

Evaluación Final Transversal

Encargo

Instrucciones y pauta de evaluación

Estudiante

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
DSY1105	Desarrollo de Aplicaciones Móviles	12 semanas	20% del EFT

1. Situación Evaluativa 1:

X	Entrega de encargo
---	--------------------

2. Instrucciones generales

Descripción
<ul style="list-style-type: none">■ La Evaluación Final Transversal corresponde a la entrega y defensa técnica de una aplicación móvil completamente funcional, desarrollada a partir de un caso o contexto real y/o simulado definido por el equipo de estudiantes. Deberá demostrar criterios de calidad, coherencia visual, arquitectura técnica y funcionamiento completo, tanto en el frontend móvil como en su integración con servicios externos y backend.■ Este producto final es el resultado de un proceso progresivo que comenzó en la Evaluación Parcial 2, centrada en el diseño y planificación técnica del proyecto, y continuó en la Evaluación Parcial 3, dedicada al desarrollo funcional e integración de componentes clave. En esta fase final, se espera que el proyecto refleje un dominio integral de las competencias adquiridas.■ El equipo deberá presentar una solución coherente, técnicamente sólida, funcional al momento de la evaluación, y que permita además realizar pruebas, ajustes y demostraciones en tiempo real.

- Esta evaluación consiste en **una entrega de encargo que medirá los siguientes Indicadores de Logro:**
- *IL1.3 Desarrolla un programa orientado a objetos, respetando el patrón arquitectónico y utilizando principios de clases, herencia y polimorfismo, incluyendo el manejo de excepciones y verificando su correcto funcionamiento en un entorno de pruebas.*
- *IL 2.3 Integra almacenamiento local y organiza la aplicación móvil usando patrones arquitectónicos y herramientas colaborativas para el desarrollo continuo de componentes visuales y funcionales, aportando al avance del proyecto y favoreciendo la mantenibilidad del código.*
- *IL2.4 Implementa funciones de acceso a recursos nativos del dispositivo móvil, garantizando su correcto funcionamiento y su integración segura en la aplicación*
- *IL3.1 Desarrolla una aplicación móvil funcional e integrada, conectando componentes de front-end y back-end mediante microservicios para garantizar coherencia y continuidad.*
- *IL 3.2 Integra pruebas unitarias para validar el comportamiento esperado asegurando el correcto funcionamiento e integración segura.*
- *IL3.3 Presenta la aplicación para su publicación mediante la generación del ejecutable firmado digitalmente, documentando el proceso con herramientas de planificación y control de versiones, en colaboración con el equipo.*

Aspectos formales:

- El proyecto se desarrolla en parejas durante 12 semanas, combinando trabajo autónomo con sesiones en el taller de alto cómputo. El encargo se asigna en la semana 6 como parte de la Evaluación Parcial 2, y la entrega final debe realizarse en la semana 18, previo al inicio del período de defensas técnicas.
- La defensa es individual, con una duración máxima de 15 minutos por estudiante. El orden de presentación será aleatorio, por lo que todos deben estar preparados desde el inicio del período evaluativo.
- Aunque el desarrollo es colaborativo (parejas), la evaluación de la defensa/demostración es individual. Cada estudiante será calificado según su claridad, dominio del proyecto y capacidad para justificar decisiones, ejecutar la aplicación y modificar el código durante la defensa.
- Si un/a estudiante no logra demostrar comprensión y control sobre el desarrollo, se asumirá que no participó activamente, lo que afectará directamente su nota, sin considerar el desempeño de su compañero/a ni la calidad general del proyecto.

Atención: En caso de que el proyecto no se ejecute correctamente al momento de la defensa o presentación, no será posible evaluar los ítems posteriores al error. Por lo tanto, dichos ítems serán calificados con puntaje cero, ya que no se cumplirán los requerimientos mínimos necesarios para realizar pruebas, demostraciones o modificaciones al código.

Requisitos Mínimos del Proyecto

Para ser evaluado, todo proyecto deberá cumplir con los siguientes criterios básicos y obligatorios:

- **Tema definido y contextualizado:**
 - El proyecto debe responder a una necesidad concreta (real o simulada). Cada equipo deberá seleccionar un tema de su interés, aprobado por todos sus integrantes, idealmente relacionado con un contexto real (emprendimiento familiar, idea propia, casos comunitarios u otros).
 - Debe ser posible de explicar, justificar y modelar desde el punto de vista funcional y técnico.
 - El tema no puede ser idéntico entre equipos, aunque puede pertenecer a una misma categoría general (ventas, foros, salud, etc.).
- **Integraciones obligatorias:**
 - Consumo de al menos una API externa pública (por ejemplo: REST Countries, OpenWeather, etc.).
 - Conexión con un backend propio desarrollado por el equipo mediante microservicios funcionales, incluyendo almacenamiento de datos, operaciones de persistencia de datos tanto locales como externos (CRUD) y endpoints correctamente expuestos.
 - Acceso al menos a dos recursos nativos.
- **Componentes técnicos adicionales:**
 - Pruebas unitarias que validen una parte de la lógica del frontend.
 - Generación del APK firmado, apto para distribución o pruebas en dispositivos reales.
- **Autenticidad y autoría del proyecto:**
 - El equipo debe demostrar evidencia de trabajo propio (historial de GitHub, Trello, commits y defensas individuales).
 - No se aceptarán proyectos armados por terceros o reutilizados sin modificación sustantiva.
- **Requisitos Técnicos Mínimos Obligatorios:**
 - Al menos 4 roles de usuario con privilegios diferenciados.
 - Formularios internos funcionales.
 - Personalización visual: colores, logos, nombre e imágenes propios del equipo.
 - Inicio de sesión, registro de usuarios, recuperación de contraseña y modificación de perfil como parte obligatoria de los requerimientos del contexto.
 - Todas las pantallas funcionales y con navegación fluida que respondan al contexto o reglas de negocios del mismo.
 - Almacenamiento local y externo (base de datos de los microservicios) bien diferenciados y justificados.
 - Gestión de estado desacoplada.
 - Animaciones funcionales y transiciones suaves durante la navegación o cambios de estado.

Instrucciones específicas de la Evaluación

Ítem I: Consideraciones para el encargo:

Cada equipo (2 personas) deberá entregar un proyecto de aplicación móvil, conectado a microservicios propios, con integración completa de funcionalidades.

El entregable debe incluir los siguientes **apartados**:

- **Requisitos obligatorios del proyecto (entregados en GitHub y AVA)**
- Diseña y entrega una interfaz visual completa y funcional: Tu app debe incluir todas las pantallas y formularios definidos, sin errores de navegación, ejecución ni validación visual.
- Formularios validados con íconos y mensajes visuales: Se verificará estén disponibles todos los formularios y que cada campo tenga retroalimentación visual ante errores.
- Validaciones manejadas desde lógica: Se revisará que la lógica no esté acoplada al componente visual.
- Animaciones funcionales: Debe haber animaciones que aporten fluidez o retroalimentación.
- Proyecto con estructura modular y persistencia local: Se revisará organización de carpetas, MVVM, y persistencia local funcionando.
- Repositorio en GitHub + planificación en Trello: Debe mostrarse actividad real en commits y tareas compartidas.
- Acceso a al menos dos recursos nativos: Se validará funcionamiento e integración en la UI.
- Construye y entrega todos los microservicios requeridos.
- Cada microservicio debe estar correctamente programado, con base de datos activa y endpoints funcionales.
- Integra tu app móvil con los microservicios desarrollados.
- Debe ser posible enviar, recibir y actualizar datos en tiempo real, cubriendo operaciones CRUD desde la interfaz.
- Incorpora una API externa al flujo de la aplicación.
- Debe consumirse vía Retrofit y mostrarse en la interfaz, sin interferir con los datos locales ni microservicios propios.
- Incluye pruebas unitarias que cubran al menos el 80% de la lógica.
- Estas pruebas deben estar implementadas en módulos como ViewModel o Repository, usando JUnit, MockK o Kotest.
- Genera el archivo APK firmado.
- Debes firmarlo correctamente con un .jks, y dejarlo funcional en el repositorio junto a su configuración.

Forma de entrega (obligatorio)

A. En GitHub (público):

- Carpeta del proyecto (código fuente).
- Archivo README.md con:
 - Nombre de la app.
 - Nombres de los integrantes.
 - Funcionalidades.
 - Endpoints usados (propios y externos).
 - Instrucciones para ejecutar el proyecto.
 - APK firmado y ubicación del archivo .jks.
 - Código fuente de microservicios y app móvil.
 - Evidencia de trabajo colaborativo (commits por persona).

B. En AVA (2 pasos):

- Subida del proyecto grupal (un solo integrante lo sube).
- Formulario individual: cada estudiante debe escribir su nombre completo, número de equipo y nombre del proyecto.
- Si el formulario individual NO se completa en el plazo, el estudiante NO podrá defender el proyecto y obtendrá 1.0 como nota individual.
 - **Penalizaciones:**
- Cambios al repositorio después de la fecha límite: 1.0 automático grupal e individual.
- No entrega en AVA: 1.0 automático sin excepción.
- Proyecto que no ejecuta: ítems posteriores no serán evaluados y obtendrán 0.

3. Pauta de Evaluación

Categoría	% logro	Descripción niveles de logro
Muy buen desempeño	100%	Demuestra un desempeño destacado, evidenciando el logro de todos los aspectos evaluados en el indicador.
Desempeño aceptable	60%	Demuestra un desempeño competente, evidenciando el logro de los elementos básicos del indicador, pero con omisiones, dificultades o errores.
Desempeño incipiente	30%	Presenta importantes omisiones, dificultades o errores en el desempeño, que no permiten evidenciar los elementos básicos del logro del indicador, por lo que no puede ser considerado competente.
Desempeño no logrado	0%	Presenta ausencia o incorrecto desempeño.

Indicador de Evaluación	Categorías de Respuesta				Ponderación Indicador de Evaluación
	Muy buen desempeño 100%	Desempeño aceptable 60%	Desempeño incipiente 30%	Desempeño no logrado 0%	
Dimensión: Encargo					
IE1.3.1 Aplica programación orientada a objetos (POO) estructurando clases, herencia y polimorfismo dentro de un proyecto funcional que respete la arquitectura propuesta	Implementa correctamente clases, herencia y polimorfismo respetando la arquitectura definida. Las clases tienen relaciones coherentes, y el código demuestra comprensión sólida de POO.	Aplica POO correctamente con herencia y clases estructuradas, aunque con pequeños errores o sin polimorfismo aplicado de forma explícita.	Utiliza clases, pero sin relaciones claras o con una implementación parcial de los principios de POO. No se evidencia arquitectura definida.	Construye clases aisladas sin conexión entre sí, con errores conceptuales graves o sin aplicar herencia ni polimorfismo o no se implementa en absoluto.	10%
IE 2.3.1 Estructura el proyecto aplicando principios de modularidad, separando responsabilidades lógicas, visuales y funcionales, e integrando persistencia de datos local.	Estructura el proyecto aplicando principios de modularidad clara, con separación adecuada de responsabilidades, integración de persistencia local y organización que favorece la mantenibilidad y la comunicación de la lógica y la UI.	Estructura el proyecto de manera funcional, pero tiene problemas de organización, escasa modularidad o integración de la persistencia local con acoplamientos.	Organiza el proyecto de manera básica y funcional, pero con limitaciones estructurales o persistencia poco integrada.	El proyecto no presenta una estructura modular clara ni persistencia local implementada de forma efectiva.	10%
IE 2.3.2 Utiliza herramientas de colaboración y control de versiones de forma efectiva, evidenciando participación activa en el desarrollo grupal.	Utiliza herramientas como GitHub y Trello, con commits distribuidos, issues resueltos y tareas colaborativas visibles en la planificación del proyecto.	Utiliza herramientas colaborativas, pero limitado a uno de los integrantes o con registros poco consistentes de la participación grupal.	Utiliza herramientas de manera básica o puntual, sin seguimiento constante ni planificación, y con una participación poco distribuida entre el equipo.	No se entrega evidencia del uso de herramientas de colaboración o el repositorio está vacío, duplicado o mal gestionado.	10%

IE 2.4.1 Accede de forma segura y funcional a al menos dos recursos del dispositivo, integrándolos con coherencia en la interfaz y el flujo de la aplicación.	Accede correctamente a dos o más recursos del dispositivo, integrándolos de forma segura, funcional y con coherencia en el flujo de la interfaz.	Accede a menos de dos recursos del dispositivo, además presenta deficiencias en permisos, seguridad o coherencia visual en uno de ellos.	-	No se accede a recursos nativos o la funcionalidad entregada no opera correctamente.	15%
IE 3.1.1 Desarrolla una app móvil con todas sus pantallas, formularios y flujos funcionales completos sin errores de navegación o ejecución.	Desarrolla la app móvil con todas sus pantallas, formularios y flujos funcionales completos sin errores de navegación o ejecución.	Desarrolla la mayoría de las pantallas, los flujos están implementados, existen errores menores o formularios incompletos.	Desarrolla parcialmente el front end de la aplicación, implementando algunos flujos y formularios completos solo en las secciones entregadas, sin cubrir la totalidad de la interfaz requerida.	Desarrolla únicamente una parte limitada del frontend, presentando errores evidentes en la navegación, formularios sin lógica funcional, o directamente omitiendo el desarrollo de la interfaz.	15%
IE 3.1.2 Desarrolla los microservicios necesarios para la lógica del proyecto, configurando la base de datos y exponiendo endpoints funcionales.	Desarrolla todos los microservicios requeridos, con su base de datos operativa y endpoints funcionales y coherentes.	Desarrolla los microservicios requeridos, pero uno o más presentan errores de estructura o endpoints no funcionales.	-	Desarrolla parcialmente el backend, con microservicios incompletos o bases de datos mal configuradas.	20%
IE 3.1.3 Consume una API externa desde la aplicación móvil, integrándola al flujo visual sin interferir con los microservicios propios.	Consumе una API externa desde la aplicación correctamente, y su uso está integrado visualmente en la app móvil, de forma coherente y funcional.	La API externa está implementada, pero sin integración clara en la interfaz o con errores parciales.	-	No se integra el consumo de una API externa o el servicio no opera.	5%
IE 3.2.1 Integra pruebas unitarias funcionales en los módulos de la aplicación móvil.	Integra las pruebas unitarias para cubrir el 80% del código lógico o más, son funcionales y están implementadas con herramientas adecuadas.	-	-	Integra las pruebas unitarias, pero no cubren el 80% del código o presentan errores parciales en su ejecución.	10%
IE 3.3.1 Genera el archivo APK firmado, incluyendo la configuración técnica necesaria (build.gradle, jks).	Genera correctamente el APK firmado, con todos los archivos necesarios y configuración técnica funcional.	-	-	El APK está generado, pero con errores en la firma o sin incluir los archivos y configuraciones correspondientes.	5%
Total					100%