

Unidad 2: Capa de Almacenamiento

1. OBJETIVOS

Comprender y Aplicar Diferencias entre Bases de Datos SQL y NoSQL:

- Los estudiantes deben ser capaces de identificar y justificar la elección entre una base de datos SQL y una NoSQL basándose en las necesidades específicas del proyecto IoT. Esto incluye entender las implicaciones de cada tipo en términos de estructura de datos, escalabilidad, rendimiento y consistencia.

2. Implementar Bases de Datos Adecuadas para el Manejo Efectivo de Datos en IoT:

- Cada grupo de estudiantes implementará al menos una base de datos SQL (MySQL) y una NoSQL (MongoDB), configurándolas adecuadamente en un entorno de servidor y realizando integraciones básicas con APIs. Esto también incluye la creación de esquemas y realización de pruebas de inserción y recuperación de datos.

3. Desarrollar y Aplicar Estrategias de Seguridad para la Protección de Datos:

- Los estudiantes deberán aplicar prácticas de seguridad en las bases de datos, incluyendo el cifrado de datos y la configuración de controles de acceso. Además, deben abordar consideraciones éticas en el manejo de datos, desarrollando y documentando políticas que cumplan con las normativas éticas y legales.

4. Elaborar Documentación Técnica Completa y Recibir Retroalimentación Crítica:

- Los alumnos redactarán documentación técnica detallada en inglés para toda la capa de almacenamiento, incluyendo la configuración, las operaciones realizadas y las políticas de seguridad. Deberán presentar su trabajo, recibir feedback y realizar mejoras basadas en las críticas recibidas.

DESARROLLO

Stack Tecnológico y Metodológico Común:

- Control de Versiones: Git y GitHub.
- Metodologías Ágiles: Scrum y Kanban.
- Aprendizaje Basado en Proyectos (ABP): Estructurado en fases de Requisitos, Investigación, Prototipo y Presentación.
- Soporte DevOps: Proporcionado por el docente para mantener la infraestructura y el flujo de trabajo del curso.

Semana 5: Elección entre SQL y NoSQL

- **Objetivo de la Semana:** Entender las diferencias fundamentales entre las bases de datos SQL y NoSQL, y elegir la adecuada para el proyecto.
- **Actividades:**
 1. Investigación sobre las diferencias entre SQL y NoSQL.
 2. Análisis de casos de uso de SQL y NoSQL en IoT.
 3. Discusión en grupo para seleccionar la base de datos para su proyecto específico.
 4. Documentación inicial en GitHub sobre la elección y justificación.
- **Historia de Usuario:** "Como estudiante de IoT, quiero entender las diferencias entre SQL y NoSQL para poder elegir la mejor base de datos para mi proyecto específico de IoT."
- **Sprint 5 [Product Backlog]:**
 1. Investigar las diferencias clave entre SQL y NoSQL.
 2. Analizar ventajas de SQL para transacciones y relaciones complejas.
 3. Evaluar ventajas de NoSQL para escalabilidad y flexibilidad.
 4. Discutir en grupo las necesidades específicas del proyecto IoT.
 5. Elegir la base de datos adecuada para el proyecto.
 6. Justificar la elección de la base de datos en base a requisitos técnicos.
 7. Crear una presentación breve sobre la elección de la base de datos.
 8. Documentar la decisión y justificación en GitHub.
 9. Preparar un debate en clase sobre la elección de la base de datos.
 10. Reflexionar sobre cómo la elección afecta al diseño general del sistema.
 11. Redactar un resumen de las características clave de la base de datos elegida.
 12. Planificar la instalación y configuración inicial para la próxima semana.

Semana 6: Implementación Práctica de Bases de Datos

- **Objetivo de la Semana:** Configurar las bases de datos seleccionadas y comenzar su integración con el sistema IoT.
- **Actividades:**
 1. Instalación y configuración de MongoDB y MySQL.
 2. Creación de esquemas y estructuras iniciales de datos.
 3. Integración básica con APIs para pruebas de inserción y consulta.
 4. Registro de configuraciones y primeras pruebas en GitHub.

- **Historia de Usuario:** "Como estudiante de IoT, necesito configurar y utilizar MongoDB y MySQL para manejar los datos de mi proyecto IoT de manera efectiva."
- **Sprint 6 [Product Backlog]:**
 1. Instalar MongoDB y MySQL en el servidor local o en Docker.
 2. Configurar las bases de datos con parámetros iniciales.
 3. Crear esquemas de datos y estructuras para ambos tipos de bases de datos.
 4. Desarrollar scripts iniciales para la inserción de datos.
 5. Implementar funciones básicas de consulta en ambas bases de datos.
 6. Probar la inserción y recuperación de datos en ambas bases de datos.
 7. Documentar la configuración y los scripts en GitHub.
 8. Realizar una revisión de código en equipo de los scripts desarrollados.
 9. Preparar ejemplos de cómo las APIs interactuarán con las bases de datos.
 10. Desarrollar un mini-taller para explicar las operaciones básicas de la base de datos.
 11. Analizar el rendimiento inicial y discutir mejoras.
 12. Registrar problemas y errores comunes durante la instalación y configuración.

Semana 7: Seguridad y Ética en el Almacenamiento de Datos

- **Objetivo de la Semana:** Asegurarse de que el almacenamiento de datos sea seguro y cumpla con las normativas éticas.
- **Actividades:**
 1. Estudio de mejores prácticas de seguridad en bases de datos.
 2. Implementación de medidas de seguridad básicas como cifrado y control de acceso.
 3. Discusión sobre ética en el manejo de datos personales.
 4. Documentación de las políticas de seguridad y ética en GitHub.
- **Historia de Usuario:** "Como estudiante de IoT, quiero implementar prácticas de seguridad en las bases de datos para asegurar los datos y cumplir con las normativas éticas."
- **Sprint 7 [Product Backlog]:**
 1. Investigar normativas de seguridad aplicables a las bases de datos en IoT.
 2. Implementar medidas de seguridad como el cifrado de datos en reposo.

3. Configurar controles de acceso para las bases de datos.
4. Desarrollar políticas de seguridad documentadas para el proyecto.
5. Organizar una discusión de clase sobre ética en el manejo de datos.
6. Redactar un documento sobre las prácticas éticas adoptadas en el proyecto.
7. Simular ataques de seguridad comunes y registrar las respuestas del sistema.
8. Crear una lista de verificación de seguridad para futuras auditorías.
9. Discutir cómo la arquitectura del proyecto afecta la seguridad de los datos.
10. Documentar todas las medidas de seguridad implementadas.
11. Evaluar la efectividad de las políticas de seguridad.
12. Preparar un informe sobre las lecciones aprendidas en seguridad.

Semana 8: Documentación Técnica y Feedback

- **Objetivo de la Semana:** Finalizar la documentación técnica del proyecto en inglés y preparar una presentación para obtener feedback.
- **Actividades:**
 1. Redacción final de toda la documentación técnica en inglés.
 2. Preparación de una presentación del proyecto para la clase.
 3. Realización de la presentación y recopilación de feedback.
 4. Reflexiones finales y ajustes en la documentación en GitHub.
- **Historia de Usuario:** "Como estudiante de IoT, necesito completar la documentación técnica en inglés y presentar mi proyecto para obtener feedback y realizar mejoras."
- **Sprint 8 [Product Backlog]:**
 1. Redactar la documentación técnica final de todo el sistema de base de datos.
 2. Preparar diagramas detallados de la arquitectura de datos.
 3. Desarrollar guías de usuario para la interacción con las bases de datos.
 4. Compilar todos los documentos en un formato coherente y profesional.
 5. Ensayar la presentación del proyecto a la clase.
 6. Recibir y registrar el feedback de compañeros y profesores.
 7. Ajustar la documentación basada en el feedback recibido.
 8. Finalizar la documentación en inglés para una audiencia global.
 9. Crear un video resumen del proyecto para su difusión.
 10. Analizar cómo la documentación puede ser mejorada en futuros proyectos.
 11. Archivar todos los documentos en GitHub con acceso adecuado.
 12. Celebrar una reunión de cierre para discutir los logros y los desafíos del proyecto.

Stack Tecnológico Específico:

- IDE: Visual Studio Code, utilizado para el desarrollo y gestión de bases de datos.
- Bases de Datos: MongoDB y MySQL, cubriendo tanto enfoques NoSQL como SQL.
- Herramientas de Contenedorización: Docker y Kubernetes, facilitando la implementación y escalabilidad de las bases de datos.
- Lenguajes de Programación: Python y JavaScript, utilizados para scripts de manejo de bases de datos y automatización de tareas.
- Herramientas Adicionales: Postman, para testear y verificar la integración de las APIs con las bases de datos.

Infraestructura:

- Servidor Raspberry Pi con Ubuntu Server y NGINX, configurado y mantenido por el docente para soportar todas las operaciones de bases de datos.