

Documento Anexo a la Unidad 3

Stacks Tecnológicos para Desarrollo de la API

Propósito

Este documento complementa la Unidad 3: "Capa de Aplicación" del programa del curso, proporcionando una visión detallada de los stacks tecnológicos propuestos para el desarrollo de la API que facilitará la comunicación entre la Capa de Conectividad y la Capa de Almacenamiento dentro de proyectos IoT.

Stacks Tecnológicos Propuestos

1. Node.js y Express

Descripción: Node.js ofrece un entorno de ejecución para JavaScript en el servidor, permitiendo el desarrollo de aplicaciones web y APIs de manera eficiente. Express, un framework minimalista para Node.js, simplifica la creación de APIs RESTful facilitando la gestión de rutas, solicitudes y respuestas.

Justificación Pedagógica: La simplicidad y la amplia adopción en la industria de Node.js y Express hacen de este stack una opción excelente para introducir a los estudiantes al desarrollo backend y la creación de APIs. La comunidad activa y los abundantes recursos de aprendizaje apoyan un entorno educativo dinámico.

Recursos Sugeridos: Documentación oficial de Node.js y Express, tutoriales interactivos, y ejercicios prácticos de codificación.

2. Python y Flask/Django

Descripción: Python es un lenguaje de programación de alto nivel con una sintaxis clara y legible. Flask y Django son frameworks web para Python, donde Flask ofrece más flexibilidad y simplicidad, y Django proporciona una solución más completa y robusta con características 'listas para usar'.

Justificación Pedagógica: La versatilidad de Python y la elección entre Flask y Django permiten adaptar el proyecto a las necesidades educativas, desde APIs simples hasta aplicaciones web completas. Python es ampliamente reconocido por su curva de aprendizaje accesible, lo que lo hace ideal para estudiantes de todos los niveles.

Recursos Sugeridos: Documentación de Flask y Django, cursos en línea de Python para desarrollo web, y proyectos de ejemplo.

3. MongoDB y Mongoose/Django REST Framework

Descripción: MongoDB es una base de datos NoSQL orientada a documentos, que almacena datos en formato JSON, facilitando su manipulación desde lenguajes de programación modernos. Mongoose es una librería para modelado de datos en MongoDB con Node.js, mientras que Django REST Framework es una poderosa herramienta para crear APIs web con Django.

Justificación Pedagógica: Enseñar a los estudiantes cómo interactuar con bases de datos NoSQL como MongoDB les proporciona habilidades relevantes para el desarrollo moderno de aplicaciones. La integración de la base de datos con la API es crucial para el manejo de datos en proyectos IoT.

Recursos Sugeridos: Documentación oficial de MongoDB, Mongoose, y Django REST Framework, tutoriales prácticos sobre integración de bases de datos con APIs.

4. Herramientas de Prototipado como Figma

Descripción: Aunque el foco de esta unidad no está en el desarrollo del frontend, herramientas de prototipado como Figma permitirán a los estudiantes diseñar la interfaz de usuario que interactuará con la API. Esto prepara el terreno para el desarrollo más detallado del frontend en la Unidad 7.

Justificación Pedagógica: Introducir herramientas de prototipado fomenta la creatividad y ayuda a los estudiantes a visualizar cómo los usuarios finales interactuarán con los datos del proyecto IoT. La habilidad para diseñar interfaces de usuario es complementaria al desarrollo técnico de la API.

Recursos Sugeridos: Tutoriales de Figma, ejercicios de diseño de interfaces de usuario, y proyectos de prototipado.

5. Desarrollo Frontend Básico con JavaScript, HTML y CSS

Descripción: Para la construcción de interfaces de usuario básicas que interactúen con la API, se utilizará JavaScript junto con HTML y CSS. Estas tecnologías permiten a los estudiantes desarrollar páginas web estáticas y dinámicas capaces de solicitar y mostrar datos desde la API desarrollada.

Justificación Pedagógica: Proporcionar a los estudiantes una introducción al desarrollo frontend les permite comprender el ciclo completo de desarrollo de aplicaciones web. A

través del manejo básico del Document Object Model (DOM) con JavaScript, los estudiantes aprenderán a interactuar con elementos HTML y aplicar estilos con CSS, lo que es fundamental para la creación de aplicaciones web modernas.

Recursos Sugeridos:

- Documentación oficial de JavaScript, HTML y CSS.
- Tutoriales interactivos y ejercicios prácticos para manipular el DOM.
- Ejemplos de código para integrar solicitudes HTTP a la API desde el frontend.
- Guías de inicio rápido en Visual Studio Code (VSCode) para el desarrollo frontend.

Actividades Clave:

- **Ejercicios de Codificación:** Prácticas guiadas para crear elementos HTML, aplicar estilos con CSS, y escribir scripts de JavaScript que realicen solicitudes HTTP a la API.
- **Taller de Integración:** Sesiones de trabajo donde los estudiantes aplican lo aprendido para desarrollar una interfaz de usuario simple que muestre datos recuperados a través de la API.
- **Proyecto de Mini-Frontend:** Desarrollo de una página web básica que utilice la API para mostrar o enviar datos, reforzando el entendimiento de cómo las capas de aplicación y presentación interactúan en un proyecto IoT.

Evaluación y Feedback:

- Los estudiantes presentarán sus interfaces de usuario, demostrando la capacidad de recuperar y mostrar datos desde la API.
- Se revisará el código para asegurar buenas prácticas en JavaScript, HTML y CSS.
- Se ofrecerá feedback constructivo para mejorar la integración frontend-backend y la usabilidad de la interfaz.