, relaciones y cardinalidades.

## **TEMA DE PROYECTOS NRO. 7.: Sistema de Monitoreo de Biodiversidad en el Parque Nacional Podocarpus de Loja.**

El Parque Nacional Podocarpus, reconocido por su riqueza biológica, requiere un sistema informático que facilite el monitoreo, registro y análisis de las especies de flora y fauna. Este sistema debe gestionar observaciones realizadas por investigadores, técnicos y visitantes capacitados, quienes registran avistamientos, comportamientos, condiciones climáticas y ubicaciones precisas mediante dispositivos móviles o estaciones fijas.

Cada tipo de especie presenta características distintas. Por ejemplo, las aves requieren registrar rutas migratorias y patrones de canto; los mamíferos, zonas de actividad y hábitos nocturnos; y las especies vegetales, ciclos fenológicos y condiciones del suelo. Estos procesos deben diferenciarse en el sistema. Además, cada punto de observación cuenta con sensores ambientales que, agregados a las observaciones, permiten generar alertas sobre el estado de conservación del ecosistema.

Se busca implementar un sistema que permita gestionar especies, observaciones, sensores, zonas de estudio, reportes y alertas. El diseño debe permitir la extensión de nuevos tipos de especies y sensores, así como el almacenamiento eficiente de registros históricos en una base de datos.

**Requerimientos Funcionales**

1. Registrar especies con datos científicos y clasificarlas según tipo (ave, mamífero, vegetal u otro).
2. Diferenciar los procesos de recolección de datos según el tipo de especie observada.
3. Asociar sensores ambientales a zonas específicas del parque y registrar sus mediciones periódicas.
4. Implementar un sistema de observaciones para registrar avistamientos, condiciones ambientales, fecha, hora, ubicación y observador.
5. Procesar de forma polimórfica información según el tipo de especie, incluyendo el análisis de patrones o comportamiento.
6. Integrar datos de sensores y observaciones para emitir alertas por cambios significativos en los hábitats.
7. Almacenar dinámicamente registros de observaciones, mediciones y alertas utilizando estructuras de datos no primitivas.
8. Persistir todos los datos en una base de datos relacional y permitir la carga masiva desde archivos CSV.
9. Aplicar excepciones para validar datos incompletos, ubicaciones erróneas o registros duplicados.
10. Cumplir con el patrón arquitectónico MVC para separar interfaz, lógica de negocio y acceso a datos.
11. Mostrar resultados e informes detallados de manera organizada y clara en consola.
12. Documentar un diagrama UML completo con tipos de clases, atributos, métodos, relaciones (herencia, agregación, composición), cardinalidades, constructores y sobreescritura de método, etc.