

Enunciado Práctico Integrado 2025

Como práctico integrador se solicita la creación de un sistema de gestión de actividades deportivas para un gimnasio, donde se destacan dos componentes:

- ★ Backend, desarrollado en Golang, que brindará todas las interfaces necesarias para dar solución al requerimiento.
- ★ Frontend, desarrollado en React, que implementa las vistas finales del usuario y consumirá los servicios provistos por el backend.

Para la construcción del sistema, se solicitan los siguientes requerimientos:

Backend

1. **Autenticación de usuarios:** Desarrollar endpoints que permitan implementar un sistema de login que maneje la sesión del usuario mediante la generación de un token JWT. El token debe soportar 2 tipos de usuarios: socio y administrador; y debe expirar luego de un tiempo definido.
2. **Funcionalidades para socios:** Desarrollar endpoints que permitan la búsqueda de actividades disponibles, obtención por ID de una actividad, listado de actividades a las que el usuario se encuentra inscripto e inscripción en una actividad para los usuarios socios. Estos endpoints no requieren validación de permisos por token.
3. **Funcionalidades para administradores:** Desarrollar endpoints que permitan la creación, edición y eliminación de actividades deportivas. Las actividades deben tener día, horario, cupo y categoría (por ej. funcional, spinning, MMA, etc.). Estas operaciones deben validar permiso usando el token. Los administradores también tienen acceso a las funcionalidades de socio.
4. **Seguridad:** Además de garantizar la seguridad de las transacciones de escritura utilizando el token JWT firmado para las operaciones de creación, edición y eliminación de actividades deportivas; el sistema debe garantizar que las contraseñas no se almacenan de manera plana. Se les debe aplicar un mecanismo de hashing (MD5 o SHA256).

Frontend

1. **Vista de inicio (Home) y Búsqueda de actividades deportivas:** Debe mostrar un listado de las actividades deportivas disponibles con información mínima (título, horario y profesor). También debe implementar una barra de búsqueda de actividades deportivas por algún campo (palabra clave, horario, categoría).
2. **Vista de detalle de la actividad deportiva:** Debe mostrar información detallada sobre una actividad deportiva específica seleccionada, incluyendo descripción, foto, instructor, horario, duración, cupo, etc. Se accede a esta vista una vez que el usuario hace clic en uno de los resultados de búsqueda.
3. **Acción de inscripción en actividad deportiva:** Se debe habilitar un botón de inscripción para que el socio pueda registrarse en una clase de la actividad de su interés. Una vez que la operación se concreta, se debe mostrar el Congrats (registrado exitosamente o error al registrarse).
4. **Mis actividades deportivas:** Mostrar un listado de las actividades deportivas a las que el usuario se encuentra inscripto, con la opción de acceder al detalle de la actividad al hacer clic.
5. **Vistas de administradores:** Se debe implementar un formulario que permita la carga de una nueva actividad deportiva. Esta vista es sólo accesible para usuarios administradores. Contemplar todos los campos definidos para la actividad. Cuando un administrador accede al detalle de una actividad, debe contar con 2 nuevas opciones: editar la actividad o eliminarla. La primera lo llevará a un nuevo formulario y la segunda realizará la operación contra el backend directamente. En todos los casos se debe mostrar la congrats correspondiente.

Consideraciones Técnicas:

1. El código fuente debe ser manejado como repositorio y debe estar subido y disponible en Github, desarrollado por todos los integrantes del grupo.
2. Como fuente de datos se utilizará un esquema MySQL que debe soportar como mínimo 3 entidades principales: usuarios, actividades e inscripciones. Se pueden agregar más entidades si se considera necesario.
3. Las tablas de la base de datos estarán mapeadas a las estructuras del backend utilizando la librería GORM que se encargará de construir las consultas.
4. El backend debe implementar la estructura de MVC con los paquetes correspondientes para controladores, dominio y servicios.
5. Se requiere Dockerizar y componer la solución completa, incluyendo los 3 componentes (frontend, backend y base de datos).

Condiciones de Regularidad y Examen Final

- **Regularidad:** Para regularizar la materia se pide el desarrollo relacionado con los puntos 1 y 2 del backend, los puntos 1, 2 y 3 del frontend y los puntos 1, 2, 3 y 4 de las consideraciones técnicas.
- **Examen Final:** Para el examen final se solicita el desarrollo completo del sistema. Es decir, los puntos 3 y 4 del backend, los puntos 4 y 5 del frontend, y el punto 5 de las consideraciones técnicas.

Criterios de Evaluación

Categoría	Punto	Descripción	Entrega
Backend	Autenticación de usuarios	El sistema permite login y genera un token JWT con expiración y roles.	Primera Entrega
Backend	Seguridad (hash de contraseñas)	Las contraseñas se almacenan hasheadas con MD5 o SHA256.	Primera Entrega
Backend	Funcionalidad para socios	Se puede obtener el detalle de una actividad correctamente usando el ID.	Primera Entrega
Backend	Funcionalidad para socios	Se pueden listar y buscar sobre todas las actividades disponibles.	Primera Entrega
Backend	Funcionalidad para socios	Se puede listar las actividades a las que el socio está inscripto.	Final
Backend	Funcionalidad para socios	Se puede inscribir un socio a una actividad.	Primera Entrega
Backend	Funcionalidad para administradores	Se pueden crear nuevas actividades validando token de administrador.	Final
Backend	Funcionalidad para administradores	Se pueden editar actividades validando token de administrador.	Final
Backend	Funcionalidad para administradores	Se pueden eliminar actividades validando token de administrador.	Final
Backend	Seguridad (JWT en escritura)	Las operaciones de escritura están protegidas por el token JWT.	Final
Frontend	Vista de inicio	Se muestran actividades deportivas con título, horario y profesor.	Primera Entrega

Frontend	Búsqueda de actividades	Funciona una barra de búsqueda por palabra clave, categoría u horario.	Primera Entrega
Frontend	Detalle de actividad	Se muestra información completa (foto, cupo, duración, instructor, etc.).	Primera Entrega
Frontend	Inscripción a actividad	El botón de inscripción funciona y muestra mensaje de éxito o error.	Primera Entrega
Frontend	Mis actividades	Se listan correctamente las actividades a las que está inscripto el socio.	Final
Frontend (Admin)	Crear actividad	Formulario funcional para crear actividades (solo admins).	Final
Frontend (Admin)	Editar actividad	El administrador puede editar una actividad desde el detalle.	Final
Frontend (Admin)	Eliminar actividad	El administrador puede eliminar actividades desde el detalle.	Final
Base de Datos	Modelo de datos	Existe un esquema con las entidades: usuarios, actividades, inscripciones.	Primera Entrega
Base de Datos	ORM (GORM)	Las entidades están correctamente mapeadas usando GORM.	Primera Entrega
Estructura Código	Patrón MVC	El backend está organizado siguiendo el patrón MVC (controladores, etc.).	Primera Entrega
DevOps	Dockerización	Los servicios están dockerizados correctamente (frontend, backend, DB).	Final
DevOps	Docker Compose	Funciona una solución completa con Docker Compose.	Final
Control de versiones	Repositorio en Github	Todo el proyecto está subido a Github y correctamente versionado.	Primera Entrega