**PROYECTO FINAL**

**Análisis Sistemas I**

**1. Sistema de Gestión Académica**

**Descripción:**

Una aplicación para gestionar inscripciones, asignación de cursos, notas y horarios en una institución educativa. Ideal para colegios, universidades o academias en línea.

**Módulos Principales:**

1. **Gestión de Usuarios:** Registro y autenticación de estudiantes, docentes y administradores.
2. **Inscripción de Cursos:** Selección de materias y horarios disponibles.
3. **Asignación de Calificaciones:** Ingreso y modificación de notas por parte de los docentes.
4. **Generación de Reportes:** Boletas de calificaciones y rendimiento académico.

**Diagramas UML Recomendados:**

* **Casos de Uso:** Inscripción de estudiantes, asignación de notas, generación de reportes.
* **Clases:** Usuario, Estudiante, Docente, Curso, Inscripción, Nota.
* **Secuencia:** Proceso de inscripción y publicación de calificaciones.
* **Estados:** Ciclo de vida de una inscripción (Pendiente, Confirmada, Cancelada).

**Herramientas Sugeridas:**

* **Base de Datos:** MySQL o PostgreSQL.
* **Frontend:** React o Angular.
* **Backend:** Node.js o Java (Spring Boot).

**2. Plataforma de Comercio Electrónico**

**Descripción:**

Un sistema para gestionar ventas en línea, incluyendo productos, pagos y envíos. Puede enfocarse en **productos físicos o digitales**.

**Módulos Principales:**

1. **Catálogo de Productos:** Búsqueda avanzada, filtros y categorías.
2. **Carrito de Compras:** Gestión de productos seleccionados antes del pago.
3. **Procesamiento de Pagos:** Integración con pasarelas de pago como PayPal o Stripe.
4. **Gestión de Pedidos y Envíos:** Seguimiento del estado del pedido.

**Diagramas UML Recomendados:**

* **Casos de Uso:** Búsqueda de productos, compra, pago, seguimiento de pedidos.
* **Clases:** Producto, Cliente, Carrito, Pedido, Pago, Envío.
* **Secuencia:** Proceso de compra, desde la selección hasta el pago.
* **Estados:** Ciclo de vida de un pedido (Procesando, Enviado, Entregado).

**Herramientas Sugeridas:**

* **Base de Datos:** MongoDB para flexibilidad en productos.
* **Frontend:** Vue.js o Next.js.
* **Backend:** Node.js (Express) o PHP (Laravel).

**3. Sistema de Reservas en Línea**

**Descripción:**

Una aplicación para **reservar servicios**, como citas médicas, mesas en restaurantes o actividades deportivas.

**Módulos Principales:**

1. **Gestión de Servicios:** Registro y administración de servicios ofrecidos.
2. **Calendario de Reservas:** Visualización de fechas y horarios disponibles.
3. **Confirmación y Notificaciones:** Confirmaciones por correo o SMS.
4. **Cancelación y Reprogramación:** Gestión de cambios en reservas.

**Diagramas UML Recomendados:**

* **Casos de Uso:** Reservar, cancelar o reprogramar citas.
* **Clases:** Usuario, Proveedor de Servicio, Reserva, Horario.
* **Secuencia:** Flujo de reserva y cancelación.
* **Estados:** Ciclo de vida de una reserva (Pendiente, Confirmada, Cancelada).

**Herramientas Sugeridas:**

* **Base de Datos:** MySQL o Firebase.
* **Frontend:** React Native (si es móvil) o Angular.
* **Backend:** Node.js (Express) o Django (Python).

**4. Sistema de Gestión de Inventarios**

**Descripción:**

Una aplicación para gestionar inventarios, pedidos y proveedores en **tiendas físicas o en línea**.

**Módulos Principales:**

1. **Gestión de Productos:** Registro de productos, precios y stock.
2. **Órdenes de Compra:** Generación automática al alcanzar el stock mínimo.
3. **Gestión de Proveedores:** Información de contacto y pedidos anteriores.
4. **Reportes:** Análisis de ventas, inventario y tendencias de demanda.

**Diagramas UML Recomendados:**

* **Casos de Uso:** Añadir productos, realizar pedidos, generar reportes.
* **Clases:** Producto, Proveedor, Pedido, Usuario.
* **Secuencia:** Proceso de ventas y actualización de inventario.
* **Estados:** Ciclo de vida de un pedido (Pendiente, Aprobado, Enviado).

**Herramientas Sugeridas:**

* **Base de Datos:** PostgreSQL o SQLite (para versiones locales).
* **Frontend:** Angular o Bootstrap (si es web).
* **Backend:** Java (Spring Boot) o Node.js (Express).

**5. Sistema de Gestión de Proyectos**

**Descripción:**

Una plataforma para **planificar, asignar y seguir el progreso** de tareas en proyectos colaborativos.

**Módulos Principales:**

1. **Gestión de Proyectos:** Crear, modificar y archivar proyectos.
2. **Tareas y Asignaciones:** Detalle de tareas con fechas de inicio y fin.
3. **Tableros Kanban y Gantt:** Visualización del progreso del proyecto.
4. **Reportes de Desempeño:** Análisis de eficiencia y cumplimiento de plazos.

**Diagramas UML Recomendados:**

* **Casos de Uso:** Crear proyecto, asignar tareas, generar reportes.
* **Clases:** Proyecto, Tarea, Usuario, Equipo.
* **Secuencia:** Flujo de asignación y seguimiento de tareas.
* **Estados:** Ciclo de vida de una tarea (Asignada, En Progreso, Completada).

**Herramientas Sugeridas:**

* **Base de Datos:** MongoDB o MySQL.
* **Frontend:** React o Vue.js.
* **Backend:** Node.js (NestJS) o Django (Python).

**6. Aplicación de Encuestas y Satisfacción del Cliente**

**Descripción:**

Una herramienta para **crear encuestas, recopilar respuestas y analizar datos** de satisfacción del cliente.

**Módulos Principales:**

1. **Diseño de Encuestas:** Personalización de preguntas y opciones de respuesta.
2. **Distribución y Notificaciones:** Envío de encuestas a clientes por correo o SMS.
3. **Análisis de Respuestas:** Gráficas y reportes analíticos.
4. **Gestión de Usuarios:** Roles de administrador, creador de encuestas y encuestados.

**Diagramas UML Recomendados:**

* **Casos de Uso:** Crear encuesta, responder encuesta, analizar resultados.
* **Clases:** Encuesta, Pregunta, Respuesta, Usuario.
* **Secuencia:** Creación y envío de encuestas.
* **Estados:** Estado de una encuesta (Borrador, Activa, Cerrada).

**Herramientas Sugeridas:**

* **Base de Datos:** Firebase (en tiempo real) o MySQL.
* **Frontend:** Angular o React.
* **Backend:** Node.js (Express) o PHP (Laravel).