

# **Propuesta de proyectos finales para el Diplomado de Desarrollo de Aplicaciones Móviles en IOS**

## Introducción

En el marco del **Diplomado en Desarrollo de Aplicaciones iOS**, se ha diseñado una serie de proyectos finales que constituyen la culminación de los conocimientos y habilidades adquiridos a lo largo del curso. Estos proyectos están orientados a proporcionar una experiencia práctica y aplicada, permitiendo a los estudiantes demostrar su competencia en el desarrollo de aplicaciones móviles para la plataforma iOS.

## Propósito de los Proyectos Finales

El propósito principal de estos proyectos es ofrecer a los estudiantes la oportunidad de integrar y aplicar de manera integral los conceptos teóricos y prácticos aprendidos durante el diplomado. A través de la creación de aplicaciones completas y funcionales, los estudiantes podrán enfrentar y resolver desafíos reales, desarrollando soluciones tecnológicas que respondan a necesidades específicas dentro del contexto universitario.

## Objetivos de los Proyectos

Los proyectos finales tienen como objetivos específicos:

1. **Aplicación de Conocimientos Técnicos:** Permitir a los estudiantes utilizar y consolidar sus conocimientos en Swift, UIKit, Storyboards, Auto Layout, y otras herramientas esenciales para el desarrollo de aplicaciones iOS.
2. **Desarrollo de Funcionalidades Avanzadas:** Implementar funcionalidades avanzadas como autenticación de usuarios, gestión de datos persistentes mediante UserDefaults o Core Data, consumo de APIs RESTful con URLSession, y manejo de errores de red.
3. **Diseño de Interfaces de Usuario:** Crear interfaces de usuario intuitivas y atractivas, siguiendo principios de UI/UX que aseguren una experiencia fluida y agradable para el usuario final.
4. **Gestión del Ciclo de Desarrollo:** Gestionar de manera eficiente todas las etapas del desarrollo de una aplicación, desde el diseño inicial hasta la implementación, pruebas y depuración.
5. **Fomento de la Innovación y Creatividad:** Estimular la creatividad de los estudiantes para proponer soluciones innovadoras que aporten valor a la comunidad universitaria.

## Expectativas y Resultados Esperados

Se espera que, al culminar estos proyectos, los estudiantes logren:

- **Demostrar Competencia Técnica:** Aplicar de manera efectiva las herramientas y tecnologías aprendidas para desarrollar aplicaciones robustas y funcionales.
- **Crear Interfaces de Usuario Eficientes:** Diseñar y desarrollar interfaces que sean no solo estéticamente agradables, sino también funcionales y fáciles de usar.
- **Implementar Soluciones Complejas:** Desarrollar funcionalidades avanzadas que respondan a necesidades específicas, mostrando capacidad de resolución de problemas.
- **Gestionar Proyectos de Manera Integral:** Coordinar todas las fases del desarrollo de una aplicación, desde la concepción hasta la finalización, siguiendo metodologías de trabajo eficientes.
- **Contribuir al Portafolio Profesional:** Generar proyectos de alta calidad que puedan ser incluidos en el portafolio profesional de los estudiantes, aumentando sus oportunidades en el mercado laboral.

## Proyecto 2: Aplicación de Gestión de Cursos



**Impacto Social:** Esta aplicación proporcionará a los estudiantes una plataforma accesible para obtener información detallada sobre los cursos disponibles, ayudándolos a planificar su trayectoria académica de manera efectiva. Esto no solo mejorará la experiencia educativa, sino que también fomentará un mejor rendimiento académico y permitirá a los estudiantes tomar decisiones informadas sobre su educación y futuro profesional. Al ofrecer una fuente confiable de información, la aplicación contribuirá a reducir la incertidumbre en la planificación académica.

**Objetivo:** Desarrollar una herramienta móvil que permita a los estudiantes explorar, informarse y gestionar los cursos universitarios disponibles. Esta aplicación ayudará a los estudiantes a evaluar los requisitos de cada curso, planificar sus horarios y decidir qué asignaturas tomar en función de sus intereses y objetivos académicos.

### **Diseño:**

Inicialmente, se deberá realizar el diseño de cada una de las pantallas utilizando un editor como, por ejemplo, Figma.

### **Requerimientos funcionales:**

#### **Autenticación de usuarios:**

- **Pantalla de inicio de sesión:**
  - Campo para ingresar el nombre de usuario.
  - Campo para ingresar la contraseña.
  - Botón de "Iniciar Sesión".
- **Proceso:**
  - Al hacer clic en "Iniciar Sesión", se envían las credenciales a un endpoint de autenticación.
  - Si las credenciales son correctas, el endpoint devolverá un token que se almacenará localmente para futuras solicitudes.
  - Manejar errores de autenticación mostrando mensajes apropiados.

#### **Onboarding:**

- **Pantallas introductorias:**
  - El onboarding debe mostrarse la primera vez que el usuario abra la aplicación.
  - Descripción breve de las principales funcionalidades.
  - Opción para omitir o finalizar el onboarding.

#### **Listado de cursos:**

- **Vista principal:**
  - Añadir una barra de búsqueda para facilitar la localización de cursos específicos.
  - Lista de cursos obtenida desde un endpoint.
  - Cada celda muestra:
    - Imagen del curso.
    - Nombre del curso.
    - Horario.

- Indicador visual si el curso ha sido marcado como favorito.

### Detalle de cursos:

- **Información detallada:**
  - Imagen del curso.
  - Descripción completa del curso.
  - Objetivos de aprendizaje.
  - Horario.
  - Requisitos o prerrequisitos.
  - Materiales o recursos (URLs).
  - Indicador visual si el curso ha sido marcado como favorito.

### Creación de cursos (para usuarios autorizados):

- **Formulario de creación:**
  - Campos para:
    - Imagen del curso.
    - Nombre del curso.
    - Descripción completa del curso.
    - Objetivos de aprendizaje.
    - Horario.
    - Requisitos o prerrequisitos.
    - Materiales o recursos (URLs).
- **Validación de campos obligatorios.**

### Actualización de cursos (para usuarios autorizados):

- **Formulario de actualización:**
  - Campos para editar:
    - Imagen del curso.
    - Nombre del curso.
    - Descripción completa del curso.
    - Objetivos de aprendizaje.
    - Horario.
    - Requisitos o prerrequisitos.
    - Materiales o recursos (URLs).
  - Botón para "Actualizar".
- **Proceso:**
  - Validar que los campos obligatorios estén completos.
  - Enviar datos al endpoint correspondiente para actualizar el curso.
  - Manejar respuestas y errores del servidor.



**Eliminación de cursos:**

- Los usuarios deben ser capaces de eliminar cursos (para usuarios autorizados).

**Marcar cursos como favoritos:**

- **Funcionalidad:**
  - Los usuarios pueden marcar cursos de su interés.
  - Utilizar un ícono (estrella o corazón) para indicar favoritos.
- **Persistencia:**
  - Almacenar los cursos favoritos localmente usando UserDefaults o Core Data.
  - Los datos almacenados deben permitir identificar y mostrar los cursos favoritos.

**Requerimientos técnicos:****Interfaz de usuario:**

- Uso de Storyboards y Auto Layout.
- Diseño propuesto por los estudiantes, enfocado en la usabilidad.

**Persistencia de datos:**

- Almacenar cursos favoritos usando UserDefaults o Core Data.

**Apartado de testing:**

- Definir pruebas unitarias para validar la funcionalidad de los componentes a nivel de código, asegurando que las clases y métodos funcionen como se espera.
- Definir pruebas de interfaz para garantizar que la interfaz de usuario responda correctamente a la interacción del usuario y que las pantallas y los flujos sean navegables y funcionales.

Course
+UUID id "Identificador único del curso"
+String name "Nombre del curso"
+String description "Descripción completa del curso, incluyendo contenido y objetivos generales"
+String learningObjectives "Objetivos de aprendizaje que los estudiantes deben alcanzar al finalizar el curso"
+String schedule "Horario detallado del curso, incluyendo días y horas de las sesiones"
+String prerequisites "Requisitos o prerrequisitos necesarios para inscribirse en el curso"
+List materials "Materiales o recursos adicionales, como enlaces a documentos, videos o páginas web relevantes"
+String imageURL "URL de la imagen representativa del curso para visualización en la aplicación"

## Documentación de Endpoints para la Aplicación de Gestión de Cursos Universitarios

<https://uam-server.up.railway.app/docs#/Course%20managements%20V1>

### Usuarios

#### 1. Login de Usuario

- **Método:** `POST`
- **Endpoint:** `/api/v1/user/login`
- **Parámetros:**
  - `email` (string, requerido): Correo electrónico del usuario.
  - `password` (string, requerido): Contraseña del usuario.
- **Respuesta:**
  - **200 OK:** Retorna un token de autenticación y la información del usuario.
  - **422 Unprocessable Entity:** Error en la validación de los datos.

#### 2. Registro de Usuario

- **Método:** `POST`
- **Endpoint:** `/api/v1/user/register`
- **Parámetros:**
  - `name` (string, requerido): Nombre del usuario.
  - `email` (string, requerido): Correo electrónico del usuario.
  - `password` (string, requerido): Contraseña del usuario.
- **Respuesta:**
  - **200 OK:** Retorna un token de autenticación y la información del usuario registrado.
  - **422 Unprocessable Entity:** Error en la validación de los datos.

#### 3. Obtener Usuario por ID

- **Método:** `GET`
- **Endpoint:** `/api/v1/user/{user_id}`
- **Parámetros:**
  - `user_id` (string, requerido): ID único del usuario.
- **Respuesta:**
  - **200 OK:** Retorna la información del usuario.
  - **422 Unprocessable Entity:** Error en la validación de los parámetros proporcionados.



## Subida de Imágenes

### 1. Subir Imagen

- **Método:** `POST`
- **Endpoint:** `/api/v1/upload_image`
- **Parámetros:**
  - `image` (archivo, requerido): Archivo de imagen a subir.
- **Respuesta:**
  - **200 OK:** URL de la imagen subida.
  - **422 Unprocessable Entity:** Error en la validación del archivo de imagen.

**Nota:** La propiedad `imageUrl`, utilizada en el recurso `course_management`, debe ser generada mediante este endpoint antes de crear o actualizar los cursos universitarios.

## Gestión de Cursos Universitarios

### 1. Obtener Gestión de Cursos

- **Método:** `GET`
- **Endpoint:** `/api/v1/course_management`
- **Parámetros:**
  - `search` (string, opcional): Filtro de búsqueda para obtener cursos específicos.
- **Respuesta:**
  - **200 OK:** Retorna una lista de cursos universitarios.
  - **422 Unprocessable Entity:** Error en la validación de los parámetros de búsqueda.

### 2. Crear Gestión de Curso

- **Método:** `POST`
- **Endpoint:** `/api/v1/course_management`
- **Parámetros:**
  - `name` (string, requerido): Nombre del curso.
  - `description` (string, requerido): Descripción del curso.
  - `learningObjectives` (string, requerido): Objetivos de aprendizaje del curso.
  - `schedule` (string, requerido): Horario en que se imparte el curso.

- **prerequisites** (string, requerido): Requisitos previos necesarios para el curso.
- **materials** (array de strings, requerido): Lista de materiales necesarios para el curso.
- **imageUrl** (string, requerido): URL de la imagen representativa del curso.  
**Nota:** Este campo debe generarse a través del endpoint de subida de imágenes (`/api/v1/upload_image`) antes de crear el curso.
- **Respuesta:**
  - **200 OK:** Retorna un objeto con la información del curso creado.
  - **422 Unprocessable Entity:** Error en la validación de los datos proporcionados.

### 3. Actualizar Gestión de Curso

- **Método:** PUT
- **Endpoint:** `/api/v1/course_management/{course_management_id}`
- **Parámetros:**
  - **course\_management\_id** (string, requerido): ID único del curso a actualizar.
  - Campos opcionales:
    - **name** (string): Nombre del curso.
    - **description** (string): Descripción del curso.
    - **learningObjectives** (string): Objetivos de aprendizaje.
    - **schedule** (string): Horario del curso.
    - **prerequisites** (string): Requisitos previos.
    - **materials** (array de strings): Materiales necesarios.
    - **imageUrl** (string): URL de la imagen del curso. **Nota:** Este campo debe generarse mediante el endpoint de subida de imágenes (`/api/v1/upload_image`) antes de actualizar el curso.
- **Respuesta:**
  - **200 OK:** Retorna un objeto con la información del curso actualizado.
  - **422 Unprocessable Entity:** Error en la validación de los datos.

### 4. Eliminar Gestión de Curso

- **Método:** DELETE
- **Endpoint:** `/api/v1/course_management/{course_management_id}`
- **Parámetros:**

- `course_management_id` (string, requerido): ID único del curso a eliminar.
- **Respuesta:**
  - **200 OK:** Confirmación de la eliminación del curso.
  - **422 Unprocessable Entity:** Error en la validación de los parámetros proporcionados.

**Nota General:** La propiedad `imageUrl` en el recurso `course_management` debe generarse usando el endpoint `/api/v1/upload_image` antes de la creación o actualización de los cursos universitarios.