

1 DESCRIPCIÓN DO ANTEPROXECTO

(Adaptar a lonxitude do documento segundo sexa preciso)

TÍTULO: DevOps Hub: Plataforma de gestión para equipos de desarrollo

Introdución

En el entorno actual del desarrollo de software, combinar diversas herramientas y tecnologías es esencial para abarcar todo el ciclo de vida de una aplicación. Los equipos reducidos y los desarrolladores independientes suelen enfrentarse al desafío de configurar y mantener por separado estas soluciones, lo que consume tiempo y recursos valiosos. **DevOps Hub** nace para simplificar este proceso, ofreciendo un espacio centralizado que integra gestión de proyectos, control de versiones, CI/CD y orquestación de contenedores.

Obxetivos

- Implementar un sistema básico de autenticación (registro e inicio de sesión)
- Desarrollar un panel de control (dashboard) para visualizar y crear proyectos
- Establecer una integración básica con la API de GitHub para visualizar repositorios existentes
- Aplicar medidas de seguridad básicas (validación de formularios y consultas parametrizadas).

Historias de usuario

Autenticación

HU1: Como usuario quiero registrarme e iniciar sesión en la plataforma para acceder a mis proyectos.

- *Técnico:* Autenticación básica con sesiones en el backend; formulario de login en el frontend.

HU2: Como usuario quiero crear nuevos proyectos desde la plataforma.

- *Técnico:* Modelo de datos para proyectos y API REST para altas.

HU3: Como usuario quiero visualizar la lista de mis proyectos en el dashboard.

- *Técnico:* Consulta de la base de datos y renderizado en el frontend.

HU4: Como usuario quiero conectar mi cuenta de GitHub para ver mis repositorios.

- *Técnico:* Uso de la API de GitHub para autenticación básica y listado de repos (solo lectura).

HU5: Como usuario quiero que mis datos estén protegidos frente a accesos no autorizados.

- *Técnico:* Validación en frontend y backend, consultas SQL parametrizadas.

Descripción técnica

La arquitectura se mantiene en **tres capas** (presentación, lógica de negocio y acceso a datos) para maximizar la claridad del código y facilitar el mantenimiento, pero simplificando las tecnologías utilizadas.

Tecnologías:

- **HTML, CSS, JavaScript y Bootstrap (Capa de presentación):** Estas tecnologías son estándar en el desarrollo web, con amplia documentación y comunidad. Bootstrap acelera la creación de interfaces responsive mediante componentes predefinidos, reduciendo tiempo de maquetación y garantizando compatibilidad móvil.
- **Node.js y Express (Capa de negocio):** Node.js ofrece un entorno asíncrono para gestionar las peticiones del cliente. Express es un framework minimalista pero extensible, ideal para construir APIs REST de forma rápida
- **SQLite (Capa de datos):** Base de datos ligera que no requiere instalación ni configuración de servidores adicionales, facilitando el desarrollo local.
- **Git y GitHub (Control de versiones e integración):** Git para el control de versiones local y GitHub para almacenar el código y para la integración con sus APIs, que permitirá a los usuarios visualizar sus repositorios.
- **Medidas de seguridad:** Implementación básica de validación de formularios en cliente y servidor, y uso de consultas parametrizadas para prevenir inyección SQL

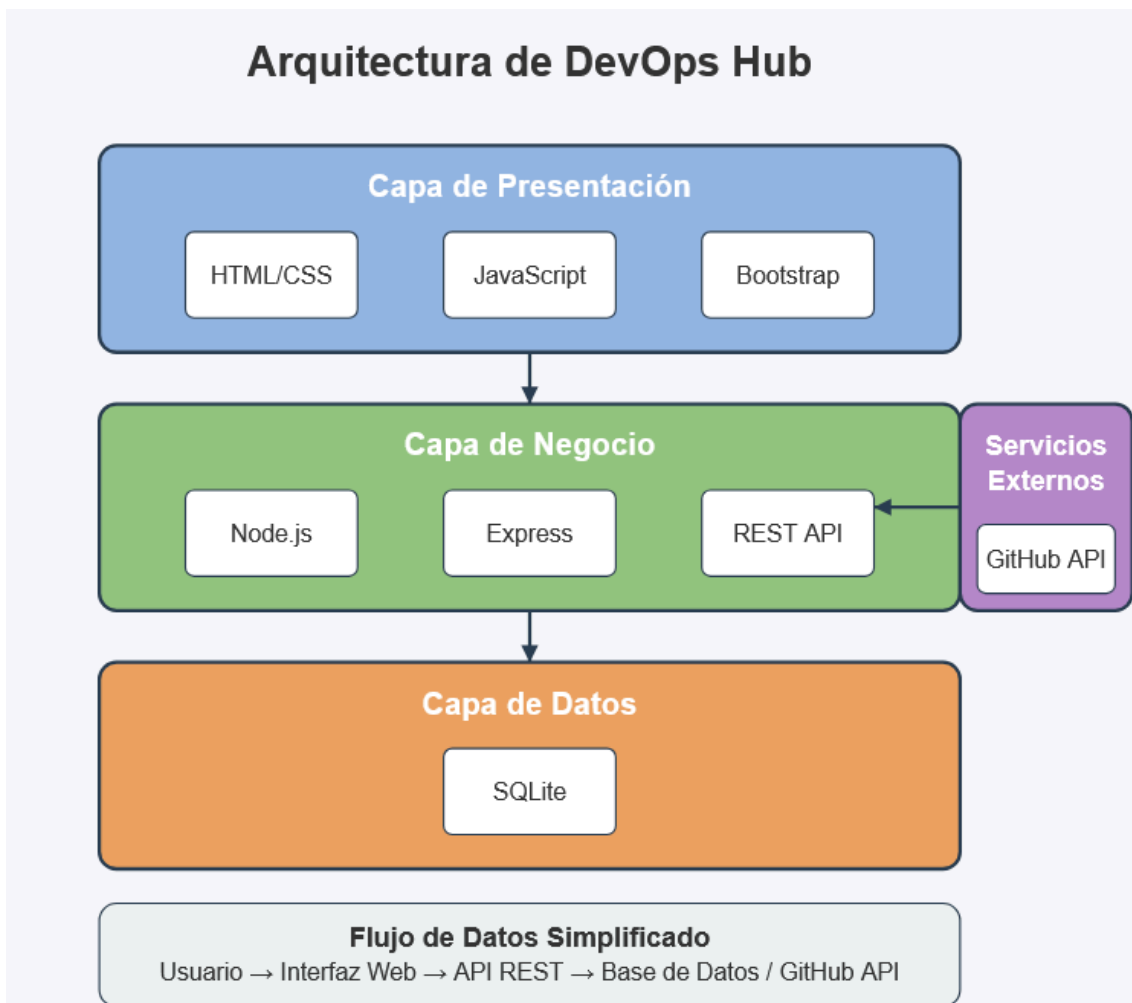


Figura 1: Arquitectura de 3 capas que implementará el sistema.

Proceso de desenvolvimiento a emplegar no proxecto

Voy a realizar el proyecto dentro del marco de trabajo conocido como **Proceso Unificado de Desarrollo de Software**, que combina la dirección por casos de uso, el foco en la arquitectura y un enfoque iterativo e incremental. Este modelo captura los requisitos funcionales y define los contenidos de cada iteración, identificando los riesgos clave desde etapas tempranas para evitar retrasos.

Características del Proceso Unificado

- **Iterativo e incremental:** Organizado en cuatro fases (Inicio, Elaboración, Construcción y Transición), cada una compuesta por varias iteraciones que añaden o mejoran funcionalidades.
- **Dirigido por casos de uso:** Los casos de uso recogen las necesidades y se emplean para planificar y evaluar cada iteración.
- **Centrado en la arquitectura:** La estructura del sistema guía el diseño y la implementación de los casos de uso.
- **Enfocado en riesgos:** Se analizan riesgos críticos de forma continua y se priorizan las tareas para mitigarlos.

Inicio

- **Objetivo:** Definir el MVP y priorizar historias de usuario esenciales (registro, creación/listado de proyectos, conexión con GitHub).
- **Actividades:** Reunión de requisitos personales, modelado básico y análisis de riesgos locales (compatibilidad de librerías, estructura del proyecto).

Elaboración

- **Objetivo:** Establecer la base técnica y validar componentes clave.
- **Actividades:** Organización de la estructura de carpetas, configuración de entorno local y prueba de concepto reducida (autenticación básica + GitHub API).

Construcción

- **Objetivo:** Desarrollar iterativamente cada historia de usuario.
- **Actividades por iteración (1-2 días):**
 1. Selección de historias de usuario según prioridad.
 2. Implementación de la funcionalidad (dashboard, visualización de repositorios).
 3. Prueba manual en entorno local.
 4. Refactorización ligera y actualización de un README con los pasos de arranque.

Transición (Prueba de entorno local funcional)

- **Objetivo:** Verificar en local el comportamiento y funcionalidad completa.
- **Actividades:** Pruebas finales de la aplicación, comprobación del flujo completo de la aplicación y preparación de la documentación final.

Planificación do traballo e estimación temporal

Datos generales

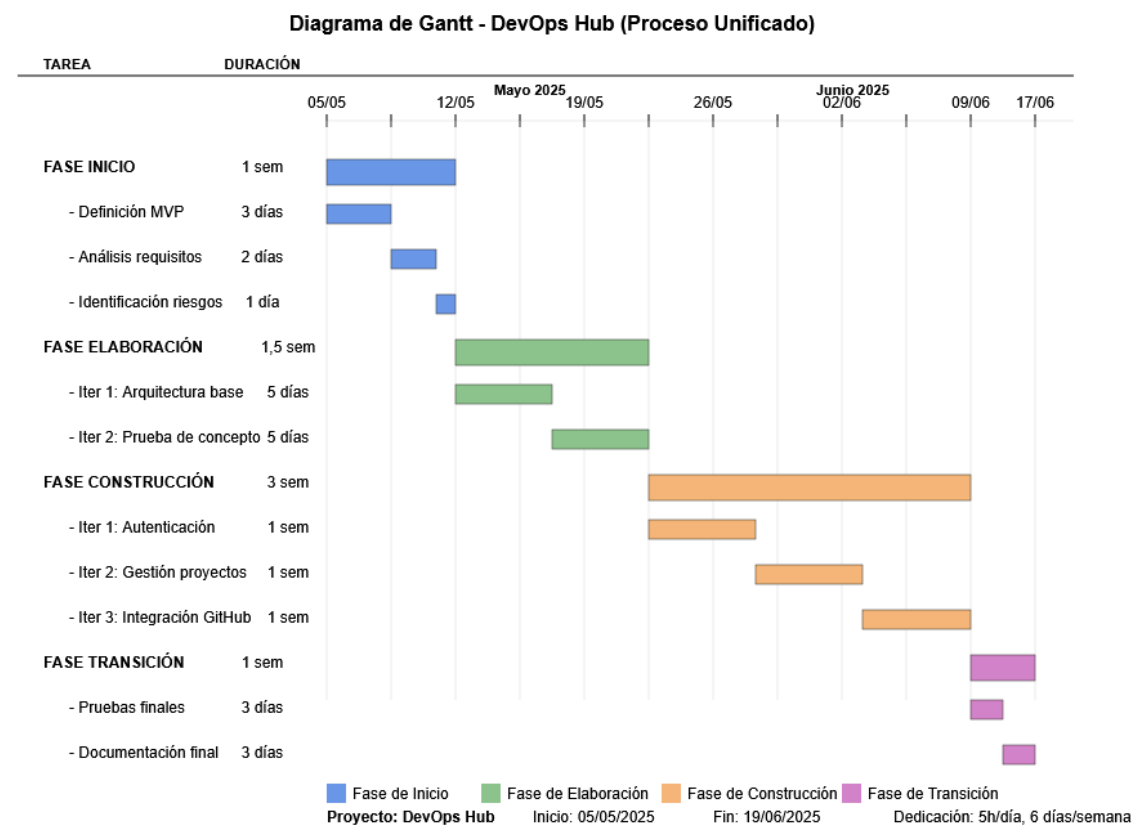
- Dedicación semanal prevista: 30 horas/semana
- Dedicación diaria prevista: 5 horas/día (lunes a sábado)
- Fecha de inicio: 5 mayo 2025
- Fecha de finalización: 19 de Junio 2025

Planificación temporal del proyecto:

Fase	Duración	Fechas	Iteraciones	Horas estimadas
Inicio	1 semana	5 Mayo - 11 Mayo	1	30 h
Definición del producto mínimo viable				15h
Análisis inicial de requisitos				10h
Identificación de riesgos				5h
Elaboración	1,5 semanas	12 Mayo - 21 Mayo	2	45 h
Iteración 1: Arquitectura base		12 Mayo - 16 Mayo		20h
Iteración 2: Prueba de concepto		17 Mayo - 21 Mayo		25h

Construcción	3 semanas	22 Mayo - 11 Junio	3	90 h
Iteración 1: Autenticación + Dashboard	1.5 semanas	22 Mayo - 28 Mayo		30h
Iteración 2: Gestión de proyectos	0.5 semanas	29 May - 4 Jun		30h
Iteración 3: Integración con GitHub		5 Jun - 11 Jun		30h
Transición	1 semana	12 Jun - 19 Jun		30h
Pruebas finales				15h
Documentación final				15h
Total	6.5 semanas	5 may – 19 jun		195 h

Diagrama de Gantt para mostrar la planificación temporal de forma más visual:



Descripción da documentación a entregar no proxecto

Toda la documentación generada se consolidará en un único apartado dividido en dos secciones:

1. - Introducción al problema y contexto del proyecto.
 - - Exposición de las necesidades detectadas y justificación de las tecnologías elegidas (HTML, CSS, JavaScript, Bootstrap, Node.js, Express y SQLite).
 - - Reflexión sobre el análisis y el diseño desarrollados, incluyendo los retos afrontados y las decisiones tomadas para garantizar la viabilidad del proyecto en el plazo establecido.
 - - Documentación de la API REST implementada para la gestión de proyectos.
 - - Manual básico de usuario con capturas de pantalla de la interfaz.
2. **Anexos técnicos (opcional):**
 - - Diagramas de arquitectura de tres capas y modelo entidad-relación de SQLite.
 - - Fragmentos de código clave o configuraciones relevantes (configuración de Express, integración con GitHub API).
 - - Instrucciones para la instalación y puesta en marcha del entorno de desarrollo local.

Medios materiais necesarios

Para el desarrollo y pruebas de DevOps Hub, se requieren los siguientes recursos:

Equipo de desarrollo:

- PC con conexión a Internet para investigación, desarrollo y pruebas locales
 - **Procesador:** Intel Core i5 o equivalente (mínimo Core 2 Duo).
 - **Memoria RAM:** 8 GB (mínimo 4 GB).
 - **Almacenamiento:** Disco duro o SSD de al menos 120 GB.

Accesorios:

- Monitor con resolución mínima de 1080p.
- Teclado y ratón.

Software

- **Sistema operativo:** Windows 10 / Linux / macOS.
- **Entorno de desarrollo:** Visual Studio Code con extensiones para JavaScript y Node.js.
- **Gestión de base de datos:** SQLite y DB Browser for SQLite.
- **Control de versiones y CI/CD:**
 - Git instalado localmente.
 - Cuenta y repositorio en GitHub.
 - GitHub Actions configurado desde el repositorio.
- **Herramientas de modelado y documentación:**
 - Editor de texto o procesador de documentos (Markdown / Word / LibreOffice Writer).
 - Herramienta de diagramación (draw.io u otra herramienta gratuita).

Tecnología	Versión	Coste	Fase del proyecto	Uso previsto
HTML, CSS, JavaScript, Bootstrap	HTML5, CSS3, JS ES6, Bootstrap v5.x	Gratuito	Elaboración y Construcción	Diseño de la interfaz de usuario y desarrollo del dashboard responsivo.
Node.js y Express	Node.js v20.x, Express v4.x	Gratuito	Elaboración y Construcción	Backend y API REST: autenticación, gestión de usuarios y proyectos.
SQLite	SQLite v3.x	Gratuito	Construcción	Base de datos local para usuarios y proyectos, sin necesidad de servidor adicional.
Git y GitHub	Git v2.40+, GitHub (plan gratuito)	Gratuito	Todas las fases	Control de versiones, alojamiento del código, integración con la API.

API de GitHub (REST v3)	v3	Gratuito	Construcción y Transición	Visualización de repositorios del usuario autenticado (modo lectura).
Visual Studio Code	Última versión estable (2025)	Gratuito	Todas las fases	Entorno de desarrollo integrado con soporte para JavaScript y Node.js.
DB Browser for SQLite	v3.12.x	Gratuito	Construcción y pruebas	Gestión y visualización de la base de datos SQLite.
Herramientas de documentación	LibreOffice Writer / draw.io	Gratuito	Todas las fases	Documentación técnica y generación de diagramas de arquitectura y modelo ER.

Despliegue

El despliegue del proyecto está previsto para realizarse en un entorno local de desarrollo. Dado que el objetivo del proyecto es formativo, no se contempla su despliegue en entornos de producción ni en servidores externos.

Modalidad de despliegue:

- **Tipo:** Despliegue local (en el equipo del desarrollador).
- **Entorno de ejecución:** Visual Studio Code con terminal integrada.
- **Pasos previstos:**
 1. Clonar el repositorio del proyecto desde GitHub.
 2. Ejecutar npm install para instalar las dependencias necesarias.
 3. Iniciar el backend mediante npm start o node index.js.
 4. Visualizar el frontend abriendo el archivo index.html en un navegador o utilizando la extensión Live Server de Visual Studio Code.

Requisitos técnicos:

- Sistema operativo: Windows 10, Linux o macOS.
- Software necesario: Node.js (última versión estable), Git, Visual Studio Code.
- Base de datos: SQLite, incluida como archivo local en el proyecto.
- Memoria RAM mínima: 4 GB (se recomienda 8 GB).
- Espacio libre en disco: Al menos 200 MB.
- Conexión a Internet: Necesaria únicamente para acceder a la API de GitHub y descargar dependencias en la instalación inicial.

Bibliografía

Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley.

Mozilla Developer Network. (2023). MDN Web Docs.
<https://developer.mozilla.org/es/>

Bootstrap. (2023). Bootstrap Documentation.
<https://getbootstrap.com/docs/>

Node.js. (2023). Node.js Documentation. <https://nodejs.org/en/docs/>

Express. (2023). Express Documentation. <https://expressjs.com/en/starter>

GitHub. (2023). GitHub REST API Documentation.
<https://docs.github.com/es/rest>

SQLite. (2023). SQLite Documentation. <https://www.sqlite.org/docs.html>

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). El Proceso Unificado de Desarrollo de Software. Addison Wesley.

Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley Professional.

Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.

Observaciones

Rúbrica IA

- ¿Qué herramienta de IA generativa usaste (nombre y versión)?
ChatGPT 4-o y [Claude.ai](#) 3.7 Sonnet
- ¿Para qué usaste la herramienta?
He usado la IA generativa para:
 - Generar la estimación temporal del tiempo que me va a llevar cada fase del proyecto (modificada para realizar un diseño incremental y prototipado)
 - Generación del diagrama de Gantt a partir de dicha estimación temporal
 - Generación de la bibliografía necesaria para documentarme para el proyecto según las tecnologías utilizadas.
 - Tabla de versiones de software, costes, y utilización de software en el proyecto.
- ¿Qué contenidos fueron generados por el agente? Etiquétalos.
Contenidos generados mencionados en la pregunta anterior.
- ¿Cómo has utilizado o cambiado la salida de la IA generativa?
No he realizado modificaciones.