

Desarrollo Backend I - Edición 2025

Microservicios con Laravel y GitHub Copilot

Descripción General

Este repositorio contiene la documentación y materiales del curso **Desarrollo Backend I** correspondiente a la edición 2025. El objetivo principal de la materia es enseñar a los estudiantes cómo desarrollar una API de microservicios utilizando Laravel. En esta edición se utilizará **vibe-coding** junto con **GitHub Copilot** para guiar eficientemente el desarrollo.

Objetivos de Aprendizaje

Objetivo Principal

Que los estudiantes aprendan a desarrollar una API REST de microservicios utilizando Laravel 12, aprovechando GitHub Copilot como asistente de codificación para mejorar la productividad y calidad del código.

Objetivos Específicos

- **Desarrollo Guiado por IA:** Aprender a utilizar GitHub Copilot para acelerar el desarrollo de microservicios
- **Arquitectura de Microservicios:** Diseñar y implementar APIs REST siguiendo patrones arquitectónicos modernos
- **Framework Laravel:** Conocer los elementos más importantes de Laravel 12 para el desarrollo de APIs sin frontend
- **Metodologías Ágiles:** Aplicar historias de usuario y criterios INVEST en el desarrollo
- **Autenticación y Autorización:** Implementar sistemas seguros con Bearer Tokens y control de roles
- **Testing:** Desarrollar pruebas automatizadas con el framework Pest

Proyecto de Práctica: Difexa API

Descripción del Sistema

Difexa API es un sistema backend orientado a la gestión y automatización de la difusión de información académica de la Facultad de Ciencias Exactas, Físicas y Naturales.

Características Principales

- **API REST** con arquitectura de microservicios
- **Autenticación** mediante Bearer Tokens (Laravel Sanctum)
- **Autorización** basada en roles con Spatie Laravel Permission
- **Integración con IA** para generación automática de contenido
- **Gestión de usuarios** con múltiples roles y permisos
- **Gestión de canales** temáticos de difusión
- **Publicaciones multimedia** con un flujo de aprobación

Roles del Sistema

- **Invitados:** Registro y solicitud de autorización
- **Registrados:** Usuarios pendientes de aprobación
- **Publicadores:** Creación y gestión de publicaciones
- **Administradores:** Gestión completa del sistema
- **Dispositivos:** Consumo de contenido para pantallas físicas

Stack Tecnológico

Backend Framework

- **Laravel 12** (última versión estable)
- **Eloquent ORM** para manejo de base de datos
- **Laravel Breeze** para autenticación base
- **Laravel Sanctum** para tokens de API

Autenticación y Autorización

- **Bearer Token Authentication**
- **Spatie Laravel Permission** para roles y permisos

- **Middleware personalizado** para control de acceso

Testing y Calidad

- **Pest** para testing automatizado
- **Mejores prácticas** de desarrollo
- **Code review** con GitHub Copilot

Base de Datos

- **SQLite** para desarrollo local
- **Migraciones** y **Seeders** de Laravel
- **Factory patterns** para datos de prueba

Herramientas de Desarrollo

- **Visual Studio Code**
- **GitHub Codespaces**

Requisitos para GitHub Copilot

- **Email institucional:** Solicitar a través del formulario: <https://forms.gle/vWNFnUUdF2t839yE7>
- **Acceso al webmail:** Una vez obtenido el email institucional, acceder a través de: <https://correo.unsj.edu.ar/roundcube/>
- **Configuración:** Seguir los pasos detallados en el documento [Pasos para obtener GitHub Copilot gratis como estudiante](#)

Contenido del Curso

Módulo 0: Sincronización de Conocimientos Previos

- **Transición de Python a PHP:** Tutorial de mapeo de conceptos y sintaxis
- **De SQL a Eloquent ORM:** Paralelismo entre consultas SQL tradicionales y el ORM de Laravel
- **Conceptos fundamentales de PHP:** Sintaxis, estructuras de datos y paradigmas
- **Introducción a la programación web:** Diferencias entre desarrollo de escritorio y web

Módulo 1: Fundamentos

- Configuración del entorno de desarrollo

- Introducción a GitHub Copilot y vibe-coding
- Creación de proyecto Laravel con Jetstream API
- Configuración de autenticación y roles

Módulo 2: Desarrollo de Microservicios

- Análisis de historias de usuario
- Generación de rutas y controladores con Copilot
- Implementación de CRUD operations
- Validaciones y reglas de negocio

Módulo 3: Autenticación y Autorización

- Sistema de roles y permisos
- Middleware de autorización
- Gestión de tokens de API
- Seguridad en APIs REST

Módulo 4: Testing y Documentación

- Pruebas unitarias con Pest
- Testing de APIs con autenticación
- Documentación automática de endpoints
- Validación de criterios de aceptación

Equipo de Cátedra

Profesor Responsable

Emilio Ormeño

Profesor titular de la materia Desarrollo Backend I

Equipo de Ayudantes

- **Martín Varela Ochoa** - Ayudante de cátedra
- **María Scheffer** - Ayudante de cátedra

Documentación Disponible

Este repositorio incluye la siguiente documentación técnica:

Documentos de Arquitectura

- [Microservicios](#) - Descripción de la arquitectura de microservicios
- [Descripción General del Software](#) - Visión completa del sistema Difexa API
- [difexa-api-arch](#) - Diagrama de arquitectura del sistema

Metodología y Desarrollo

- [Historias de Usuario para Difexa API](#) - Casos de uso completos del sistema
- [Prompt para historias de usuario](#) - Guía para crear historias de usuario con criterios INVEST

Guías Técnicas

- [Plantilla de Prompt para Generación de Rutas de Microservicios Laravel](#) - Template para generar código con Copilot
- [tutorial-laravel12-microservicios](#) - Tutorial paso a paso del setup inicial

Tutoriales de Transición (Próximamente)

- [mapeo-python-php.md](#) - Guía de transición de Python a PHP con ejemplos paralelos
- [sql-eloquent-orm.md](#) - Equivalencias entre consultas SQL y Eloquent ORM

Metodología de Trabajo

Vibe-Coding con GitHub Copilot

El curso enfatiza el uso de **vibe-coding**, una metodología que combina:

- **Desarrollo iterativo** guiado por IA
- **Historias de usuario** como punto de partida
- **Prompts estructurados** para generación de código
- **Review y refinamiento** continuo del código generado

Proceso de Desarrollo

1. **Nivelación** de conocimientos desde Python/SQL hacia PHP/Laravel
2. **Análisis** de historias de usuario siguiendo criterios INVEST
3. **Generación** de código base con GitHub Copilot
4. **Refinamiento** y adaptación del código generado
5. **Testing** automatizado de la funcionalidad
6. **Validación** contra criterios de aceptación

Requisitos Previos

Conocimientos de Base

- **Programación en Python** (nivel intermedio)
- **Bases de datos relacionales y SQL** (consultas básicas e intermedias)
- **Conceptos básicos de APIs REST**
- **Manejo básico de terminal/línea de comandos**
- **Git y GitHub** (básico)

Herramientas Necesarias

- PHP 8.2 o superior
- Composer
- Laravel Installer
- SQLite
- Editor de código con GitHub Copilot
- Cuenta en GitHub con acceso a Copilot

Evaluación

La evaluación del curso se basará en:

- **Desarrollo del proyecto Difexa API** (60%)
- **Uso efectivo de GitHub Copilot** (20%)
- **Testing y documentación** (10%)
- **Participación y metodología ágil** (10%)

Contacto y Soporte

Para consultas académicas y técnicas, contactar al equipo de cátedra a través del grupo de Whatsapp

Facultad de Ciencias Exactas, Físicas y Naturales

Universidad Nacional de San Juan

Año 2025