

Enunciado del Proyecto Integrado

Aplicación Multiplataforma para un Rally Fotográfico

Contenido

1. Introducción	1
2. Objetivo del proyecto integrado	2
3. Estructura de la aplicación propuesta	2
4. Requisitos de implementación	3
5. Documentación requerida	4
6. Entrega del proyecto integrado	4
7. Defensa del proyecto integrado	4
8. Fechas de referencia	5
9. Rúbrica de calificación	5

1. Introducción

Un *rally fotográfico* es una actividad en la que los participantes deben capturar fotografías dentro de un marco temporal y temático definido. Estas imágenes son evaluadas en función de su calidad técnica, creatividad y cumplimiento de las directrices del rally. Para gestionar este tipo de eventos, se requiere una aplicación multiplataforma que permita la inscripción de participantes, la subida y validación de fotografías, la votación pública y la visualización de resultados.

Este proyecto integrado tiene como objetivo que el alumnado demuestre las competencias adquiridas en los módulos del ciclo formativo de Desarrollo de Aplicaciones Multiplataforma, combinando conocimientos de programación, bases de datos, desarrollo de interfaces, desarrollo cliente-servidor y despliegue de aplicaciones. La aplicación propuesta debe ser funcional y con aspecto profesional, cumpliendo con los requisitos técnicos y de usabilidad establecidos.

2. Objetivo del proyecto integrado

El objetivo principal es desarrollar una aplicación multiplataforma para la gestión de un rally fotográfico, que permita:

- La inscripción y gestión de participantes.
- La subida, validación y publicación de fotografías.
- La votación pública de las imágenes.
- La visualización de resultados en tiempo real.

El alumnado debe demostrar su capacidad para:

- Diseñar y desarrollar una aplicación multiplataforma completa.
- Gestionar bases de datos relacionales/NoSQL de manera eficiente.
- Implementar interfaces de usuario intuitivas y atractivas.
- Desplegar la aplicación en un entorno similar a producción.
- Documentar y justificar las decisiones técnicas tomadas durante el desarrollo.

3. Estructura de la aplicación propuesta

La aplicación debe contar con los siguientes módulos y funcionalidades:

3.1. Pantalla de Inicio (Home)

- Información sobre el rally fotográfico (bases del concurso, plazos, premios, etc.).
- Botón para registrarse como participante.
- Botón para hacer login como participante o administrador.
- Acceso a la galería de fotografías publicadas por los participantes.

3.2. Usuario administrador

- La aplicación tendrá usuarios (al menos uno) con rol administrador.
- Configuración de parámetros del rally (plazo de recepción de fotografías, límite de fotos por participante, plazo permitido de votación, etc.).
- Gestión de usuarios (baja y modificación).
- Validación de fotografías (estados: pendiente, admitida, rechazada).
- Visualización de estadísticas y resultados.

3.3. Participantes

- La aplicación permitirá darse de alta como participante del rally fotográfico.
- La solicitud de alta consiste en un formulario que pide algunos datos personales esenciales. Dicho formulario debe validar los datos introducidos.

- Los usuarios dados de alta como participantes del Rally, pueden acceder a su perfil y modificar datos.
- Subida de fotografías: los usuarios dados de alta como participantes del Rally, pueden subir un número determinado de fotos (establecido en las bases del concurso, configurado por el administrador).
 - El formulario de subida de fotos validará que cumplan determinadas características (tamaño máximo, formato, etc.) de acuerdo a las bases del concurso (a decidir por vosotros).
 - Las fotos subidas no se publican inmediatamente en la galería del concurso, deben ser previamente aceptadas como válidas por el administrador. Por tanto, pueden tener estados "pendiente", "admitida", "rechazada" o similares.
- Visualización y gestión de sus fotografías (estado, eliminación).

3.4. Público general

- En la página principal se podrá acceder a la solicitud de alta y a las bases del concurso. Pueden ambientarlo en el contexto o zona geográfica que quieran.
- Visualización de la galería de fotografías.
- Votación de fotografías (con límite de votos).
- Visualización de rankings y gráficos de votaciones.

4. Requisitos de implementación

- **Tecnologías:**
 - Frameworks multiplataforma, como Flutter o React Native, o lenguajes como Kotlin o Java para el desarrollo nativo en Android. **Queda excluido cualquier CMS o generadores automáticos de código.**
 - Base de datos: Sistemas relacionales como MySQL, PostgreSQL o MariaDB (no se permiten SQLite, Access, etc.) o NoSQL como Firebase Firestore, MongoDB, etc. debidamente argumentado.
 - Despliegue: archivo APK.
- **Funcionalidades mínimas:**
 - Operaciones CRUD para usuarios y fotografías.
 - Validación de datos en formularios.
 - Autenticación y control de acceso (roles de administrador y participante).
 - Despliegue en un entorno similar a producción.
- **Requisitos valorables:**
 - Mecanismo de votación.

- Uso de un sistema de control de versiones (Git) con repositorio remoto (GitHub, GitLab, etc.).
- Documentación clara y completa del proyecto.
- Código optimizado para producción.

5. Documentación requerida

La documentación debe incluir, como mínimo:

1. **Tecnologías utilizadas:** Listado de tecnologías, versiones y justificación de su elección.
2. **Diagramas:**
 - Diagrama de la base de datos: si es SQL, diagrama entidad-relación, si es NoSQL, diagrama de estructura de colecciones y documentos, diagrama de agregación o representación en JSON que refleje la organización de los datos.
 - Los diagramas de la arquitectura de la aplicación, de casos de uso, etc. que el alumnado considere.
3. **Desarrollo del proyecto:** Comentarios sobre decisiones técnicas, dificultades encontradas y soluciones implementadas.
4. **Manual de instalación:** Instrucciones completas para el despliegue autónomo de la aplicación. Esta documentación puede ser incluida, idealmente, en el fichero README de los repositorios, con secciones habituales: descripción / instalación / uso / créditos / licencia.
5. **Manual de usuario:** Instrucciones para el uso de la aplicación.
6. **Manual de administrador:** Instrucciones para la gestión de la aplicación.
7. **Consideraciones de ampliación futura:** Propuestas de mejora o funcionalidades adicionales.

6. Entrega del proyecto integrado

- **Código fuente:** Entregar el código fuente (.zip) y el enlace al repositorio remoto Git público.
- **Documentación:** Entregar en formato PDF.
- **Despliegue:** En la exposición y defensa del proyecto integrado, la aplicación debe estar desplegada y accesible para su demostración.

7. Defensa del proyecto integrado

- **Presentación:** Exposición del proyecto mediante un proyector, mostrando el funcionamiento de la aplicación.

- **Preguntas:** El alumnado deberá responder a preguntas sobre el código, las decisiones técnicas y la documentación.
- **Duración:** 20 minutos (15 minutos de presentación y 5 minutos de preguntas).

8. Fechas de referencia

- **Entrega del enunciado:** 10 de marzo de 2025.
- **Entrega del proyecto:** 13 de junio de 2025.
- **Defensa del proyecto:** 16-20 de junio de 2025.

Durante el periodo comprendido entre la entrega del enunciado y la entrega del proyecto, se llevará a cabo un seguimiento de la realización del proyecto integrado. Habrá un curso habilitado en la plataforma de Moodle Centros para centralizar las comunicaciones y entregas intermedias requeridas.

Se establecen 3 fases para el seguimiento de la realización del proyecto integrado:

- Fase 1: Definición del Proyecto
- Fase 2: Desarrollo Iterativo
- Fase 3: Cierre y Entrega

El detalle de los contenidos a abordar en cada fase se acompañará con sesiones presenciales en el centro que serán informadas a través del mencionado curso en Moodle Centros.

9. Rúbrica de calificación

La evaluación se realizará en base a los siguientes criterios:

Criterio	Descripción
Cumplimiento de requisitos mínimos	Implementación completa de las operaciones CRUD y funcionalidades básicas.
Diseño del modelo de datos	Correcto diseño e implementación de la base de datos (diagrama E/R, normalización o diagrama del modelo de datos para noSQL).
Interfaz de usuario	Diseño intuitivo, atractivo y profesional.
Funcionalidad y usabilidad	Correcto funcionamiento de la aplicación y experiencia de usuario satisfactoria.
Documentación	Completa, clara y con formato profesional.
Despliegue	Correcto despliegue en un entorno similar a producción.
Gestión del proyecto	Evidencia de desarrollo incremental (contribuciones en el repositorio Git).
Exposición y defensa	Claridad en la presentación y dominio del código y las decisiones técnicas.