



HERNÁNDEZ GARCÍA ALFREDO

DESARROLLO DE SISTEMAS WEB

UNIDAD-2

ACTIVIDAD-2.2: LAS FASES DE DESARROLLO DEL BACKEND

ASESORA: URSULA DEL PILAR GONZÁLEZ ROBLES

# Propuesta de Fases de Desarrollo Backend (Metodología Scrum)

## Visión general del proyecto

### Objetivo:

Construir un backend escalable, modular, seguro y de alto rendimiento, basado en **microservicios**, con **Node.js**, **Express**, **MongoDB**, **Kafka**, y desplegado en contenedores **Docker** para la nube (AWS o Azure).

**Duración estimada:** 6 a 8 Sprints (de 2 semanas y media cada uno).

### Equipo Scrum:

- **Product Owner (PO):** Define prioridades y requerimientos funcionales.
- **Scrum Master:** Facilita la metodología y elimina impedimentos.
- **Equipo de desarrollo Backend:** 3-5 desarrolladores.
- **DevOps Engineer:** Encargado del despliegue, CI/CD y orquestación.

---

## Sprint 1 – Análisis, Configuración e Infraestructura Base

**Objetivo:** Establecer la base técnica y el entorno de desarrollo.

### Actividades:

- Definición del backlog.
- Configuración del entorno local y repositorios (Git).
- Definición de arquitectura en microservicios.
- Creación de API con Express.
- Configuración de bases de datos: MongoDB y Redis (con Docker).

### Entregables:

- Repositorio backend estructurado en GitHub.
- Entorno listo para desarrollo colaborativo.

---

## Sprint 2 – Módulo de Autenticación y Seguridad

**Objetivo:** Implementar el sistema de autenticación y autorización seguro.

### Actividades:

- Creación del microservicio de autenticación con Node.js/Express.

- Implementación en (Google, GitHub).
- Pruebas unitarias y de integración.

**Entregables:**

- API de autenticación funcional.
- 

## Sprint 3 – Módulo de Gestión de Usuarios y Perfiles

**Objetivo:** Desarrollar el microservicio de usuarios (empresas y desarrolladores).

**Actividades:**

- CRUD completo de usuarios y perfiles profesionales.
- Integración con MongoDB (colecciones dinámicas para perfiles).
- Subida y gestión de imágenes (foto de perfil, logo empresa).
- Búsqueda y filtrado de usuario por habilidades.

**Entregables:**

- Microservicio de usuarios funcional.
- 

## Sprint 4 – Módulo de Ofertas Laborales y Reclutamiento

**Objetivo:** Permitir la gestión integral de ofertas y procesos de contratación.

**Actividades:**

- CRUD de ofertas laborales.
- Asociación de ofertas a empresas y postulaciones de desarrolladores.
- Filtros y búsquedas por tecnologías y modalidad.
- Validaciones y pruebas de carga.

**Entregables:**

- Documentación y pruebas automatizadas.
-

## **Sprint 5 – Módulo de Notificaciones y Comunicación Asíncrona**

**Objetivo:** Implementar comunicación entre microservicios mediante mensajería.

**Actividades:**

- Configuración de Kafka.
- Envío y consumo de eventos (nueva oferta, nueva postulación, mensaje recibido).
- Implementación de colas y tópicos para procesamiento asíncrono.
- Pruebas de carga.

**Entregables:**

- Sistema de mensajería funcional entre microservicios.
  - Integración con módulos de usuario y ofertas.
- 

## **Sprint 6 – Módulo de Mensajería y Actualizaciones en Tiempo Real**

**Objetivo:** Habilitar la comunicación instantánea entre usuarios y alertas en tiempo real.

**Actividades:**

- Implementación de WebSockets con Socket.io.
- Creación del microservicio de mensajería (chat entre usuarios).

**Entregables:**

- Sistema de mensajería en tiempo real.
  - Notificaciones activas entre usuarios.
- 

## **Sprint 7 – Despliegue, Monitoreo y Escalabilidad**

**Objetivo:** Preparar la infraestructura para producción y garantizar la escalabilidad.

**Actividades:**

- Configuración de contenedores Docker por microservicio.
- Despliegue en la nube (AWS, Azure o GCP).
- Integración de logs centralizados (Elastic Stack, Grafana o Prometheus).
- Pruebas de estrés y balanceo de carga.

**Entregables:**

- Sistema desplegado en entorno de staging.
  - Escalado automático configurado.
  - Monitoreo y alertas operativas.
- 

## Sprint 8 – Optimización, Mantenimiento y Documentación Final

**Objetivo:** Asegurar mantenimiento, rendimiento y preparación para evolución futura.

**Actividades:**

- Refactorización y optimización de código.
- Mejora del rendimiento de consultas (indexación, caché).
- Documentación completa de APIs (Swagger/OpenAPI).
- Manual técnico y de despliegue.
- Revisión final de seguridad y cumplimiento de estándares.

**Entregables:**

- Backend optimizado, documentado y mantenable.
  - Manual de operación y API público.
- 

## Resultado Final

Al finalizar los sprints, el sistema backend estará compuesto por **microservicios independientes**, integrados y desplegados en la nube, con:

- Seguridad robusta (JWT, OAuth2, HTTPS).
- APIs RESTful documentadas.
- Comunicación asíncrona y tiempo real.