



UNIVERSIDAD DEL VALLE  
FACULTAD DE INGENIERÍA

**CODELAB EDT proyecto ShopU**

**AUTORES:**

Manuel Alexander Serna Jaraba - 202259345

Santiago Villa Salazar - 202259527

Adrian Felipe Velasquez Arias - 202259456

Edgar Fabian Rueda Colonia - 2259606

Kevin Alejandro Marulanda Escobar - 2380697

**DOCENTE:**

SALAZAR VICTORIA, Alvaro Hernando

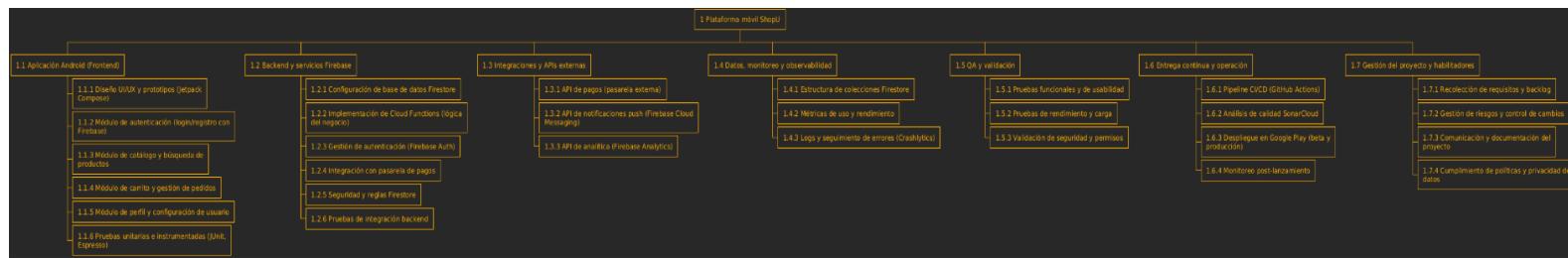
**CURSO:**

Introducción a la Gestión de Proyectos de Software

Tuluá, Valle del Cauca

Agosto de 2025

Usando las instrucciones y teoría del codelab construimos el siguiente EDT:



o link ya que no se aprecia bien:

<https://bit.ly/EDTshopU>

El **nivel 1 (raíz)** es el producto principal: “*Plataforma móvil ShopU*”. Cada rama de **nivel 2** representa **componentes o áreas de entregables**, no fases (Frontend, Backend, Integraciones, QA, etc.). Los **niveles 3** ya son **paquetes de trabajo gestionables** (8–80 h aprox). Cumple la **regla del 100% y MECE**, sin solapamientos. Los identificadores (1.1.1, 1.2.3, etc.) siguen la **codificación jerárquica** que luego usare en el diccionario WBS y RACI.

A continuación el diagrama OBS:



<https://bit.ly/OBSshopU>

ahora seguiremos con el diccionario wbs:

| ID WBS | Nombre del paquete            | Descripción  | Entregables  | DoD (criterios de aceptación)                                   | Exclusiones                         | Supuestos                                     | Dependencias | Responsable (OBS)     | Esfuerzo (h) | Duración (d) | Riesgos                      | Métricas                            |
|--------|-------------------------------|--|--|---|-------------------------------------|---|--------------|-----------------------|--------------|--------------|------------------------------|-------------------------------------|
| 1.1.1  | Diseño UI/UX                  | Diseñar interfaces y flujo de navegación en Figma para la app ShopU    | Prototipo Figma, mapa de navegación, guía de estilos | Revisión aprobada por PO; prototipo navegable sin errores       | No incluye desarrollo en código     | Se cuenta con lineamientos de marca           | —            | Diseñador UI/UX       | 20           | 3            | Cambios frecuentes de diseño | % de pantallas aprobadas            |
| 1.1.2  | Autenticación (Firebase Auth) | Implementar login, registro y recuperación de contraseña               | Pantallas de login/registro; conexión Firebase Auth  | Login funcional y seguro; usuarios creados visibles en Firebase | No incluye roles avanzados          | Firebase configurado previamente              | 1.1.1        | Desarrollador Android | 30           | 4            | Errores de conexión          | Tasa de autenticaciones exitosas    |
| 1.1.3  | Catálogo y búsqueda           | Implementar vista de productos con búsqueda por texto                  | Lista de productos, barra de búsqueda funcional      | Catálogo visible y filtrable sin errores                        | No incluye gestión de stock         | Base de datos poblada con productos de prueba | 1.2.1        | Desarrollador Android | 35           | 5            | Datos inconsistentes         | Tiempo de carga de catálogo         |
| 1.1.4  | Carrito y pedidos             | Crear flujo de compra: agregar, quitar y confirmar pedido              | Vista de carrito, resumen de pedido, confirmación    | Flujo completo desde selección hasta confirmación               | No incluye integración real de pago | Firebase Firestore operativa                  | 1.1.3        | Desarrollador Android | 40           | 5            | Errores en lógica            | % pedidos registrados correctamente |
| 1.1.5  | Perfil de usuario             | Crear módulo de perfil (datos y foto)                                  | Pantalla de perfil, actualización de datos           | Datos editables y sincronizados con Firebase                    | No incluye configuración avanzada   | Usuario autenticado                           | 1.1.2        | Desarrollador Android | 20           | 3            | Fallas en sincronización     | Tasa de actualización exitosa       |
| 1.2.1  | Firestore DB                  | Definir y crear colecciones principales (usuarios, productos, pedidos) | Estructura Firestore documentada y creada            | Consultas CRUD exitosas en cada colección                       | No incluye migraciones              | Acceso a Firebase habilitado                  | —            | Desarrollador Backend | 24           | 4            | Estructura ineficiente       | % consultas exitosas                |
| 1.2.2  | Cloud Functions               | Programar funciones básicas: registrar pedidos, validar pagos          | Código desplegado en Firebase Functions              | Funciones invocables sin error; logs correctos                  | No incluye integración externa real | Firebase CLI operativo                        | 1.2.1        | Desarrollador Backend | 32           | 4            | Fallos en despliegue         | % funciones exitosas                |
| 1.3.1  | Notificaciones push           | Enviar notificaciones de pedido y promociones                          | Configuración de FCM; envío de pruebas               | Recibe notificación al generar pedido                           | No incluye segmentación avanzada    | Tokens de usuarios disponibles                | 1.2.2        | Desarrollador Android | 16           | 2            | Falla en permisos o FCM      | % entregas de notificación          |

|       |                              |   |   |  |                                     |                                |       |                     |    |   |                           |                               |
|-------|------------------------------|---|---|--|-------------------------------------|--------------------------------|-------|---------------------|----|---|---------------------------|-------------------------------|
| 1.4.1 | Pruebas funcionales          | Ejecutar pruebas de las principales funciones de la app | Lista de casos de prueba, reporte de defectos | Todas las pruebas críticas superadas (>90%)      | No incluye performance testing      | App estable                    | 1.1.* | QA Tester           | 20 | 3 | Bugs no detectados        | % casos aprobados             |
| 1.4.2 | Validación con usuarios      | Demostración funcional y retroalimentación              | Informe de resultados, mejoras sugeridas      | 80% de tareas aprobadas por usuarios evaluadores | No incluye correcciones posteriores | Usuarios de prueba disponibles | 1.4.1 | QA / PO             | 12 | 2 | Baja participación        | Nivel de satisfacción usuario |
| 1.5.1 | Entrega beta interna         | Generar versión instalable (APK) y subir a repositorio  | APK firmado, documentación de instalación     | APK instalable y sin fallas críticas             | No incluye Play Store               | CI local funcional             | 1.4.1 | DevOps              | 10 | 1 | Fallos en compilación     | Tasa de instalación exitosa   |
| 1.5.2 | Entrega final y demo         | Preparar versión final y presentación                   | APK final, presentación, manual técnico       | Demo funcional sin errores críticos              | No incluye soporte postproyecto     | Todos los módulos completos    | 1.5.1 | Gerente de proyecto | 16 | 2 | Fallo durante la demo     | Cumplimiento de entregables   |
| 1.6.1 | Control de cambios y riesgos | Documentar y gestionar riesgos y cambios                | Matriz de riesgos actualizada                 | Lista revisada semanalmente                      | No incluye mitigaciones costosas    | Reuniones semanales activas    | 1.6.1 | Gerente de proyecto | 8  | 1 | Cambios tardíos           | Riesgos controlados (%)       |
| 1.6.2 | Documentación y presentación | Elaborar documentación final técnica y de usuario       | Informe técnico y manual de usuario           | Documentos entregados y revisados                | No incluye video tutorial           | Equipo disponible              | 1.5.2 | Gerente de proyecto | 16 | 2 | Retrasos en consolidación | % documentos entregados       |

A continuación, la matriz RACI:

| ID WBS | Paquete / Entregable               | Product Owner (PO) | Gerente de Proyecto | Líder Android | Desarrollador Android | Desarrollador Backend (Firebase) | QA / Tester | DevOps | Diseñador UI/UX |
|--------|------------------------------------|--------------------|---------------------|---------------|-----------------------|----------------------------------|-------------|--------|-----------------|
| 1.1.1  | Diseño UI/UX                       | A                  | I                   | C             | I                     | I                                | I           | I      | R               |
| 1.1.2  | Autenticación (Firebase Auth)      | C                  | I                   | A             | R                     | C                                | C           | I      | I               |
| 1.1.3  | Catálogo y búsqueda                | C                  | I                   | A             | R                     | C                                | C           | I      | C               |
| 1.1.4  | Carrito y pedidos                  | I                  | C                   | A             | R                     | C                                | C           | I      | I               |
| 1.1.5  | Perfil de usuario                  | I                  | I                   | A             | R                     | C                                | C           | I      | I               |
| 1.2.1  | Firestore DB                       | I                  | I                   | C             | C                     | A/R                              | I           | I      | I               |
| 1.2.2  | Cloud Functions                    | I                  | I                   | C             | C                     | A/R                              | C           | I      | I               |
| 1.3.1  | Notificaciones push                | I                  | C                   | A             | R                     | C                                | I           | I      | I               |
| 1.4.1  | Pruebas funcionales                | I                  | I                   | C             | C                     | C                                | A/R         | I      | I               |
| 1.4.2  | Validación con usuarios            | A                  | I                   | C             | C                     | I                                | R           | I      | C               |
| 1.5.1  | Entrega beta interna               | I                  | C                   | I             | C                     | I                                | I           | A/R    | I               |
| 1.5.2  | Entrega final y demo               | C                  | A/R                 | I             | I                     | I                                | C           | I      | C               |
| 1.6.1  | Control de cambios y riesgos       | I                  | A/R                 | I             | I                     | I                                | C           | I      | I               |
| 1.6.2  | Documentación y presentación final | C                  | A/R                 | C             | C                     | C                                | C           | I      | C               |

Pasamos con una evaluación final de lo anteriormente trabajado:

## Evaluación Final

| Criterio de evaluación                             | Resultado                                  | Evidencia  |
|--|--|--|
| La EDT está orientada a entregables (sustantivos)  | Cumple <input checked="" type="checkbox"/> | Todos los ítems son productos o resultados verificables, sin verbos de acción.               |
| Todos los paquetes tienen DoD y responsable        | Cumple <input checked="" type="checkbox"/> | Cada paquete define criterios de aceptación medibles y un responsable en la OBS.             |
| La RACI tiene un A por fila                        | Cumple <input checked="" type="checkbox"/> | Cada paquete cuenta con un único Accountable, conforme al modelo RACI.                       |
| Los artefactos están versionados en Git            | Cumple <input checked="" type="checkbox"/> | El proyecto se gestiona mediante repositorio GitHub, con control de versiones de artefactos. |
| Existe trazabilidad EDT ↔ OBS ↔ Diccionario ↔ RACI | Cumple <input checked="" type="checkbox"/> | Se garantiza la trazabilidad entre todos los artefactos del proyecto ShopU.                  |

## Conclusión

El proyecto ShopU cumple todos los criterios de la práctica guiada del CODELAB. Los artefactos presentan coherencia, trazabilidad y cumplimiento de buenas prácticas de gestión de proyectos de software, garantizando un modelo end-to-end sólido, verificable y alineado con los estándares planteados