

Grupo Clouds

Desarrollo de Aplicaciones para Dispositivos Móviles 2° D

Prof. Kevin Axel Del Bello

Equipo de trabajo

- ★ Eduardo Moreno
- ★ Leandro Paryszewski
- ★ Marcelo Moreno
- ★ Melissa Galeano Ibañez

Links de Proyecto

 Figma: [Figma](#)
 Sistema Original: [DSOO](#) / [MDS](#)

✓ RECOPIACIÓN DE TAREAS REALIZADAS EN ETAPA PREVIA

Las siguientes tareas y artefactos fueron producidos durante las etapas de análisis y diseño del sistema original:

- **Definición de Requerimientos del Cliente:**
 - Establecimiento del objetivo principal: Automatizar las operaciones básicas del club para mejorar la administración y la experiencia del usuario.
 - Identificación de funcionalidades esenciales:
 - Registro y gestión de personas (socios y no socios).
 - Gestión de administradores con autenticación.
 - Control de actividades deportivas ofrecidas.
 - Seguimiento de cuotas sociales.
 - Emisión de carnets de socio.
- **Diseño de la Arquitectura y Estrategia de Solución:**
 - Propuesta de una arquitectura basada en capas lógicas: Capa de Presentación (formularios), Capa de Lógica de Negocio (clases del modelo) y Capa de Acceso a Datos (conexión directa a base de datos).
 - Selección de tecnologías: C# para el desarrollo del sistema y MySQL como sistema de gestión de bases de datos.
- **Modelado del Dominio y Base de Datos:**
 - **Identificación de Entidades y Atributos:** Se definieron las variables clave del sistema, incluyendo identificadores de personas, estados de membresía, fechas, montos y descripciones de actividades.

- **Diagrama de Clases (Modelo Entidad-Relación):** Creación de un modelo que detalla las entidades principales (Persona, Socio, NoSocio, Administrador, Actividad, Cuota, PagoActividad) y sus relaciones, atributos y tipos de datos.
- **Diseño de Base de Datos:** La base de datos MySQL fue diseñada con tablas normalizadas y relaciones claras entre entidades. Se menciona la existencia de un script para su creación y precarga de datos.
- **Modelado del Comportamiento del Sistema:**
 - **Diagramas de Casos de Uso:** Identificación de los actores principales (Cliente, Administración) y los casos de uso fundamentales del sistema (Registrar como socio, Pagar cuota o actividad, Entregar carnet, Mirar listas de cuotas vencidas), junto con sus relaciones.
 - **Especificación de Casos de Uso:** Detalle exhaustivo para casos de uso críticos, como "Registrar Cliente", incluyendo pre-condiciones, post-condiciones, flujos normales, flujos alternativos y manejo de excepciones.
 - **Diagramas de Secuencia:** Modelado de las interacciones temporales entre los actores y los componentes del sistema para procesos específicos, como el registro de un cliente.
 - **Diagramas de Actividades:** Representación del flujo de control para la inscripción de un cliente, ilustrando decisiones y acciones resultantes.
- **Diseño de la Interfaz de Usuario (UI):**
 - Diseño conceptual de formularios para las principales funcionalidades del sistema, abarcando desde el inicio de sesión hasta la gestión de actividades y cuotas.
- **Juegos de Prueba Sencillos:**
 - Definición de casos de prueba básicos para funcionalidades clave como el inicio de sesión, el registro de personas y la emisión de carnets, especificando entradas y resultados esperados.
- **Codificación y Estructura del Proyecto:**
 - Implementación del sistema en C# con Windows Forms. El proyecto incluye archivos de clases para el modelo y la configuración de la base de datos MySQL.
- **Conclusiones del Desarrollo:**
 - Identificación de áreas de mejora como la ausencia de una capa DAL separada, la necesidad de validaciones de datos más robustas y problemas en el cierre de conexiones a la base de datos en caso de error.

✓ REVISIÓN CRÍTICA Y PUNTOS DE MEJORA

La revisión del material existente revela varios puntos críticos y áreas de oportunidad para la migración a una aplicación Android en Kotlin:

- **Experiencia de Usuario (UI/UX):**

- **Punto Débil:** La interfaz está diseñada para Windows Forms, un entorno de escritorio. Las convenciones de diseño, la navegación y el tamaño de los componentes no son adecuados para dispositivos móviles.
 - **Impacto en Mobile:** Se requiere un rediseño completo de la UI/UX para adoptar los principios de diseño de Android (Material Design), optimizar para pantallas táctiles, tamaños de pantalla variados y patrones de navegación móvil.
 - **Manejo de Errores y Robustez:**
 - **Punto Débil:** Las conclusiones señalan que la "validación de datos podría ser más robusta" y que "la conexión a la base de datos no se cierra correctamente si ocurre un error".
 - **Impacto en Mobile:** En un entorno con conectividad variable y llamadas a servicios remotos, el manejo robusto de errores (de red, de servidor, de validación) y la provisión de feedback claro al usuario son fundamentales para una aplicación estable y confiable.
 - **Consideraciones Específicas de Desarrollo Móvil:**
 - **Punto Débil:** El diseño existente no contempla aspectos intrínsecos al desarrollo móvil como la gestión de la conectividad (modo offline), notificaciones push, permisos del dispositivo (cámara, ubicación), optimización del consumo de batería o accesibilidad.
 - **Impacto en Mobile:** Estos elementos son clave para ofrecer una aplicación completa e integrada en el ecosistema Android, y deben ser incorporados desde la fase de análisis.
-

CAMBIOS A REALIZAR PARA EL DESARROLLO

Para una migración exitosa y el desarrollo de una aplicación Android moderna y funcional en Kotlin, se introducirán los siguientes cambios específicos:

- **Rediseño Completo de la Interfaz de Usuario (UI/UX) para Android:**
 - **Adopción de Material Design:** Todas las pantallas y flujos de usuario serán rediseñados, utilizando componentes nativos de Android, tipografía y paleta de colores adecuadas.
 - **Navegación Intuitiva Móvil:** Implementar patrones de navegación estándar de Android (ej. Navigation Components, BottomNavigationView, DrawerLayout) para asegurar una experiencia de usuario fluida e intuitiva.
 - **Diseño Responsivo:** Garantizar que la interfaz se adapte de forma óptima a los diferentes tamaños, resoluciones y orientaciones de pantalla de los dispositivos Android.
 - **Optimización de Flujos:** Simplificar y optimizar los flujos de usuario existentes (login, registro, pagos, etc.) para la interacción táctil y las limitaciones de una pantalla móvil. Por ejemplo, la configuración inicial de la base de datos se trasladará al backend.

- **Manejo Avanzado de Datos y Errores:**
 - **Validación de Datos en Ambos Lados:** Implementar validación de datos tanto en el cliente móvil (para retroalimentación inmediata al usuario) como, de forma crítica, en el backend (para asegurar la integridad y consistencia de los datos).
 - **Gestión de la Conectividad de Red:** Integrar lógica en la aplicación Kotlin para detectar el estado de la red (online/offline) y manejar proactivamente los errores de conectividad, informando al usuario y ofreciendo una experiencia robusta incluso con interrupciones de red.
 - **Retroalimentación al Usuario:** Proporcionar indicadores de carga claros para operaciones asíncronas, mensajes de error explícitos y concisos, y notificaciones de éxito/fracaso de las operaciones.
- **Integración de Funcionalidades Nativas de Android:**
 - **Notificaciones Push:** se podrían integrar para enviar notificaciones importantes (ej. cuotas vencidas, recordatorios de actividades) a los usuarios del club.
 - **Gestión de Permisos:** Identificar y solicitar los permisos de Android necesarios (ej. cámara para escanear carnets, ubicación si fuera relevante) de forma transparente y en el momento adecuado.
- **Estrategia de Pruebas Integral:**
 - **Pruebas Unitarias:** Implementar pruebas unitarias extensivas para la lógica de negocio en el backend y para los ViewModel/presenters en la aplicación Android..
 - **Pruebas de UI/Instrumentación:** Crear pruebas de interfaz de usuario para validar la funcionalidad y los flujos de la aplicación Android en distintos dispositivos.
- **Actualización de la Documentación:**
 - Crear una nueva documentación técnica detallada del backend y un "Manual de Usuario" específico para la aplicación móvil.

Plantilla de Planificación	
Actividad	Descripción
Entorno de Estudio	Desarrollo en Android Studio con lenguaje Kotlin, base de datos MySQL. Se parte de un sistema previo implementado en C#, el cual se migrará y se adaptará a una aplicación móvil.
Objetivo del Proyecto	Crear una aplicación móvil para gestionar las operaciones básicas de un club deportivo: Registro y gestión de personas (socios y no socios) ,gestión de administradores con autenticación, control de actividades deportivas ofrecidas por el club, seguimiento de cuotas sociales y emisión de carnets de socio.
Identificar Procesos	<ul style="list-style-type: none"> - Registro y autenticación de administradores. - Alta y gestión de socios / no socios. - Inscripción y gestión de actividades.

Plantilla de Planificación	
Actividad	Descripción
	<ul style="list-style-type: none"> - Pago y control de cuotas sociales. - Generación de credenciales/carnets digitales.
Análisis de Factibilidad(O,T,E)	
Operativa	Los procesos actuales del club (login, registro, cuotas, actividades) se adaptan fácilmente a la aplicación, facilitando la gestión y reduciendo errores.
Técnica	Se usará Android Studio con Kotlin, base MySQL vía API REST y se cuenta con el código del sistema previo.
Económica	El desarrollo se realiza con herramientas gratuitas (Android Studio, MySQL Community Edition, GitHub). Los costos se limitan al tiempo de desarrollo y posibles gastos de hosting de la base de datos en la nube.
Gestión de Stakeholders	<ul style="list-style-type: none"> - Directivos del club: interesados en los reportes y control general. - Administradores del club: quienes gestionan a las personas, cuotas y actividades. - Equipo de desarrollo: encargado de la implementación y las pruebas. - Socios: usuarios registrados que abonan de forma mensual. - No socios: usuarios sin registrar que abonan por actividad. - Empleados: profesores, nutricionistas, médicos.
Roles	<ul style="list-style-type: none"> - Product Owner: Representante del club, quien define los requerimientos. - Desarrollador: Implementa la aplicación en Android Studio. - Diseñador UI/UX: Define la interfaz y la experiencia de usuario. - Tester: Realiza pruebas y controla la calidad de la aplicación. - Administrador de base de datos: Gestiona y mantiene la base de datos.

Historia de Usuario 1: Registrar Persona

ID	Rol	Características / Funcionalidad	Razón / Resultado	Nº de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado
1	Recepcionista	Registrar a una persona como socio o no socio	Tener un registro único de socios/no socios	1	Todos los campos obligatorios completados (nombre, apellido, DNI, fecha nacimiento)	Cuando un ciudadano se presenta para inscripción	Completar el formulario y guardar	El sistema registra correctamente al socio/no socio

				2	DNI ingresado ya existe en la base de datos	Al intentar guardar un nuevo registro	Validar DNI en sistema	Se muestra advertencia "DNI ya registrado", no se guarda duplicado
				3	Campos obligatorios incompletos	Cuando el recepcionista intenta guardar con datos faltantes	Validación de formulario	El sistema alerta "Debe completar los campos obligatorios"
				4	Ficha médica no presentada (campo no marcado)	Al guardar el registro de socio	Validación de campo de ficha médica	Se guarda registro, pero el sistema marca estado "Ficha médica pendiente"
				5	Registro cancelado por usuario	Cuando el recepcionista decide cerrar formulario sin guardar	Botón "Cerrar" en pantalla	El sistema no guarda nada y vuelve al menú principal

Historia de Usuario 2: Pagar Cuota o Actividad

ID	Rol	Características / Funcionalidad	Razón / Resultado	Nº de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado
2	Recepcionista	Registrar el pago de cuota mensual (socio) o actividad (no socio)	Mantener regularidad de pagos y acceso al club	1	DNI válido de socio/no socio existente	El socio/no socio se presenta a abonar	Ingreso de DNI, monto, medio de pago, fecha	Pago registrado y comprobante generado

				2	DNI no registrado en el sistema	Cuando se ingresa DNI inválido	Validación contra base de datos	Mensaje de error "El DNI no existe en el sistema"
				3	Pago con datos incompletos (sin monto o medio de pago)	Al intentar registrar pago con campos vacíos	Validación de formulario	Sistema muestra error "Complete todos los campos"
				4	Pago rechazado por error del sistema o medio de pago inválido	Cuando se intenta procesar el pago	Validación de transacción	Mensaje de error "No se pudo procesar el pago"
				5	Cancelación de operación por el recepcionista	Antes de confirmar el pago	Botón "Cerrar" en formulario	Se cancela operación y no se guarda registro
				6	Registro de pago diario (no socio)	Cuando un no socio paga una actividad específica	Ingreso de datos + confirmación	El pago queda registrado con fecha y actividad asociada

Historia de Usuario 3: Listar Cuotas Vencidas

ID	Rol	Características / Funcionalidad	Razón / Resultado	Nº de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado
3	Recepcionista / Sistema	Generar un listado de socios con cuotas vencidas	Controlar acceso y seguimiento de pagos	1	Existen socios con cuotas vencidas	Se consulta el listado	Selección de opción "Ver vencidas"	Sistema genera listado con datos correctos

				2	No existen socios con cuotas vencidas	Al ejecutar búsqueda	Selección de opción "Ver vencidas"	El sistema muestra "No hay cuotas vencidas"
				3	Listar cuotas vigentes	Cuando se quiere ver socios al día	Selección de opción "No vencidas"	El sistema lista todos los socios con cuotas activas
				4	Listar cuotas que vencen hoy	Cuando se consulta el estado diario	Selección "Vence hoy"	Sistema muestra listado de socios cuya cuota vence en la fecha
				5	Filtrar por DNI	El recepcionista ingresa un DNI específico	Opción "Filtrar"	Se muestran solo los datos de ese socio (si existen)
				6	Cancelar la consulta	Recepcionista decide salir	Botón "Cerrar"	Se regresa al menú principal sin mostrar resultados

Historia de Usuario 4: Control de Actividades Deportivas

ID	Rol	Características / Funcionalidad	Razón / Resultado	Nº de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado
4	Recepcionista /	Registrar una nueva actividad	Ofrecer actividades	1	Todos los campos completos y válidos	Al crear una nueva actividad	Ingresar datos y presionar	Actividad registrada y

	Administrador	deportiva (nombre, tipo, profesor, horario, capacidad, costo)	organizadas y disponibles				r "Guardar"	disponible en el sistema
				2	Campos incompletos o inválidos	Cuando se intenta guardar sin datos completos	Validación de formulario	Mensaje de error indicando campos faltantes
				3	Edición de actividad existente	Cuando el administrador actualiza datos	Modificar y guardar	Actividad actualizada correctamente
				4	Eliminación de actividad	Cuando se decide borrar una actividad	Confirmar eliminación	La actividad se elimina de la base de datos
				5	Listado de todas las actividades	Cuando se consulta el listado general	Selección de "Ver todas"	El sistema muestra todas las actividades con sus datos
				6	Búsqueda de actividad específica	Cuando se filtra por nombre, tipo o profesor	Aplicar filtros	El sistema devuelve solo las actividades coincidentes

Historia de Usuario 5: Emisión de Carnets de Socio

ID	Rol	Características / Funcionalidad	Razón / Resultado	Nº de Escenario	Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado
----	-----	---------------------------------	-------------------	-----------------	------------------------	----------	--------	-------------------------------------

5	Recepcio nista	Emitir carnet para un socio registrado	Identificar al socio y permitir acceso a actividades	1	DNI válido de socio registrado	Al ingresar DNI correcto	Presion ar botón “Entrega r”	Carnet emitido y marcado como entregado
				2	DNI inexistente o incorrecto	Cuando se ingresa DNI no válido	Validaci ón en sistema	Mensaje de error “Socio no encontrado”
				3	Carnet ya emitido previamente	Cuando se intenta emitir carnet duplicado	Validaci ón de estado de carnet	Mensaje “El carnet ya fue emitido”
				4	Emisión exitosa	Al confirmar emisión con datos correctos	Presion ar “Entrega r”	El sistema actualiza estado de carnet a “Entregado”
				5	Cancelación de emisión	Recepcionist a decide no emitir	Presion ar “Salir” o cerrar formular io	No se emite carnet y se vuelve al menú principal