

## Trabajo Obligatorio

Diseñar e implementar un sistema de información para la **gestión de salas de estudio** en la universidad. El sistema debe permitir la **reserva, control de asistencia y generación de reportes** que apoyen tanto a la gestión académica como a la toma de decisiones.

### Sistema para Gestión de Reserva de Salas de Estudio

La UCU cuenta en sus instalaciones con “*Salas de Estudio*”, que son espacios diseñados para diversos usos, como reuniones entre estudiantes o docentes, videoconferencias, etc. Actualmente, el seguimiento y administración de las reservas es bastante rudimentario, siendo realizado en planillas de papel por funcionarios de biblioteca, secretaría y administración.

Lo que se pretende con este proyecto es facilitarles a estos colaboradores, un sistema de gestión de reservas unificado que permita mantener un control y trazabilidad de las salas en toda la universidad, de forma que permita regular de manera equilibrada el uso de estas instalaciones.

Los turnos de las salas empiezan a las 8 de la mañana y terminan a las 11 de la noche, solo es posible reservar la sala por bloques de hora. Por ejemplo, en caso de querer reservar una sala desde las 8:30 AM hasta las 10 de la mañana, el alumno tiene que reservar 2 bloques de una hora (de 8:00 AM a 9:00 AM, de 9:00 AM a 10 AM).

Los alumnos pueden ser estudiantes de grado o de posgrado, los docentes también pueden reservar salas, existen 3 tipos de salas:

- Uso libre
  - Profesores, estudiantes de grado o posgrado las pueden reservar
- Exclusivas de posgrado
- Exclusivas de docentes

No se pueden ocupar las salas por más de 2 horas diarias en cualquiera de los edificios y no es posible participar de más de 3 reservas activas en una semana. En el caso de los docentes y estudiantes de posgrado, no tienen ninguna de estas limitaciones con las salas que son exclusivas para ellos.

Cada reserva tiene asociado los datos de todos los participantes (alumnos y/o profesores) que van a ocupar la sala, la cantidad de participantes no puede exceder la capacidad de la sala. El sistema deberá registrar la asistencia de cada uno de los participantes, en caso de que ninguno de los participantes se manifieste en el día y horario de la reserva de la sala, serán notificados y sancionados con dos meses sin poder realizar reservas.

## Lo que se pide

La UCU está buscando implementar un primer acercamiento que facilite a los administrativos:

- Alta, baja y modificación (ABM) de participantes
- ABM de salas
- ABM de reservas de las salas (teniendo en cuenta las reglas y restricciones enunciadas anteriormente)
- ABM de sanciones a participantes

## Consultas

Con el fin de evaluar de obtener métricas para el equipo de BI de la universidad, se solicita además un sistema de reportes donde se pueda consultar:

- Salas más reservadas
- Turnos más demandados
- Promedio de participantes por sala
- Cantidad de reservas por carrera y facultad
- Porcentaje de ocupación de salas por edificio
- Cantidad de reservas y asistencias de profesores y alumnos (grado y posgrado)
- Cantidad de sanciones para profesores y alumnos (grado y posgrado)
- Porcentaje de reservas efectivamente utilizadas vs. canceladas/no asistidas
- Adicionar otras tres consultas sugeridas por ustedes.

## Base de Datos

Para esto deberá crear una base de datos relacional (SQL) con al menos, las siguientes estructuras:

- login (correo, contraseña)
- participante (ci, nombre, apellido, email)
- programa\_académico (nombre\_programa, id\_facultad, tipo [grado, posgrado])
- participante\_programa\_académico (id\_alumno\_programa, ci\_participante, nombre\_programa, rol [alumno, docente])
- facultad (id\_facultad, nombre)
- sala (nombre\_sala, edificio, capacidad, tipo\_sala [libre, posgrado, docente])
- edificio (nombre\_edificio, dirección, departamento)
- turno (id\_turno, hora\_inicio, hora\_fin)
- reserva (id\_reserva, nombre\_sala, edificio, fecha, id\_turno, estado [activa, cancelada, sin asistencia, finalizada])

- reserva\_participante (ci participante, id reserva, fecha\_solicitud\_reserva, asistencia [true, false])
- sancion\_participante (ci participante, fecha inicio, fecha fin)

## Requisitos del entregable

- Script completo SQL para creación de la base de datos
- Base de datos cargada con datos maestros
  - **Estos tienen que ser suficientes** como para demostrar correctamente el funcionamiento el día de la defensa
- Aplicación funcional:
  - Debe compilar
  - Debe implementar todos los requerimientos funcionales
- Instructivo completo para correr la aplicación de forma local
- Informe del trabajo realizado:
  - Fundamentar decisiones de implementación
  - Mejoras implementadas o consideradas en el modelo de datos
  - Bitácora del trabajo realizado
  - Bibliografía

## Consideraciones

- El backend debe ser desarrollado en Python
- La base de datos debe ser MySQL
- El framework de frontend es de libre elección (si aplica)
- No se puede utilizar ningún ORM
- Restricciones de seguridad a nivel de bases de datos
- Validación de campos en todas las capas (front-end, back-end y base de datos)

## Se valorará:

- Utilización de repositorio de GitHub público
  - El instructivo para correr la aplicación debe estar en el Readme del repo
- Dockerización de la aplicación y Docker-compose con servicios de app, bases de datos, etc

Entrega de letra: **26/09/2025**

Avance: **31/10/2025**

Entrega final: **23/11/2025**

Defensas: **05/12/2025**