

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Sistemas de bases de datos 2 - Sección A  
Ing. Otto Amílcar Rodríguez Acosta  
Aux. Emiliano Jose Alexander Velasquez Najera



# Proyecto 1

Escuela de Ciencias y  
Sistemas Sistemas de Bases  
de Datos 2

**1er. Semestre 2025**

## Objetivo General

Diseñar e implementar un sistema distribuido de gestión de reservas de espacios compartidos utilizando Apache Cassandra, asegurando escalabilidad, alta disponibilidad y eficiencia en consultas de grandes volúmenes de datos.

## Objetivos Específicos

### Modelado y Diseño de Datos

- Diseñar un modelo de datos optimizado para Apache Cassandra, aplicando Primary Keys compuestas y particionamiento eficiente.
- Implementar tablas especializadas con estrategias de denormalización controlada, eliminando la necesidad de JOINS.

### Implementación de un Clúster de Cassandra

- Configurar y desplegar un clúster de al menos 3 nodos, asegurando una distribución óptima de datos.
- Establecer el Replication Factor y definir la estrategia de replicación más adecuada (SimpleStrategy o NetworkTopologyStrategy).

### Optimización y Manejo de Datos

- Insertar y manejar un gran volumen de datos (+100,000 registros de reservas), aplicando Batch Writes para optimización.
- Implementar consultas eficientes en CQL, minimizando el uso de ALLOW FILTERING.
- Aplicar estrategias de compactación y caché para mejorar la latencia de lectura y escritura.

### Pruebas de Escalabilidad y Tolerancia a Fallos

- Simular fallos de nodos y analizar recuperación del sistema y disponibilidad de datos.
- Evaluar el impacto de diferentes Consistency Levels (ONE, QUORUM, ALL) en el rendimiento y la consistencia de los datos.

# Descripción

En este proyecto, los estudiantes diseñarán e implementarán una base de datos distribuida en Apache Cassandra, optimizada para la gestión de reservas de espacios compartidos en un sistema de coworking o renta de salas de conferencia.

El sistema deberá garantizar:

- Baja latencia en consultas mediante una correcta elección de claves de partición.
- Alta disponibilidad con configuración de clúster multi-nodo.
- Eficiencia en inserción y consulta para gestionar múltiples reservas simultáneas.
- Uso adecuado de modelado basado en queries, evitando consultas costosas. Escalabilidad horizontal, asegurando rendimiento ante alta concurrencia.

**El desarrollo del proyecto debe abarcar las siguientes fases:**

**Modelo Conceptual:** Definición de entidades, relaciones y atributos mediante un **modelo de entidad-relación (ER)**.

**Modelo Lógico:** Transformación del modelo conceptual a una estructura optimizada para **Apache Cassandra**, evitando relaciones tradicionales y aplicando **denormalización controlada**.

**Modelo Físico:** Implementación de particionamiento de datos, claves primarias y secundarias, replicación y políticas de indexación.

# Lógica del Negocio: Sistema de Reservas de Espacios Compartidos

- El sistema gestionará la reserva de espacios en un centro de coworking o salas de conferencias, permitiendo:
- Usuarios: Personas que pueden realizar reservas.
- Espacios: Salas, escritorios o auditorios disponibles para reservar.
- Reservas: Registros de ocupación de un espacio en un horario determinado.

## Relaciones y atributos:

- Cada **usuario** puede hacer **múltiples reservas**.
  - Cada **reserva** pertenece a **un único espacio** en una fecha y hora específica.
  - Cada **espacio** tiene un **estado de disponibilidad** y una capacidad **máxima**.
- 
- De cada Usuario se debe guardar el Nombre, Email, DPI, Telefono, NIT o CF
  - De cada Espacio se debe guardar el Nombre, Tipo, Capacidad Máxima y Ubicación
  - De cada Reserva se debe guardar el Nombre, El Usuario que reserva, El Espacio Reservado, Fecha, Hora Inicio, Hora Fin.

## Reglas de Negocio

- Un usuario puede reservar varios espacios en diferentes fechas.
- Un espacio solo puede estar reservado por un usuario a la vez en un mismo horario.
- Las reservas pueden modificarse o cancelarse antes de la fecha de uso.
- Se deben registrar horarios de uso, duración y estado de la reserva.
- Se deben permitir consultas de disponibilidad en base a espacio, fecha y usuario.

# Fases de Desarrollo del Proyecto

## Modelo Conceptual

- Diseñar un modelo entidad-relación (ER) que represente la lógica del negocio.
- Definir entidades principales: Usuarios, Espacios, Reservas.
- Establecer relaciones y atributos clave.

## Modelo Lógico

- Convertir el modelo conceptual en **estructuras optimizadas para Cassandra**.
- Eliminar relaciones tradicionales (**evitar JOINS y normalización excesiva**).
- Crear **tablas denormalizadas** según las consultas más frecuentes.

## Modelo Físico

Esta fase define la optimización y gestión del almacenamiento de datos en el sistema distribuido.

## Llaves Primarias y Particionamiento

- Definir claves de partición eficientes para **distribuir datos en nodos de forma balanceada**.
- Establecer **claves de clustering** para ordenar los datos según las necesidades de consulta.

## Índices y Optimización de Consultas

- Crear **Secondary Indexes** solo cuando sea estrictamente necesario.
- Implementar **Materialized Views** para consultas optimizadas.

## Replicación y Tolerancia a Fallos

- Configurar **replicación** con **SimpleStrategy** o **NetworkTopologyStrategy**.
- Ajustar **Consistency Level** para balancear disponibilidad y rendimiento.

## Estrategias de Compactación y Caché

- Configurar **Memtable y Compaction Strategy** para mejorar rendimiento.
- Implementar **TTL (Time-To-Live)** para eliminar reservas expiradas automáticamente.

# Requerimientos Técnicos del Proyecto

## Modelado de Datos en Cassandra

- Diseñar una estructura eficiente en **Apache Cassandra**, asegurando una correcta **distribución de datos en nodos**.
- Aplicar **Primary Key compuesta** con **Partition Key bien diseñada** para garantizar acceso rápido a la información.
- Desnormalizar los datos para optimizar consultas (**sin JOINS en Cassandra**).
- Crear **múltiples tablas especializadas** para evitar **ALLOW FILTERING**.

## Configuración del Clúster de Cassandra

- Implementar un **clúster de 3 nodos** en entornos locales o en la nube.
- Configurar el **Replication Factor** y el **Consistency Level** adecuados para balancear disponibilidad y rendimiento.
- Ajustar la estrategia de replicación **SimpleStrategy o NetworkTopologyStrategy**, dependiendo del entorno.

## Inserción y Consulta de Datos

- Insertar **al menos 100,000 reservas** en el sistema.
- Usar **Batch Writes** para optimizar la inserción masiva.
- Implementar consultas eficientes para:
  - **Consultar disponibilidad de un espacio en una fecha.**
  - **Ver historial de reservas de un usuario.**
  - **Obtener ocupación de espacios en un rango de fechas.**

---

## Pruebas de Escalabilidad y Tolerancia a Fallos

- **Simular fallos de nodos** apagando un nodo del clúster y verificando la recuperación.
- Medir el impacto de diferentes **Consistency Levels** (**ONE**, **QUORUM**, **ALL**) en rendimiento y confiabilidad.
- Optimizar **estrategias de compactación y caché** para mejorar latencias.
- **Uso incorrecto de claves de partición o clustering keys.**
- **Consultas ineficientes** que requieran **ALLOW FILTERING** en grandes volúmenes de datos.
- **Falta de configuración de replicación y consistencia adecuada.**

## Restricciones

- El proyecto se realizará en grupos no mayores de 3 personas.
- Las entregas tarde están sujetas a una penalización correspondiente.
- Copias totales o parciales tendrán nota de 0

**FECHA DE ENTREGA:** 4 DE ABRIL 2025 A LAS 23:59