



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

Universidad Nacional de Colombia - Sede Bogotá  
Facultad de Ingeniería  
Curso: Ingeniería de Software I, Gr. 1  
**Grupo de Proyecto 8**

## **Proyecto Final**

### **Punto 2: Levantamiento de requerimientos**

#### **a) Narrativa del levantamiento inicial de requerimientos**

En los primeros encuentros entre los integrantes del grupo, en horas de clase, se realizaron algunas discusiones y lluvias de ideas sobre posibles proyectos, dados los requerimientos y alcance del curso, además del tiempo disponible; dichas reuniones fueron complementadas con comunicación asincrónica.

Eventualmente, se determinó que el proyecto tomase un enfoque académico, en torno a educación superior, esto considerando que, como estudiantes activos de la Universidad, sería más fácil identificar posibles problemas a solucionar, necesidades que satisfacer o incluso, herramientas que podrían facilitar ciertos procesos en el entorno universitario, todo esto basado en experiencias generales del grupo de desarrolladores.

Finalmente, surgió la idea de centralizar el seguimiento y entrega virtual/asincrónica de proyectos, asignaciones o tareas en una sola plataforma, esto con el fin de abordar problemáticas relacionadas con la falta de eficiencia, coordinación y consistencia entre profesores y estudiantes a la hora de asignar, entregar (parcial o totalmente) y retroalimentar dichos proyectos académicos.

Se sabe que, en contexto de entregas virtuales o asíncronas, se suele utilizar más de una herramienta existente para las tareas ya mencionadas. Por ejemplo, correo electrónico para asignar las entregas, Moodle o Classroom para obtener material y realizar las entregas, y, posiblemente, nuevamente el correo electrónico para las entregas intermedias o retroalimentación directa que surja.

Este modelo de gestión virtual puede conllevar ciertas dificultades organizativas requiriendo cambiar de aplicaciones constantemente, potencialmente dificultando la monitorización de proyectos entregados en cada paso del desarrollo del proyecto y la comunicación retroalimentativa, conllevando posibles equivocaciones, retrasos y falta de transparencia.

Se busca proporcionar un sistema en el que el público objetivo, profesores y estudiantes, encuentre funcionalidades prácticas e intuitivas para la creación de grupos o cursos concretos, asignación de proyectos, entrega de avances de proyecto o proyectos ya finalizados y dar/entregar su retroalimentación correspondiente de forma oportuna, gestión de varios proyectos, subida de documentos relacionados con los proyectos, registro de actividades para hacer un seguimiento del progreso individual o grupal, entre otras.

La expectativa general es que este sistema simplifique la interacción entre docentes y estudiantes, minimizando la pérdida de información y mejorando la experiencia académica, todo sin perder funciones que otorgan calidad de vida a la creciente necesidad de

digitalizar los procesos académicos, como por ejemplo, las notificaciones de nueva tarea que otorgan plataformas como Classroom.

Como equipo de estudiantes de ingeniería y desarrollo de software, esperamos que este proyecto nos permita poner en práctica nuestros conocimientos en programación, bases de datos, y gestión de proyectos en un contexto realista, a medida que los adquirimos, aprendiendo nuevas herramientas, lenguajes y frameworks relevantes para el desarrollo backend y frontend, como FastAPI o React, sin perder de vista el mejoramiento de nuestras habilidades colaborativas de trabajo en equipo y coordinación, ya sea en tiempo real o asíncrona.

También deseamos contribuir una posible solución adaptable a contextos educativos reales como un aporte potencialmente significativo y tangible a la mejora de estos procesos académicos.

## **b) Identificación inicial de requerimientos**

En este paso se logra, también, identificar algunos posibles procesos asociados a los requerimientos.

- **Gestión de Usuarios:** Los Usuarios deben poder registrarse e iniciar sesión en el sistema, con roles diferenciados para estudiantes y docentes. El Administrador gestiona los roles.
- **Creación y Asignación de Proyectos y Tareas:** Los Docentes pueden crear proyectos académicos (véase, tareas/asignaciones), definiendo un título, descripción, fecha de entrega, y asignando Estudiantes al proyecto.
- **Gestión de Proyectos y Tareas:** Los Docentes pueden actualizar el estado del proyecto.
- **Seguimiento del Estado del Proyecto:** Cada proyecto tiene un estado (por ejemplo, "En progreso", "Finalizado") que se actualiza según el progreso de las tareas. Los Docentes pueden actualizar el estado del proyecto. Los Estudiantes pueden consultar el estado del proyecto.
- **Actualización de Avances y Observaciones:** Los estudiantes pueden registrar avances en cada tarea y recibir comentarios del docente para asegurar el cumplimiento de objetivos.
- **Adjuntar Documentos y Archivos:** Se permite que los Estudiantes suban archivos o enlaces relevantes para sus proyectos asignados. Los Docentes pueden subir archivos y enlaces a cada proyecto (por ejemplo, guía o materiales relacionados a este).
- **Accesibilidad y Usabilidad:** La aplicación debe funcionar en una plataforma web, accesible tanto en computadoras de escritorio como en dispositivos móviles. Debe ser una plataforma intuitiva.
- **Seguridad:** La plataforma almacenará datos personales de las personas involucradas (estudiantes, docentes y otro personal académico), además de credenciales de inicio de sesión acorde a cada rol asignado. Estos datos deben ser protegidos adecuadamente mediante técnicas de cifrado de datos.
- **Notificaciones de Actividades:** El sistema envía notificaciones sobre actualizaciones importantes, como nuevos comentarios, tareas asignadas o cambios en las fechas límite.

- Historial de Actividades: La aplicación mantiene un registro de todas las actividades relacionadas con cada proyecto, así, los Estudiantes y Docentes pueden consultar el progreso de estas.
- Visualización: Proporcionar a la plataforma un diseño intuitivo y legible, con tipografías claras y simples, íconos y botones llamativos, y colores profesionales.

### Punto 3: Análisis de requerimientos

#### a) Narrativa del levantamiento inicial de requerimientos

En los primeros encuentros en

#### a) Lista consolidada de requerimientos

Organizando, desglosando y consolidando los requerimientos y procesos identificados con el desarrollo y síntesis de la idea del proyecto. Se visualizan primero los Requerimientos Funcionales (RF), luego los No-Funcionales (RNF).

Requerimientos Funcionales (RF):

**\*Prioridad asociada antes de hacer la clasificación MoSCoW**

ID	Usuario (Opcional): Funcionalidad	Proceso Asociado	Prioridad*
RF-001	Usuario (cualquiera): El Usuario se registra en la Aplicación con su información– el registro no otorga rol por defecto.	Gestión de usuarios	Alta
RF-002	Usuario (cualquiera): Iniciar sesión en la Aplicación con sus datos registrados.	Gestión de usuarios	Alta
RF-003	Administrador: Asignar roles de Estudiante o Profesor al Usuario	Gestión de usuarios	Alta
RF-004	Profesor: Crear grupos (clases) y asignar Estudiantes a dichos grupos para el seguimiento o entrega de sus proyectos.	Gestión de usuarios	Alta
RF-005	Profesor: Crear proyectos (véase: tareas o asignaciones) para un grupo (clase). Contiene información como título, descripción, fecha de entrega y estado.	Gestión de asignaciones	Alta
RF-006	Profesor: Asignar proyectos a Estudiante(s) en específico del curso en cuestión.	Gestión de asignaciones	Media
RF-007	Profesor: Actualizar el estado de un proyecto acorde al avance de las tareas (entregas intermedias registradas).	Gestión de asignaciones	Baja
RF-008	Estudiante: Consultar el estado de un proyecto asignado y su información correspondiente.	Gestión de asignaciones	Alta
RF-009	Estudiante: Registrar un avance (ej. entrega intermedia) en cada proyecto asignado, según sea requerido.	Gestión de asignaciones	Media
RF-010	Estudiante: Realizar una entrega final para un proyecto asignado.	Gestión de asignaciones	Alta
RF-011	Profesor: Otorgar retroalimentación a las entregas de proyecto realizadas por estudiantes asignados; se entrega a ellos.	Gestión de asignaciones	Media

RF-012	Estudiante: Subir archivos o enlaces relevantes para las entregas de proyectos asignados.	Archivos adjuntos	Alta
RF-013	Profesor: Subir archivos o enlaces para cada proyecto (ej. guías o materiales).	Archivos adjuntos	Alta
RF-014	Usuario (cualquiera): Subir una foto o ícono de perfil asociada al usuario en cuestión.	Archivos adjuntos	Baja
RF-015	Usuarios: Consultar progresos anteriores de actividades asignadas.	Registro de actividades	Baja

Requerimientos No-Funcionales (RNF):

**\*Prioridad asociada antes de hacer la clasificación MoSCoW**

ID	Usuario (Opcional): Funcionalidad	Proceso Asociado	Prioridad*
RNF-001	La Aplicación debe ser compatible con dispositivos móviles (smartphones, tablets) y de escritorio (PC).	Accesibilidad y usabilidad	Alta
RNF-002	La Aplicación debe ser intuitiva y fácil de usar.	Accesibilidad y usabilidad	Alta
RNF-003	Asegurar la protección de datos personales y/o sensibles mediante cifrado.	Seguridad	Alta
RNF-004	Utilizar credenciales seguras para evitar brechas de seguridad o accesos de rol no autorizados.	Seguridad	Alta
RNF-005	La Aplicación en menos de 5 segundos en conexiones en condiciones estándar.	Rendimiento	Alta
RNF-006	La Aplicación debe soportar usuarios múltiples sin detrimento del rendimiento.	Rendimiento	Media
RNF-007	La Aplicación envía notificaciones a los Usuarios sobre actualizaciones importantes (ej. nueva tarea, comentarios, cambios en fechas, entrega realizada).	Notificaciones	Media
RNF-008	La Plataforma mantiene un registro de actividades de cada proyecto.	Registro de actividades	Baja
RNF-009	El diseño del sitio debe permitir la integración a futuro de nuevas funcionalidades sin requerir una reestructuración completa.	Escalabilidad	Media
RNF-010	Diseño visual limpio, con una tipografía clara.	Diseño	Media
RNF-011	Paleta de colores profesional. Íconos y botones llamativos.	Diseño	Media

### b) Procesos Asociados identificados

- Para Requerimientos Funcionales (RF):
  - Gestión de usuarios
  - Gestión de asignaciones
  - Archivos adjuntos
  - Gestión de cursos (opcional)
- Para Requerimientos No-Funcionales (RNF):
  - Accesibilidad y usabilidad
  - Seguridad
  - Rendimiento
  - Notificaciones
  - Registro de actividades
  - Escalabilidad
  - Diseño

### c) Clasificación MoSCoW

Se han tenido en cuenta ciertos factores para la clasificación de las funcionalidades y su estimación de esfuerzo.

**Recursos disponibles:** Un equipo de 4 estudiantes universitarios que harán las veces de desarrolladores, arquitectos (de software), diseñadores de la ux y testers. Se cuenta con ciertos recursos cuyas limitaciones varían:

- Tiempo para el diseño, desarrollo y testing: varía según cada miembro del equipo. Se tienen en cuenta factores como los estudios universitarios y trabajos fuera de la universidad.
- Repositorio: Para este proyecto se utilizará un repositorio de GitHub. Al contar con una cuenta Pro, se dispone de herramientas de despliegue, en caso de ser necesarias.
- Almacenamiento en la nube: para la sincronización y/o construcción colaborativa de documentos varios, se utilizan servicios como Google Docs o Microsoft 365, disponibles mediante el correo universitario. El límite de almacenamiento varía entre 15 y 100 GB, según cada servicio. La sincronización del desarrollo del proyecto depende del repositorio de GitHub.
- Herramientas externas: El software de desarrollo preferido también puede variar según el desarrollador y posteriormente pueden surgir más opciones a considerar; entre ellos pueden incluirse Visual Studio Code, PyCharm Community Edition\*, etc. Entre los frameworks considerados, se encuentran: React y FastAPI. Otras herramientas de apoyo pueden incluir GitHub Copilot Pro y OpenAI ChatGPT.

*(\*al no usarse en un entorno de desarrollo profesional, no debería implicar violación a los ToS).*

**Complejidad técnica:** dadas las características y necesidades del proyecto, además de los recursos disponibles con los que cuenta el equipo de desarrollo, se han identificado tres tipos de tareas, según su complejidad de implementación, independientemente de su prioridad:

- Tareas Básicas (2 a 5 puntos)
- Tareas Medias/moderadas (5 a 8 puntos)
- Tareas Complejas (8 a 13 puntos)

### Impacto en la experiencia del usuario y Relación con los objetivos del proyecto:

En primera instancia, el objetivo se centra en construir una aplicación capaz de trabajar en sus funcionalidades primordiales, priorizadas como Must (y Should) en este módulo inicial. Las mejoras en experiencia de usuario son prioridad en funcionalidades posteriores, proyectadas como Should, Could y Won't.

La clasificación detallada de las funcionalidades, cada una con su valor de esfuerzo y justificación es la siguiente:

<b>Must Have (M)</b> Esenciales para funcionamiento básico.		
<b>Funcionalidad</b>	<b>Esfuerzo (Fibonacci)</b>	<b>Motivo</b>
<b>RF-001</b> - Registro de usuarios	3	Requiere validación de datos como correo electrónico (formato único y correcto), además de almacenar contraseñas de forma segura.
<b>RF-002</b> - Inicio de sesión	5	Involucra autenticación segura, y validación de usuario. Requiere respuesta rápida del sistema.
<b>RF-003</b> - Asignación de roles	5	Requiere gestión y control de acceso según los roles asignados, probablemente mediante una interfaz destinada a ello.
<b>RF-005</b> - Creación de proyectos	5	Involucra almacenamiento en la base de datos, manejo de formularios y validación de datos.
<b>RF-008</b> - Consulta de estado de proyectos	3	Necesita consultas a la base de datos. Visualización en frontend.
<b>RF-010</b> - Entrega final de proyectos	5	Involucra permitir carga de archivos y/o enlaces externos, validación del estado del proyecto y control de fechas previstas.
<b>RNF-001</b> - Compatibilidad multiplataforma web	8	Requiere pruebas en varias plataformas, además de un diseño responsivo.
<b>RNF-002</b> - Usabilidad	5	Necesita un diseño bien estructurado de navegación, UI con un diseño bien realizado.
<b>RNF-003</b> - Protección de	8	Puede involucrar la implementación de

<b>Must Have (M)</b> Esenciales para funcionamiento básico.		
datos		medidas de seguridad tales como cifrado de datos sensibles.
<b>RNF-004</b> - Seguridad de credenciales	13	Puede involucrar hashing de contraseñas, además de otras políticas de contraseñas seguras, además de medidas de protección de roles autorizados tales como autenticación segura, basada en roles o protección contra escalada de privilegios.
<b>RNF-009</b> - Diseño modular	5	Necesita una buena estructuración de la arquitectura del proyecto que permita escalabilidad

<b>Should Have (S)</b> Importantes, pero no críticas en el módulo 1		
Funcionalidad	Esfuerzo (Fibonacci)	Motivo
<b>RF-004</b> - Creación de grupos (clases)	5	Necesita consultas de bases de datos, además de implementar un sistema de asociación entre profesores y estudiantes (como IDs de grupos en la base de datos).
<b>RF-011</b> - Retroalimentación de entregas de proyectos	5	Debe crearse una interfaz para comentarios y almacenamiento en la base de datos. Deben manejarse permisos para la retroalimentación (que se entregue a el/los estudiante(s) de la entrega en particular).
<b>RF-012</b> - Subida de archivos por estudiantes	8	Implica almacenamiento en el servidor. Necesita restricciones de tamaño y formato de archivos.
<b>RF-013</b> - Subida de archivos por profesores	8	Similar a la subida de archivos por estudiantes. Necesita cambios en ciertos permisos.
<b>RNF-006</b> - Capacidad para múltiples usuarios sin detrimento de rendimiento	8	Necesita optimizar consultas en la base de datos, además de un manejo eficiente de las sesiones.



<b>Could Have (C)</b> Deseables, pero pueden implementarse luego		
<b>Funcionalidad</b>	<b>Esfuerzo (Fibonacci)</b>	<b>Motivo</b>
<b>RF-006</b> - Asignación de proyectos a estudiantes en específico	5	Requiere implementar un mecanismo adicional de asignación y control de permisos en proyectos (ejemplo, IDs de proyecto).
<b>RF-007</b> - Actualización del estado de avance del proyecto por parte del profesor	5	Puede postergarse, debido a que el primer módulo sólo maneja entregas finales. Requiere manejo de estados “intermedios” de un proyecto (ejemplo, un campo de estado reportado) en la base de datos. Necesita una interfaz para el registro y la visualización.
<b>RF-009</b> - Registro de avances en proyectos (como entregas intermedias) por parte de estudiantes	5	Similar a la funcionalidad de actualización de estado por parte del profesor, requiere manejo de estados “intermedios” de un proyecto en la base de datos (ejemplo, historial de entregas intermedias). Necesita una interfaz para el registro y la visualización.
<b>RF-015</b> - Consulta de progresos previos	2	Similar al registro de actividades. Requiere mayor organización en base de datos y consultas especializadas por proyectos y usuarios.
<b>RNF-005</b> - Respuesta rápida del sistema en conexiones estándar	5	Gran parte de la optimización es una consecuencia de la implementación inicial de la arquitectura y la base de datos. Puede optimizarse más una vez se desarrolle la funcionalidad principal.
<b>RNF-007</b> - Notificaciones	8	Requiere manejo de eventos y notificaciones, principalmente por correo electrónico. Se considera manejarlas en tiempo real también.
<b>RNF-008</b> - Registro de actividades	5	Puede añadirse posteriormente sin afectar la funcionalidad principal de la aplicación. Requiere consultas en base de datos para cada proyecto e interfaz de visualización.
<b>RNF-010</b> - Diseño visual limpio	3	Puede mejorarse en iteraciones posteriores sin afectar la funcionalidad del sistema. Requiere indagar sobre e implementar buenas prácticas de diseño web.

<b>Won't Have (W)</b> Fuera del alcance actual del proyecto		
Funcionalidad	Esfuerzo (Fibonacci)	Motivo
<b>RF-014</b> - Personalizar ícono de perfil	3	No impacta la funcionalidad del sistema. Implica almacenamiento en el servidor, además de un control de permisos (cada usuario sólo maneja su foto; dicha foto es pública).
<b>RNF-011</b> - Paleta de colores profesional, elementos llamativos	2	Al igual que el diseño visual limpio, requiere implementar buenas prácticas de diseño y puede implementarse posteriormente.

La priorización realizada, tanto para Requerimientos Funcionales (RF) como para Requerimientos No-Funcionales (RNF), puede resumirse así:

Must Have (M)	Should Have (S)	Could Have (C)	Won't Have (W)
RF-001	RF-004	RF-006	RF-014
RF-002	RF-011	RF-007	<i>RNF-011</i>
RF-003	RF-012	RF-009	
RF-005	RF-013	RF-015	
RF-008	<i>RNF-006</i>	<i>RNF-005</i>	
RF-010		<i>RNF-007</i>	
<i>RNF-001</i>		<i>RNF-008</i>	
<i>RNF-002</i>		<i>RNF-010</i>	
<i>RNF-003</i>			
<i>RNF-004</i>			
<i>RNF-009</i>			

#### Punto 4: Análisis gestión de software

Este análisis lo componen tres elementos fundamentales: Tiempo, Costo y Alcance. Después de indagar, se ha concluido lo siguiente.

##### a) Tiempo

Basándonos en el tiempo disponible para la parte final del semestre– las fases del proyecto realizadas hasta la semana 11, además del contexto grupal– 4 estudiantes en etapa de aprendizaje de ingeniería de software, se han estimado los siguientes tiempos para las funcionalidades clasificadas como Must y Should; se han agrupado en módulos, los cuales, aunque similares a los procesos asociados, no necesariamente coinciden uno-a-uno:

Módulo	Funcionalidades	Tiempo estimado	Actividades principales
Autenticación y gestión de usuarios	<ul style="list-style-type: none"> <li>RF-001</li> <li>RF-002</li> <li>RF-003</li> </ul>	6-7 días	<ul style="list-style-type: none"> <li>Implementación de registro e inicio de sesión de usuarios</li> <li>Gestión de roles</li> </ul>
Gestión de asignaciones: Asignación de proyectos	<ul style="list-style-type: none"> <li>RF-005</li> <li>RF-008</li> </ul>	4-5 días	<ul style="list-style-type: none"> <li>Creación y asignación de proyectos por parte del profesor</li> <li>Consulta de detalles</li> </ul>
Gestión de asignaciones: Entrega de proyectos	<ul style="list-style-type: none"> <li>RF-010</li> <li>RF-012</li> <li>RF-013</li> </ul>	6-7 días	<ul style="list-style-type: none"> <li>Gestión de entregas finales de proyectos</li> <li>Implementación y gestión de archivos para entregas y materiales</li> </ul>
Creación de clases, registro de retroalimentación	<ul style="list-style-type: none"> <li>RF-004</li> <li>RF-011</li> </ul>	5-6 días	<ul style="list-style-type: none"> <li>Crear sistema de asociación de clase-profesor</li> <li>Implementación de interfaz de comentarios</li> </ul>
Usabilidad, optimización y seguridad	<ul style="list-style-type: none"> <li>RNF-001</li> <li>RNF-002</li> <li>RNF-003</li> <li>RNF-004</li> <li>RNF-006</li> </ul>	8-10 días	<ul style="list-style-type: none"> <li>Implementación de cifrado de contraseñas y validaciones de seguridad</li> <li>Diseño de aplicación responsiva y fácil de usar</li> <li>Optimización para múltiples usuarios en simultánea</li> </ul>
Documentación, pruebas y ajustes finales	<ul style="list-style-type: none"> <li>RNF-009</li> </ul>	6-7 días	<ul style="list-style-type: none"> <li>Realizar Testing e integración</li> <li>Corrección de errores, optimización</li> <li>Redacción de</li> </ul>

			documentación faltante
--	--	--	------------------------

**b) Costo**

La estimación de costo de desarrollo de software se ha realizado a modo semanal, teniendo en cuenta ciertos factores.

*Nota: para precios en Dólares Estadounidenses (USD), su estimación equivalente en Pesos Colombianos (COP) se ha realizado utilizando el tipo de cambio con corte a 30 de enero de 2025, siendo este \$1(USD) = \$4176(COP).*

**i) Sueldos de desarrolladores:**

Rol/Experiencia/Especialidad	Salario promedio (COP) mensual	Equivalente en USD
Desarrollador Junior	\$2'200.000	\$530
Desarrollador semi-Senior	\$3'610.854	\$870
Desarrollador Senior/Arquitecto	\$5'400.000	\$1.301
Full-Stack	\$5'400.000	\$1.301
Back-End	\$5'400.000	\$1.301
Front-End/UX	\$4'300.000	\$1.036
Web	\$4'300.000	\$1.036
Tester	\$2'600.000	\$626

Fuentes:

- <https://platzi.com/tutoriales/3208-programacion-basica/24348-descubre-cuanto-gana-un-programador-en-mexico-colombia-argentina-y-ee-uu-en-2023/>
- <https://evalart.com/es/blog/cuanto-ganan-los-desarrolladores-en-america-latina/>
- [https://www.glassdoor.com.ar/Sueldos/bogota-colombia-qa-tester-sueldo-SRCH\\_IL.0%2C15\\_IM1064\\_KO16%2C25.htm](https://www.glassdoor.com.ar/Sueldos/bogota-colombia-qa-tester-sueldo-SRCH_IL.0%2C15_IM1064_KO16%2C25.htm)
- <https://co.indeed.com/career/tester/salaries>

**ii) Servicios en la nube:**

Se han utilizado estimados que se aproximan a las necesidades y alcance de este proyecto. Se incluyen servicios de API, Almacenamiento e Integración Continua para esta estimación.

Servicio	Precio en USD mensual	Equivalente en COP
Microsoft Azure	\$21	\$87.159
Google Cloud	\$22.67	\$94.090
AWS	\$21.94	\$91.060

Fuentes:

- <https://azure.microsoft.com/es-es/pricing/calculator/>
- <https://cloud.google.com/products/calculator>

- <https://calculator.aws/>

### iii) Herramientas externas

Estas herramientas incluyen APIs, Frameworks y Software de Desarrollo. En el caso de las APIs, muchas ofrecen niveles gratuitos con limitaciones en el uso, a su vez que los planes de pago varían según el proveedor y cantidad de solicitudes. En cuanto a las librerías, la mayoría son de código abierto y gratuitos (FOSS), aunque algunas pueden requerir licencias comerciales según su uso.

En cuanto al software de desarrollo, para el alcance de este proyecto, se tienen tres opciones principales viables:

IDE	Tipo de Licencia	Costo USD/COP	Notas
Microsoft Visual Studio Code	Privada y comercial	Gratuita	Permite uso privado y comercial
Microsoft Visual Studio IDE	Profesional (Empresa)	\$45 USD/\$186.768 COP por mes	Licencia por usuario + plan básico de Azure DevOps
JetBrains PyCharm Professional	Profesional (Organizaciones)	\$24.90/\$103.345 COP por mes	Licencia por usuario

Fuentes:

- <https://code.visualstudio.com/docs/supporting/faq>
- <https://visualstudio.microsoft.com/es/vs/pricing/?tab=business>
- <https://www.jetbrains.com/es-es/pycharm/buy/?section=commercial&billing=monthly>

Con estos factores en consideración, a continuación se estiman los potenciales costos mensuales y semanales del desarrollo para este proyecto:

Estimación de costos de personal de desarrollo			
Roles de Desarrollador	Costo Mensual c/u (USD/COP)	Costo Semanal c/u (USD/COP)	Notas
Desarrollador Junior	\$450 / \$1'872.174	\$113 / \$470.123	Equipo de 4 desarrolladores para este proyecto.
	Costo Mensual total (USD/COP)	Costo Semanal total (USD/COP)	
	\$1.800 / \$7'488.695	\$450 / \$1'872.174	
Estimación de costo de servicios en la nube y software de desarrollo			
Servicios y programas	Costo Mensual (USD/COP)	Costo Semanal (USD/COP)	Notas
AWS	\$50 / \$208.025	\$13 / \$54.087	Haciendo una estimación

Estimación de costos de personal de desarrollo			
			de precio más conservadora.
Microsoft Visual Studio Code	Gratuito (\$0 / \$0)	Gratuito (\$0 / \$0)	Software a utilizar en el proyecto.
Total de estimaciones de Costo			
	Costo Mensual (USD/COP)	Costo Semanal (USD/COP)	Notas
<b>TOTAL</b>	\$1.850 / \$7'696.715	\$463 / \$1'926.259	Los centavos (decimales) de cada moneda han sido redondeados.

### c) Alcance

De manera general, el objetivo de este proyecto es proporcionar una plataforma funcional para la gestión, asignación, seguimiento, entrega y retroalimentación de proyectos en un entorno académico.

Los límites establecidos en este módulo del proyecto, dividen las funcionalidades en dos grupos: las que están incluidas en el MVP y las cuales quedan fuera del alcance del proyecto.

Las funcionalidades incluidas en el MVP incluyen las funcionalidades clasificadas como Must (M) y Should (S); aseguran que el sistema sea funcional para los usuarios destinados (estudiantes y profesores) en su versión inicial.

Funcionalidades incluidas en el MVP			
Módulo	ID	Funcionalidad	Proceso Asociado
Autenticación y gestión de usuarios	RF-001	Registro de usuarios	Gestión de usuarios
	RF-002	Inicio de sesión	Gestión de usuarios
	RF-003	Asignación de roles	Gestión de usuarios
Gestión de asignaciones: Asignación de proyectos	RF-005	Creación de proyectos	Gestión de asignaciones
	RF-008	Consulta de estado de proyectos	Gestión de asignaciones
Gestión de	RF-	Entrega final de proyectos	Gestión de

Funcionalidades incluidas en el MVP			
asignaciones: Entrega de proyectos	010		asignaciones
	RF-012	Subida de archivos por estudiantes	Archivos adjuntos
	RF-013	Subida de archivos por profesores	Archivos adjuntos
Creación de clases, registro de retroalimentación	RF-004	Creación de grupos (clases)	Gestión de usuarios
	RF-011	Retroalimentación de entregas de proyectos	Gestión de asignaciones
Usabilidad, optimización y seguridad	RNF-001	Compatibilidad multiplataforma web	Accesibilidad y usabilidad
	RNF-002	Usabilidad de la aplicación	Accesibilidad y usabilidad
	RNF-003	Protección de datos	Seguridad
	RNF-004	Seguridad de credenciales	Seguridad
	RNF-006	Capacidad para múltiples usuarios sin detrimento de rendimiento	Rendimiento
	RNF-009	Diseño modular	Escalabilidad

Las características que quedan fuera del alcance del MVP son las clasificadas como Could (C) y Won't (W). No son esenciales para el funcionamiento inicial del sistema, aunque pueden considerarse para implementar en futuras iteraciones.

Funcionalidades fuera del alcance del MVP			
Módulo	ID	Funcionalidad	Proceso Asociado
Gestión avanzada de proyectos	RF-006	Asignación de proyectos a estudiantes en específico	Gestión de asignaciones
	RF-007	Actualización del estado de avance del proyecto por parte del profesor	Gestión de asignaciones
	RF-009	Registro de avances en proyectos (como entregas intermedias) por	Gestión de asignaciones

Funcionalidades fuera del alcance del MVP			
		parte de estudiantes	
Mejoras en la experiencia del usuario	RNF-005	Respuesta rápida del sistema en conexiones estándar	Rendimiento
	RNF-007	Notificaciones relevantes de eventos	Notificaciones
	RNF-008	Registro de actividades	Registro de actividades
	RF-015	Consulta de progresos previos	Registro de actividades
Diseño y personalización	RF-014	Personalizar ícono de perfil	Archivos adjuntos
	RNF-010	Diseño visual limpio	Diseño
	RNF-011	Paleta de colores profesional, elementos llamativos	Diseño