



SBS Proyecto Final Integrado

Desarrollo de Aplicación Python con acceso a Base de Datos

Desarrollo de Aplicaciones en Python integradas a SQL Server

Descripción General

Como parte de la evaluación final del curso, cada equipo de trabajo deberá seleccionar **uno de los nueve proyectos propuestos** para su desarrollo e implementación. El proyecto seleccionado deberá ser desarrollado aplicando **todos los conocimientos, técnicas y herramientas** abordadas durante las sesiones de clase, incluyendo: el diseño y normalización de bases de datos relacionales en **SQL Server**, la implementación del patrón de arquitectura **MVC (Modelo-Vista-Controlador)**, el desarrollo backend utilizando **Python** con el framework **Flask** y el ORM **SQLAlchemy** para el mapeo objeto-relacional, así como la construcción de interfaces de usuario responsivas empleando **HTML5, CSS3, Bootstrap 5** y **Jinja2** como motor de plantillas.

Cada equipo tendrá un plazo de **3 a 4 semanas** para completar el desarrollo, debiendo demostrar competencias tanto en programación como en el manejo de bases de datos, implementando operaciones CRUD completas, validaciones, manejo de sesiones, control de errores, y la generación de reportes específicos del dominio de negocio asignado.

El proyecto será evaluado mediante una rúbrica detallada con un valor total de **100 puntos**, considerando aspectos técnicos de base de datos, calidad del código backend, diseño de la interfaz frontend, correcta implementación de la lógica de negocio, y la documentación entregada.

Instrucciones Generales

1. Cada equipo deberá comunicar su selección de proyecto al docente.
2. No se permitirá que dos equipos de la misma especialidad trabajen en el mismo proyecto.
3. Deben realizar los diagramas Entidad-Relación e implementar CRUD completo para todas las entidades.
4. La asignación será por orden de solicitud (primero en llegar, primero en elegir).
5. Una vez seleccionado, el proyecto no podrá ser cambiado.

Fecha Límite de Entrega

28 Enero 2026



SBS Proyecto Final Integrado

Desarrollo de Aplicación Python con acceso a Base de Datos

Proyectos Elegibles

Nº	Proyecto	Funcionalidades esperadas
1	Sistema de Gestión de Hospital	<ul style="list-style-type: none">• Entidades principales: Pacientes, Médicos, Especialidades, Citas, Historiales Médicos <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">• Programación de citas médicas (validar disponibilidad del médico)• Historial médico por paciente• Reportes/Consultas: Citas por día, pacientes por médico, especialidades más solicitadas <p>Complejidad adicional: Control de horarios de atención por médico</p>
2	Sistema de Gestión de Hotel	<ul style="list-style-type: none">• Entidades principales: Huéspedes, Habitaciones, Tipos Habitación, Reservaciones, Servicios <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">• Reservación de habitaciones (validar disponibilidad por fechas)• Check-in / Check-out• Facturación con servicios adicionales• Reportes/Consultas: Ocupación por período, ingresos por tipo de habitación <p>Complejidad adicional: Cálculo automático de tarifas por temporada</p>
3	Sistema de Gestión de Restaurante	<p>Entidades principales: Clientes, Mesas, Menú, Categorías, Pedidos, Detalles Pedido</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">• Gestión de mesas (disponible/ocupada/reservada)• Toma de pedidos con múltiples productos• Cálculo de cuenta con impuestos y propinas• Reportes/Consultas: Ventas diarias, platillos más vendidos, mesas más utilizadas



SBS Proyecto Final Integrado

Desarrollo de Aplicación Python con acceso a Base de Datos

		Complejidad adicional: Sistema de reservaciones por horario
4	Sistema de Gestión de Gimnasio	<p>Entidades principales: Miembros, Membresías, TiposMembresía, Clases, Entrenadores, Inscripciones</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">• Registro de membresías con fechas de vigencia• Inscripción a clases (control de cupo máximo)• Control de acceso (membresía activa/vencida)• Reportes/Consultas: Miembros activos, clases más populares, ingresos por membresía <p>Complejidad adicional: Renovación automática y alertas de vencimiento</p>
5	Sistema de Gestión de Veterinaria	<p>Entidades principales: Propietarios, Mascotas, Especies, Veterinarios, Consultas, Tratamientos</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">• Registro de mascotas por propietario• Historial de consultas y tratamientos• Programación de citas y vacunaciones• Reportes/Consultas: Consultas por período, especies atendidas, tratamientos frecuentes <p>Complejidad adicional: Calendario de vacunación con recordatorios</p>
6	Sistema de Gestión de Taller Mecánico	<p>Entidades principales: Clientes, Vehículos, Mecánicos, OrdenesServicio, Servicios, Repuestos</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">• Registro de vehículos por cliente• Órdenes de servicio con múltiples trabajos y repuestos• Estados de orden (recibido, en proceso, terminado, entregado)



SBS Proyecto Final Integrado

Desarrollo de Aplicación Python con acceso a Base de Datos

		<ul style="list-style-type: none">Reportes/Consultas: Servicios más solicitados, ingresos por mecánico, historial por vehículo <p>Complejidad adicional: Cotización previa y aprobación del cliente</p>
7	Sistema de Gestión de Cine	<p>Entidades principales: Películas, Salas, Funciones, Clientes, Boletos, Géneros</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">Programación de funciones por sala y horarioVenta de boletos (control de asientos disponibles)Cancelación de boletos con política de tiempoReportes/Consultas: Películas más vistas, ocupación por sala, ventas por función <p>Complejidad adicional: Selección visual de asientos</p>
8	Sistema de Gestión de Inmobiliaria	<p>Entidades principales: Propiedades, TiposPropiedad, Propietarios, Clientes, Contratos, Agentes</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">Catálogo de propiedades (venta/alquiler)Gestión de contratos de venta o alquilerAsignación de agentes a propiedadesReportes/Consultas: Propiedades disponibles, ventas por agente, comisiones <p>Complejidad adicional: Cálculo de comisiones por tipo de operación</p>
9	Sistema de Gestión de Escuela/Academia	<p>Entidades principales: Estudiantes, Profesores, Cursos, Materias, Inscripciones, Calificaciones</p> <p>Funcionalidades requeridas:</p> <ul style="list-style-type: none">Inscripción de estudiantes a cursosRegistro de calificaciones por materiaControl de cupos por cursoReportes/Consultas: Promedio por estudiante, aprobados/reprobados, cursos más demandados



SBS Proyecto Final Integrado

Desarrollo de Aplicación Python con acceso a Base de Datos

		Complejidad adicional: Generación de boletas de calificaciones
--	--	---

Tecnologías Requeridas

- **Lenguaje:** Python 3.x
- **Framework Web:** Flask
- **ORM:** SQLAlchemy
- **Base de Datos:** Microsoft SQL Server (LocalDB o Express)
- **Frontend:** HTML5, CSS3, Bootstrap 5, JavaScript
- **Motor de Plantillas:** Jinja2
- **Arquitectura:** MVC (Model-View-Controller)

Estructura de Carpetas Requerida:

```
proyecto/
├── app.py
├── config.py
├── database.py
├── models.py (o carpeta model/)
├── controllers/
│   ├── __init__.py
│   └── [entidad]_controller.py
├── templates/
│   ├── base.html
│   ├── index.html
│   └── [entidad]/
│       ├── lista.html
│       ├── nuevo.html
│       ├── editar.html
│       └── detalle.html
├── static/
│   ├── css/
│   └── js/
└── README.md
```

Funcionalidades Mínimas:

1. Mínimo 5 tablas relacionadas



SBS Proyecto Final Integrado

Desarrollo de Aplicación Python con acceso a Base de Datos

2. CRUD completo para cada entidad principal
3. Al menos 2 relaciones uno-a-muchos
4. Al menos 1 relación muchos-a-muchos (o tabla intermedia)
5. Validaciones en formularios
6. Mensajes flash de confirmación/error
7. Navegación funcional entre todas las secciones
8. Al menos 3 reportes/consultas específicas del negocio

Lorenzo Martínez