# Plan de Administración del Proyecto

**CampusGo**

***JaveSpark***

****

Pontificia Universidad Javeriana  
Fecha: 03 de Mayo, 2025

Equipo de trabajo:

* Carlos Mercado
* David Mercado
* Sara Albarracín
* Victoria Roa
* Alejandro Caicedo

## Historial de Cambios

| **Fecha** | **Descripción del Cambio** | **Autor** |
| --- | --- | --- |
| 03/05/2025 | Creación inicial del documento | Sara Albarracín |

## Tabla de Contenidos

1. Introducción
2. Alcance del Proyecto
3. Administración del Proyecto  
   3.1 Modelo del Ciclo de Vida  
   3.2 Lenguajes y Herramientas  
   3.3 Entidades Externas  
   3.4 Cronograma y Sprints
4. Procesos de Soporte  
   4.1 Gestión del Progreso  
   4.2 Administración de Configuración  
   4.3 Procesos de Calidad
5. Anexos

## Introducción

Este documento describe la planificación y gestión del proyecto CampusGo, una aplicación móvil diseñada para conectar estudiantes de diversas universidades, facilitando transacciones seguras de productos mediante funcionalidades innovadoras como seguimiento en tiempo real, autenticación robusta y comunicación integrada.

## Alcance del Proyecto

CampusGo permitirá:

* Registro y autenticación con carnet institucional.
* Compra y venta de artículos académicos y personales.
* Confirmación segura de entregas mediante NFC o código de confirmación.
* Seguimiento de pedidos en tiempo real.
* Calificaciones y reseñas mutuas entre usuarios.
* Comunicación directa mediante chat integrado (texto y voz).
* Verificación de identidad mediante reconocimiento facial o huella.

No incluye:

* Versiones para plataformas distintas de Android.
* Integración con sistemas externos no mencionados explícitamente.

## Administración del Proyecto

### 3.1 Modelo del Ciclo de Vida

CampusGo utiliza un modelo ágil basado en Scrum, caracterizado por iteraciones cortas y entregas frecuentes. Las prácticas específicas incluyen:

* Historias de usuario para definición de funcionalidades.
* Reuniones semanales (Weekly Meetings).
* Prototipado iterativo con mockups realizados en Marvel.
* Revisión continua del código (Peer Reviews).
* Diagramas de clase y casos de uso actualizados constantemente.

### 3.2 Lenguajes y Herramientas

**Lenguajes:**

* Kotlin (lógica de negocio)
* XML (interfaces gráficas)

**Herramientas:**

* Android Studio (IDE principal para desarrollo)
* Firebase (autenticación, base de datos)
* GitHub (control de versiones)
* Canva y Marvel (diseño de mockups)
* NFC Tools (manejo de tecnología NFC)
* OSM y OSM Bonuspack (servicios de mapas y rutas)

### 3.3 Entidades Externas

| **Entidad** | **Descripción** | **Responsabilidades** | **Comunicación** |
| --- | --- | --- | --- |
| Firebase | Plataforma de autenticación y almacenamiento. | Gestión de usuarios y datos | firebase.google.com |
| OSM y OSM Bonuspack | Proveedores de servicios de mapas interactivos y rutas | Visualización y seguimiento de pedidos y rutas | openstreetmap.org / bonuspack |

### 3.4 Cronograma y Sprints

| **Sprint** | **Fechas** | **Tareas principales** | **Entregables clave** |
| --- | --- | --- | --- |
| 1 | 01/05/2025 - 15/05/2025 | Autenticación, navegación inicial, mockups XML | Prototipo inicial funcional |
| 2 | 16/05/2025 - 31/05/2025 | Gestión de productos, carrito de compras, chats (texto y voz) | Prototipo completo intermedio |
| 3 | 01/06/2025 - 15/06/2025 | Integración de mapas (OSM), NFC o código de confirmación, confirmaciones y calificaciones | Producto completo funcional |

## 4. Procesos de Soporte

### 4.1 Gestión del Progreso

**Métricas:**

* Cantidad de historias de usuario completadas.

**Reportes:**

* Reuniones semanales (Weekly Meetings).

**Acciones correctivas:**

* Repriorización inmediata del backlog.
* Asignación adicional de recursos en caso de retrasos significativos.

### 4.2 Administración de Configuración

**Ramas GitHub:**

* **main:** Versión estable del producto.
* **develop:** Integración continua para nuevas funcionalidades.
* **feature/x:** Desarrollo individual de funcionalidades específicas.

**Caso de desarrollo:** Desarrollo iterativo con revisión obligatoria mediante pull requests antes del merge.

### 4.3 Procesos de Calidad

| **Proceso** | **Momento del proyecto** |
| --- | --- |
| Revisión de código | Al completar cada historia de usuario |
| Auditorías internas | Al final de cada sprint |

## 5. Anexos

En esta sección se incorporarán documentos complementarios relevantes para ampliar información específica según se requiera en fases posteriores del proyecto.