

Práctica Supervisada

- **Sigla:** CI-0156
- **Créditos:** 4
- **Horas lectivas:** 5 horas de práctica
- **Requisitos:** CI-0128 Proyecto Integrador de Ingeniería de Software y Bases de Datos, CI-0139 Administración de Proyectos, CI-0143 Seguridad de Sistemas Computacionales
- **Correquisitos:** CI-0142 Pruebas de Software
- **Ciclo de carrera:** II ciclo, 4to año (énfasis de Ingeniería de Software)
- **Docente:** Andrey Delgado Araya
- **Contacto:** andrey.delgado@ucr.ac.cr
- **Grupo:** 01
- **Semestre y año:** I Ciclo 2025
- **Horario de clases:** L 18-20:50 / J 19-20:50 Aula 303

Red Social Universitaria "UCR Connect"

Descripción General

"UCR Connect" será una red social exclusiva para la comunidad de la Universidad de Costa Rica, diseñada para fomentar la comunicación, colaboración y sentido de comunidad entre estudiantes, profesores y personal administrativo. La plataforma permitirá a los usuarios compartir contenido (texto, imágenes, GIFs), interactuar a través de comentarios, y mantenerse informados sobre eventos y noticias universitarias.

El proyecto incluirá:

- Una aplicación móvil para usuarios finales (estudiantes, profesores)
- Una plataforma web administrativa para moderación y gestión
- Un backend robusto que soporte ambas interfaces

Componentes Principales

Aplicación Móvil (Flutter)

- **Feed de noticias:** Visualización de publicaciones en orden cronológico o algorítmico
- **Creación de publicaciones:** Soporte para texto, imágenes y GIFs
- **Sistema de comentarios:** Comentarios de primer nivel en publicaciones
- **Perfil de usuario:** Información personal, historial de publicaciones
- **Notificaciones:** Alertas sobre interacciones y eventos relevantes
- **Configuraciones:** Preferencias de privacidad, notificaciones, temas

Portal Web Administrativo (React)

- **Dashboard:** Métricas y estadísticas de uso
- **Gestión de usuarios:** Visualización, búsqueda y gestión de cuentas
- **Moderación de contenido:** Revisión y acción sobre contenido reportado
- **Reportes analíticos:** Tendencias, engagement, crecimiento
- **Configuración del sistema:** Parámetros globales, reglas de moderación

Backend (Express/Node.js)

- **API RESTful:** Endpoints para todas las funcionalidades
- **Autenticación:** Integración con Firebase Authentication

- **Base de datos:** PostgreSQL para datos estructurados
- **Almacenamiento:** Firebase Storage para imágenes y medios
- **Notificaciones:** Firebase Cloud Messaging
- **Seguridad:** Validación, protección contra ataques

Roles y Responsabilidades por Equipo

Equipo Backend (5 miembros)

- **Arquitectura de datos:** Diseño de esquemas PostgreSQL optimizados
- **Desarrollo de API:** Implementación de endpoints RESTful
- **Integración con Firebase:** Autenticación, almacenamiento y notificaciones
- **Seguridad:** Implementación de JWT, validación de datos, protección contra ataques
- **Optimización:** Consultas eficientes, caché, manejo de concurrencia

Tareas específicas:

1. Diseño e implementación de esquemas de base de datos en PostgreSQL
2. Desarrollo de API RESTful con Express/Node.js
3. Integración con Firebase Authentication para gestión de usuarios
4. Configuración de Firebase Storage para almacenamiento de medios
5. Implementación de Firebase Cloud Messaging para notificaciones
6. Desarrollo de lógica de negocio para feed, publicaciones, comentarios
7. Implementación de seguridad y validación de datos
8. Creación de tests unitarios y de integración
9. Implementación de monitoreo con Sentry | Crashlytics

Equipo Web (5 miembros)

- **Dashboard administrativo:** Desarrollo de interfaz para administradores
- **Visualización de datos:** Gráficos, tablas y reportes
- **Herramientas de moderación:** Interfaces para gestión de contenido
- **Gestión de usuarios:** Herramientas para administrar cuentas
- **Configuración del sistema:** Interfaces para parámetros globales

Tareas específicas:

1. Desarrollo de dashboard con métricas en tiempo real
2. Implementación de sistema de gestión de usuarios (CRUD)
3. Creación de herramientas de moderación de contenido (CRUD)
4. Desarrollo de reportes analíticos con visualizaciones (Download)
5. Implementación de sistema de configuración global (Remote Config | Database)
6. Diseño e implementación de UI/UX para administradores
7. Integración con API backend
8. Implementación de tests para componentes React
9. Separación Componentes (Design System)

Equipo Mobile (6 miembros)

- **Interfaz de usuario:** Desarrollo de UI/UX en Flutter (Ideas con Mobbin.com)
- **Feed de noticias:** Visualización y actualización de contenido
- **Creación de contenido:** Herramientas para publicar texto, imágenes, GIFs
- **Interacción social:** Sistema de comentarios, likes, Shared
- **Perfil y configuraciones:** Gestión de información personal

Tareas específicas:

1. Desarrollo de interfaz de usuario con Flutter
2. Implementación de feed de noticias con scroll infinito
3. Creación de editor de publicaciones con soporte multimedia
4. Desarrollo de sistema de comentarios
5. Implementación de perfil de usuario y configuraciones
6. Integración con Firebase para autenticación
7. Configuración de notificaciones push
8. Implementación de caché local para uso offline
9. Creación de tests para la aplicación Flutter
10. Agregar Sentry | Cracslityc para gestión errores
11. Design System

Equipo DevOps (3 miembros)

- **Infraestructura:** Configuración de servidores y servicios cloud
- **CI/CD:** Implementación de pipelines automatizados
- **Monitoreo:** Configuración de herramientas de observabilidad
- **Seguridad:** Hardening de infraestructura, auditorías
- **Backups:** Estrategias de respaldo y recuperación
- **Seguridad:** Agregar Cloudflare para protección.

Tareas específicas:

1. Configuración de infraestructura en la nube (AWS/GCP/Azure o otro)
2. Implementación de CI/CD con GitHub Actions
3. Configuración de Docker para contenedores
4. Implementación de monitoreo con ELK Stack | Prometheus/Grafana
5. Configuración de backups automáticos para PostgreSQL
6. Implementación de seguridad a nivel de infraestructura (Terraform)
7. Configuración de entornos de desarrollo y (pruebas | producción)
8. Automatización de despliegues

Equipo Project Management (2 miembros)

- **Planificación:** Definición de sprints y roadmap
- **Seguimiento:** Control de avance y métricas
- **Coordinación:** Facilitación de comunicación entre equipos
- **Gestión de riesgos:** Identificación y mitigación
- **Documentación:** Mantenimiento de documentación del proyecto

Tareas específicas:

1. Definición y seguimiento de sprints
2. Facilitación de ceremonias ágiles (daily (Lunes/Jueves), planning (Cada inicio de Release), review (Cada semana), (retro / Final de mes))
3. Gestión de backlog de producto
4. Coordinación entre equipos
5. Comunicación con Product Owner
6. Documentación del proyecto (Markdown o selección)
7. Gestión de riesgos y dependencias

Equipo Quality Assurance (1 miembro)

- **Pruebas manuales:** Validación de funcionalidades
- **Automatización:** Desarrollo de tests automatizados (Postman o Script | Herramientas Cypress, Appwrite)

- **Reportes de bugs:** Documentación y seguimiento
- **Pruebas de usabilidad:** Validación de experiencia de usuario
- **Pruebas de seguridad:** Identificación de vulnerabilidades

Tareas específicas:

1. Diseño de casos de prueba
2. Ejecución de pruebas manuales
3. Desarrollo de pruebas automatizadas
4. Pruebas de integración entre sistemas
5. Validación de experiencia de usuario
6. Pruebas de seguridad básicas
7. Documentación y seguimiento de bugs
8. Verificación de correcciones

Tecnologías a Utilizar

Backend

- **Lenguaje:** JavaScript/TypeScript con Node.js
- **Framework:** Express.js
- **Base de datos:** PostgreSQL
- **ORM:** Sequelize o TypeORM o SQL Nativo
- **Autenticación:** Firebase Authentication + JWT
- **Almacenamiento:** Firebase Storage
- **Notificaciones:** Firebase Cloud Messaging
- **Otros servicios:** Firebase Remote Config, Firebase Analytics, Firebase Crashlytics, Hosting entre otros.
- **Testing:** Jest, Supertest (Pruebas software)

Frontend Web

- **Framework:** React
- **Gestión de estado:** Redux, Context API, TanStack Query, Zustand
- **UI Components:** Material-UI o Chakra UI o Tailwind
- **Gráficos:** Chart.js o D3.js
- **Formularios:** Formik o React Hook Form
- **Testing:** Jest, React Testing Library

Mobile

- **Framework:** Flutter
- **Gestión de estado:** Provider o Bloc
- **Almacenamiento local:** Hive y SharedPreferences
- **Networking:** Dio o http package
- **Testing:** Flutter Test, Bloc Test

DevOps

- **Control de versiones:** Git/GitHub
- **CI/CD:** GitHub Actions
- **Contenedores:** Docker
- **Orquestación:** Docker Compose o Kubernetes (básico)
- **Monitoreo:** ELK Stack o Prometheus/Grafana /
- **Cloud:** AWS o Google Cloud Platform o Azure o selección de grupo

Fases de Desarrollo (16 semanas)

Fase 1: Planificación y Configuración (Semanas 1-2)

- Definición detallada de requisitos
- Diseño de arquitectura técnica
- Configuración de entorno de desarrollo
- Configuración inicial de CI/CD
- Diseño de esquemas de base de datos
- Prototipos de UI/UX (web y mobile)

Fase 2: Desarrollo de Funcionalidades Core (Semanas 3-6)

- Implementación de autenticación con Firebase
- Desarrollo de API básica para usuarios y publicaciones
- Implementación de esquemas PostgreSQL
- Desarrollo de interfaz web administrativa básica
- Implementación de estructura básica de app móvil
- Configuración de Firebase Storage

Fase 3: Desarrollo de Funcionalidades Principales (Semanas 7-10)

- Implementación completa de CRUD para publicaciones
- Desarrollo de sistema de comentarios
- Implementación de feed de noticias
- Desarrollo de perfiles de usuario
- Implementación de notificaciones con Firebase
- Desarrollo de herramientas de moderación web

Fase 4: Integración y Funcionalidades Avanzadas (Semanas 11-13)

- Integración completa entre frontend y backend
- Implementación de reportes y analíticas
- Desarrollo de configuraciones de usuario
- Optimización de rendimiento
- Implementación de caché y estrategias offline
- Refinamiento de UI/UX

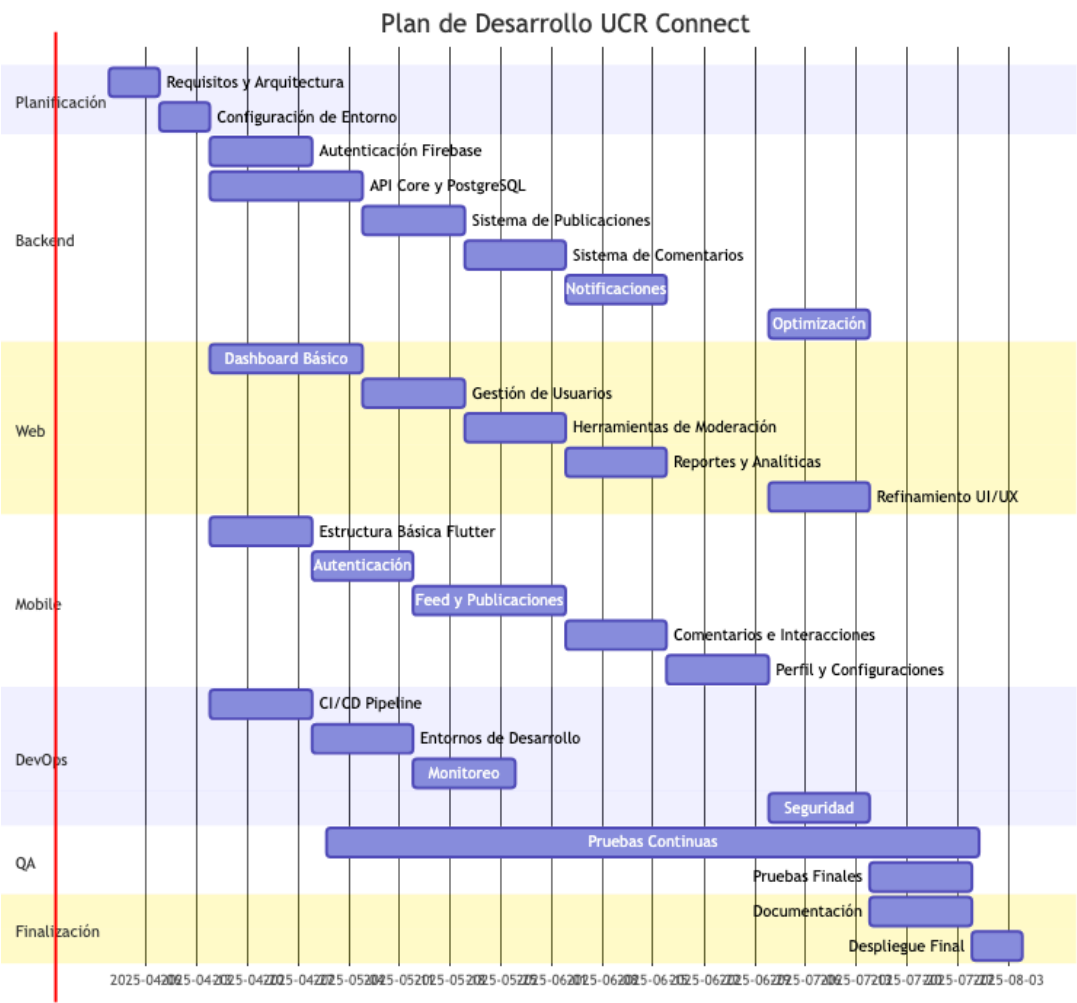
Fase 5: Pruebas y Optimización (Semanas 14-15)

- Pruebas de integración end-to-end
- Pruebas de carga y rendimiento
- Optimización de consultas a base de datos
- Corrección de bugs
- Pruebas de usabilidad
- Auditoría de seguridad

Fase 6: Finalización y Entrega (Semana 16)

- Documentación final
- Preparación de entorno de producción
- Despliegue final
- Presentación del proyecto
- Retrospectiva y lecciones aprendidas

Diagrama de Gantt Simplificado



Beneficios Educativos del Proyecto

- Experiencia con tecnologías modernas:** Trabajo con stack tecnológico actual y relevante
- Desarrollo full-stack:** Experiencia en frontend, backend y mobile
- Prácticas DevOps:** Exposición a CI/CD, contenedores y cloud
- Metodologías ágiles:** Experiencia práctica con Scrum/Kanban
- Trabajo en equipo:** Colaboración entre diferentes equipos técnicos
- Seguridad y privacidad:** Implementación de buenas prácticas
- Escalabilidad:** Diseño de sistemas para crecimiento futuro
- Portafolio profesional:** Proyecto completo para mostrar a futuros empleadores

Desafíos y Oportunidades de Aprendizaje

Desafíos Técnicos

- Rendimiento del feed:** Optimización de consultas y paginación
- Almacenamiento de medios:** Gestión eficiente de imágenes y GIFs

- **Notificaciones en tiempo real:** Implementación de sistema push
- **Experiencia offline:** Estrategias de caché y sincronización
- **Seguridad:** Protección contra inyección SQL, XSS, CSRF

Oportunidades de Aprendizaje

- **Arquitectura escalable:** Diseño para crecimiento futuro
- **Optimización de bases de datos:** Índices, consultas eficientes
- **UX/UI:** Diseño centrado en el usuario
- **Testing automatizado:** Cobertura completa de pruebas
- **Observabilidad:** Monitoreo y logging avanzados