

**Informe del Sprint: Taller del Taller de Gestión de Proyectos Jira**

**Equipo Sopas: Ricardo Chamorro, Oscar Stiven Muñoz, Diego Andres Polanco, Sebastian Erazo, Luis Manuel Rojas Correa**

**Curso: Proyecto Integrador 1**  
**Docentes: Ingri Lorena Jojoa, Milton Sarria, Claudia Castiblanco**

**Universidad ICESI**  
**Facultad de Ingeniería**  
**Departamento de Computación y Sistemas Inteligentes**

**Fecha: 01 de Septiembre de 2024**

## Objetivo del Informe

Este informe tiene como objetivo presentar los resultados y documentación desarrollada durante el sprint “0” del proyecto de creación de una aplicación de marketplace para freelancers de la empresa Encora. El sprint “0” se enfocó en la elaboración de toda la documentación necesaria para sentar las bases del desarrollo, incluyendo la definición y creación de las Historias de Usuario, la arquitectura inicial del sistema, el modelo relacional de la base de datos a utilizar y los mockups que ayudan a orientar al cliente de cómo será la aplicación. Este proceso inicial es crucial para garantizar una comprensión clara y compartida de los objetivos del proyecto, los alcances del sistema, y las expectativas del cliente antes de avanzar hacia las fases de desarrollo. Este informe detalla las actividades realizadas, las decisiones tomadas y proporciona una base documental sólida para los futuros sprints.

## Descripción del Sprint

Características	Descripción
<b>Sprint Goal</b>	El objetivo del Sprint 0 es establecer una base sólida para el desarrollo del proyecto, asegurando que todos los aspectos fundamentales, desde la planificación del backlog hasta el diseño del sistema y la organización del equipo, estén claramente definidos y documentados. Esto permitirá que el equipo inicie el desarrollo con una visión clara y bien estructurada, minimizando riesgos y facilitando la gestión del proyecto en sprints futuros.
<b>Fechas del Sprint</b>	<b>5 de Agosto - 1 de Septiembre</b>
<b>Resumen de Actividades Realizadas</b>	<p>Durante el Sprint 0, el equipo SOPAS completó las siguientes actividades:</p> <ol style="list-style-type: none"><li>1. <b>Product Backlog:</b> Se creó y estimó un backlog detallado en Jira, incluyendo historias de usuario con sus respectivos criterios de aceptación.</li><li>2. <b>Modelo de Datos:</b> Se elaboraron el Diagrama Entidad-Relación (ERD) y el Modelo Relacional que definen la estructura de la base de datos.</li><li>3. <b>Diagrama de Clases UML:</b> Se desarrolló el diagrama de clases que describe la arquitectura del sistema.</li><li>4. <b>Diagramas de Secuencias:</b> Se diseñaron diagramas de secuencias para los cuatro procesos más relevantes del sistema.</li><li>5. <b>Mockups:</b> Se crearon mockups que representan la interfaz de usuario prevista, proporcionando una visión visual del producto final.</li><li>6. <b>Casos de Uso:</b> Se actualizaron y refinaron los diagramas</li></ol>

	<p>de casos de uso para asegurar que cubran todas las funcionalidades esenciales.</p> <p>7. Seguimiento Diario (Daily Scrum): Se documentó el progreso diario del equipo, abordando obstáculos y planificando los próximos pasos.</p> <p>8. Retrospectiva del Sprint 0: Se llevó a cabo una retrospectiva para reflexionar sobre lo que funcionó bien, lo que se puede mejorar, y cómo se abordarán los próximos sprints.</p>
--	---

Enlace: [Historias de usuario con criterios de aceptación](#)

Actividades Planificadas y Realizadas

<input checked="" type="checkbox"/> SPII-20	Creacion de primera presentacion para el cliente sobre NexTask (Marketplace Freelancer)	DONE	26 AUG	LC
<input checked="" type="checkbox"/> SPII-14	Arquitectura: Desarrollar los diagramas de casos de uso asociados a la arquitectura en el Diagrama de Clases	DONE	01 SEP	
<input checked="" type="checkbox"/> SPII-15	Gestión de Calidad: Diseñar una primera versión del plan de pruebas asociados al proyecto	DONE	01 SEP	SG
<input checked="" type="checkbox"/> SPII-9	Requerimientos: escribir dudas sobre los requerimientos en el SRC	DONE	25 AUG	
<input checked="" type="checkbox"/> SPII-10	Requerimientos: identificar y anotar las principales ambigüedades encontradas en el SRC para presentarlas al cliente	DONE	25 AUG	OR
<input checked="" type="checkbox"/> SPII-8	Requerimientos: Analizar la abstracción y comprensión de los requerimientos del SRC. Buscando dar una mejor abstracción	DONE	01 SEP	
<input checked="" type="checkbox"/> SPII-12	Arquitectura: Desarrollar un diseño preliminar -diagrama de clases- sobre la arquitectura interna del proyecto	DONE	21 AUG	RM
<input checked="" type="checkbox"/> SPII-11	Base de Datos: Desarrollar un diseño preliminar de la base de datos relacional	DONE	19 AUG	LC
<input checked="" type="checkbox"/> SPII-7	Figma: Desarrollar modelo preliminar del R7	DONE	25 AUG	SG
<input checked="" type="checkbox"/> SPII-6	Figma: Desarrollar modelo preliminar del R6	DONE	25 AUG	OR
<input checked="" type="checkbox"/> SPII-10	Figma: Desarrollo modelo preliminar del R5 - Colaboracion y comunicacion	DONE	01 SEP	RM

<input checked="" type="checkbox"/> SPII-16	Figma: Desarrollo modelo preliminar del R5 - Gestion de Proyectos e Hitos	DONE	25 AUG	SG
<input checked="" type="checkbox"/> SPII-4	Figma: Desarrollar modelo preliminar del R4	DONE	20 AUG	RM
<input checked="" type="checkbox"/> SPII-3	Figma: Desarrollar modelo preliminar del R3	DONE	25 AUG	LC
<input checked="" type="checkbox"/> SPII-2	Figma: Desarrollar modelo preliminar del R2	DONE	20 AUG	OR
<input checked="" type="checkbox"/> SPII-1	Figma: Desarrollar modelo preliminar del R1	DONE	25 AUG	OR
<input checked="" type="checkbox"/> SPII-17	Figma: Desarrollo modelo preliminar del R5 - Tareas y seguimiento de Progreso	DONE	26 AUG	RM
<input checked="" type="checkbox"/> SPII-19	Figma: Desarrollo modelo preliminar del R5 - Control de acceso y permisos	DONE	25 AUG	OR
<input checked="" type="checkbox"/> SPII-27	Creación de Diagramas de Secuencia: Contratación de Freelancer.	DONE	01 SEP	SG
<input checked="" type="checkbox"/> SPII-24	Creación de Diagrama de Secuencia: Creación de Usuario -Freelancer- y Creación de Proyecto	DONE	01 SEP	LC
<input checked="" type="checkbox"/> SPII-25	Creación de Diagramas de Secuencia: Creación de Tarea.	DONE	01 SEP	SG
<input checked="" type="checkbox"/> SPII-21	Creación de Diagramas Casos de Uso: Requerimiento R6	DONE	01 SEP	SG
<input checked="" type="checkbox"/> SPII-32	Creación de Diagramas Casos de Uso: Requerimiento R5	DONE	01 SEP	OR

<input checked="" type="checkbox"/>	SPII-26	Creación de Diagramas Casos de Uso: Requerimiento R4 y R7	DONE	01 SEP	RM
<input checked="" type="checkbox"/>	SPII-22	Creación de Diagramas Casos de Uso: Requerimiento R2 y R3	DONE	01 SEP	LC
<input checked="" type="checkbox"/>	SPII-23	Creación de Diagramas Casos de Uso: Requerimiento R1	DONE	01 SEP	OR
<input checked="" type="checkbox"/>	SPII-28	Creación de Historia de Usuario: Requerimiento 7	DONE	01 SEP	SG
<input checked="" type="checkbox"/>	SPII-31	Creación de Historia de Usuario: Requerimiento 6	DONE	01 SEP	CO
<input checked="" type="checkbox"/>	SPII-33	Creación de Historia de Usuario: Requerimiento 5	DONE	01 SEP	OR
<input checked="" type="checkbox"/>	SPII-29	Creación de Historia de Usuario: Requerimiento 4	DONE	01 SEP	RM
<input checked="" type="checkbox"/>	SPII-30	Creación de Historia de Usuario: Requerimiento 3	DONE	01 SEP	LC
<input checked="" type="checkbox"/>	SPII-34	Creación de Historia de Usuario: Requerimiento 2	DONE	01 SEP	SG
<input checked="" type="checkbox"/>	SPII-35	Creación de Historia de Usuario: Requerimiento 1	DONE	01 SEP	SG

## Tareas seleccionadas en el Sprint

*SPII-20: Creación de presentación para el cliente sobre NexTask (Marketplace Freelancer)*

*SPII-14: Arquitectura: Desarrollar los diagramas de caso asociados a la arquitectura en el Diagrama de Clases*

*SPII-15: Gestión de Calidad: Diseñar una primera versión del plan de pruebas asociados al proyecto*

*SPII-9: Requerimientos: Escribir dudas sobre los requerimientos en el SRS*

*SPII-10: Requerimientos: Identificar y anotar las principales ambigüedades encontradas en el SRS para presentarlas al cliente*

*SPII-8: Requerimientos: Analizar la abstracción y comprensión de los requerimientos del SRS. Buscando dar una mejor abstracción*

*SPII-12: Arquitectura: Desarrollar un diseño preliminar -diagrama de clases- sobre la arquitectura interna del proyecto*

*SPII-11: Base de Datos: Desarrollar un diseño preliminar de la base de datos relacional*

*SPII-7: Figma: Desarrollar modelo preliminar del R7*

*SPII-6: Figma: Desarrollar modelo preliminar del R6*

*SPII-10: Figma: Desarrollar modelo preliminar del R5 - Colaboración y Comunicación*

*SPII-16: Figma: Desarrollar modelo preliminar del R5 - Gestión de proyectos y Hitos*

*SPII-4: Figma: Desarrollar modelo preliminar del R4*

*SPII-3: Figma: Desarrollar modelo preliminar del R3*

*SPII-2: Figma: Desarrollar modelo preliminar del R2*

*SPII-1: Figma: Desarrollar modelo preliminar del R1*

*SPII-17: Figma: Desarrollar modelo preliminar del R5 - Tareas y seguimiento de proyecto*

*SPII-19: Figma: Desarrollar modelo preliminar del R5 - Control de accesos y permisos*

*SPII-27: Creación de Diagramas de Secuencia: Contratación de Freelancer*

*SPII-24: Creación de Diagramas de Secuencia: Creación de Usuario-Freelancer-y Creación de proyecto*

*SPII-25: Creación de Diagramas de Secuencia: Creación de Tarea*

*SPII-21: Creación de Diagramas Casos de Uso: Requerimiento R6*

*SPII-32: Creación de Diagramas Casos de Uso: Requerimiento R5*

*SPII-26: Creación de Diagramas Casos de Uso: Requerimiento R4 y R7*

*SPII-22: Creación de Diagramas Casos de Uso: Requerimiento R2 y R3*

*SPII-23: Creación de Diagramas Casos de Uso: Requerimiento R1*

*SPII-28: Creación de Historia de Usuario: Requerimiento 7*

*SPII-31: Creación de Historia de Usuario: Requerimiento 6*

*SPII-33: Creación de Historia de Usuario: Requerimiento 5*

*SPII-29: Creación de Historia de Usuario: Requerimiento 4*

*SPII-30: Creación de Historia de Usuario: Requerimiento 3*

*SP11-34: Creación de Historia de Usuario: Requerimiento 2*

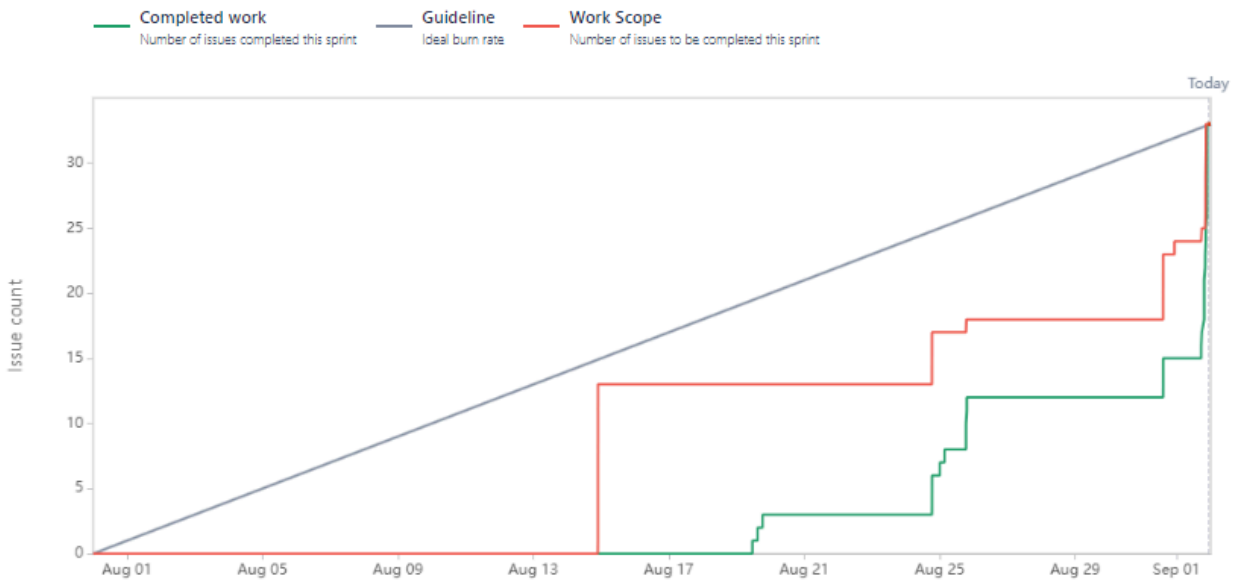
*SP11-35: Creación de Historia de Usuario: Requerimiento 1*

## Ejecución del Sprint y Monitoreo de Progreso

### Burnup report

El Burnup report es una herramienta indispensable para la visualización del progreso del equipo durante el sprint. El gráfico muestra la cantidad de puntos de historia completados a lo largo del sprint y el trabajo restante en comparación con el ideal o estipulado.

**Sprint goal** - Haber completado la fase de Diseño del Proyecto



#### Observaciones:

- **Inicio Lento (Aug 01 - Aug 13):**

Durante los primeros días (hasta Aug 13), no se observa ningún progreso en la línea verde. Esto sugiere que nuestro equipo no completó ninguna tarea en este período, lo cual podría deberse a diversos factores como una fase de planificación más prolongada o dificultades iniciales como por las parciales, jornadas y talleres.

- **Ajustes en el Alcance (Work Scope):**

La línea roja muestra varios incrementos, lo que indica que se agregaron nuevas tareas a lo largo del sprint. Esto podría reflejar un aumento en el alcance del trabajo, cambios en los requisitos y la identificación de nuevas tareas necesarias para completar la fase de diseño.

- **Progreso Acelerado (Aug 17 - Aug 29):**

A partir del 17 de agosto, se observa un avance considerable en la línea verde, con picos de trabajo completados. Este período muestra que el equipo comenzó a resolver tareas más activamente, logrando completar varias de ellas en un corto período de tiempo.

- **Finalización de Tareas:**

La línea verde continúa subiendo hasta el final del sprint, lo que indica que se mantuvo un ritmo constante de trabajo y se logró completar una buena cantidad de tareas hacia el final del sprint, acercándose al objetivo.

- **Resultado Final:**

Aunque el equipo ha hecho un progreso significativo y ha completado una cantidad importante de tareas, la línea verde no alcanza completamente la guideline gris, lo que sugiere que no todas las tareas planificadas se completaron dentro del tiempo previsto. Esto podría implicar que el equipo no logró cumplir con todos los aspectos del Sprint Goal o que se enfrentaron a desafíos que impidieron cumplir con el plan ideal.

## Sprint Burndown Report

**Sprint goal** - Haber completado la fase de Diseño del Proyecto



## Observaciones:

### 1. Inicio del Sprint (Aug 01 - Aug 13):

- Al inicio del sprint, no hay tareas registradas como restantes hasta aproximadamente el 13 de agosto, momento en el cual se agrega un conjunto significativo de tareas (el trabajo pendiente se eleva de 0 a más de 12 tareas). Esto indica que hubo una identificación tardía de tareas o una actualización del backlog en Jira.

### 2. Fluctuaciones en el Trabajo Restante:

- La línea roja muestra varias fluctuaciones significativas, ya que las tareas fueron agregadas o eliminadas durante el sprint. Esto fue resultado de cambios en el alcance, ajustes en las estimaciones, y una planificación de las tareas.

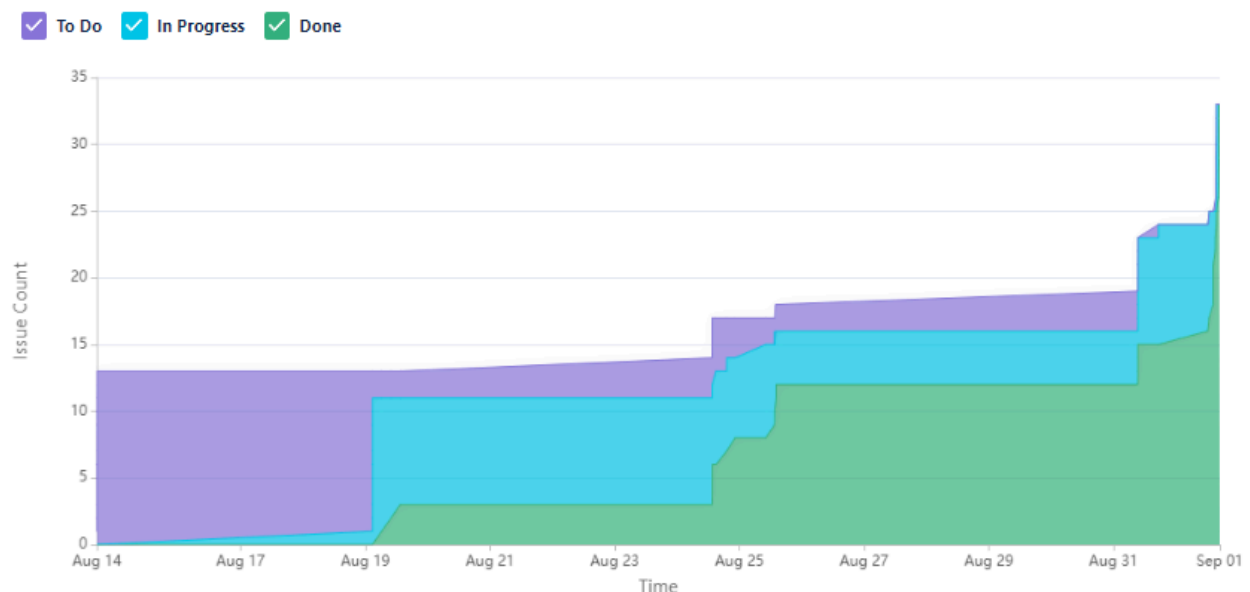
### 3. Progreso hacia el Final del Sprint (Ago 25 - Sep 01):

- A medida que el sprint avanza hacia su final, se observa una disminución considerable del trabajo restante, lo que nuestro equipo logró completar una cantidad significativa de tareas en los últimos días del sprint. Sin embargo, los picos indican que nuevas tareas siguieron apareciendo o que hubo correcciones que reintrodujeron trabajo pendiente.

### 4. Resultado Final:

- Al final del sprint, el trabajo restante se reduce a cero, lo cual es un indicativo positivo de que el equipo completó todas las tareas planificadas, aunque con variaciones en el camino. Sin embargo, la presencia de picos y caídas en el trabajo pendiente es porque el sprint tuvo ajustes dinámicos en su alcance.

## Cumulative Flow Diagram



## Observaciones:

### 1. Tareas en 'To Do' (Morado):

- El área morada muestra un incremento a lo largo del tiempo, especialmente hacia el final del sprint. Esto es por las nuevas tareas que fueron añadidas continuamente durante el sprint, incrementando la carga de trabajo pendiente.

### 2. Tareas 'In Progress' (Azul):

- El área azul también crece, lo que indica que a medida que avanzaba el sprint, más tareas pasaron del estado 'To Do' al estado 'In Progress'. Este crecimiento muestra un flujo constante de trabajo, aunque parece que algunas tareas se mantuvieron en progreso sin ser completadas inmediatamente.

### 3. Tareas Completadas 'Done' (Verde):

- El área verde crece de manera constante, lo cual es positivo, ya que indica que el equipo estuvo completando tareas a lo largo del sprint. Hacia el final, hay un aumento notable, lo que indica que una cantidad significativa de tareas fue completada en los últimos días del sprint.

### 4. Incremento al Final del Sprint:

- Hacia el final del sprint, hay un notable aumento en todas las áreas (To Do, In Progress, y Done). Esto es porque se añadieron muchas tareas nuevas en los últimos días, y el equipo trabajó intensamente para completarlas.

## Revisión y Reflexión del Sprint

### [Sprint review](#)

Hallazgo	Descripción
Creación de Historias de Usuario	Las tareas SPII-28, SPII-31, SPII-33, SPII-29, SPII-30, SPII-34, SPII-35 involucran la creación de historias de usuario para diferentes requerimientos de NexTask, describiendo de manera clara y detallada las necesidades y expectativas de los usuarios de la plataforma.
Desarrollo de Modelos en Figma	Las tareas SPII-7, SPII-6, SPII-10, SPII-16, SPII-4, SPII-3, SPII-2, SPII-1, SPII-17, y SPII-19 se centraron en desarrollar modelos preliminares de los requerimientos del sistema en Figma, abarcando aspectos críticos para la experiencia de usuario en NexTask, como la colaboración, gestión de proyectos, tareas, control de accesos, y más.
Actas de Reunión Documentadas en Jira	Permitió mantener la comunicación clara y



	resolver impedimentos rápidamente
Fortalezas	Buena comunicación y coordinación entre los miembros del equipo, uso efectivo de Jira para gestionar las tareas y seguimiento del progreso.
Áreas de Mejora	Reducción de la inactividad al inicio del sprint y mejor estimación de los tiempos para tareas futuras.
Planes de Acción	Implementar técnicas de priorización de tareas más robustas y mejorar la planificación para evitar retrasos en el futuro.
Creación de Diagramas de Secuencia	Se modelaron la secuencia de 4 procesos fundamentales en el programa para obtener una mejor claridad sobre qué es lo que, paso a paso, sucede
Diseño de bases de datos	Se diseñaron las bases de datos relacionales para maximizar la eficiencia de las consultas y una mejor abstracción del problema
Diagramas de casos de uso	Se realizaron diferentes casos de uso de todas las historias de usuario teniendo representación de lo que quieren los usuarios.
Requerimientos del Proyecto	Las tareas SPII-9, SPII-10, y SPII-8 involucraron el análisis de los requerimientos del SRS de NexTask, identificando ambigüedades, dudas, y abstracción para asegurar una mejor comprensión y precisión en el desarrollo de esta plataforma de freelancers.
Gestión de Calidad (pruebas)	La tarea SPII-15 se dedicó a diseñar una primera versión del plan de pruebas para NexTask, con el fin de asegurar que todas las funcionalidades del marketplace se implementen de acuerdo con los estándares de calidad requeridos.

## Diagrama de clases

### [Enlace](#)

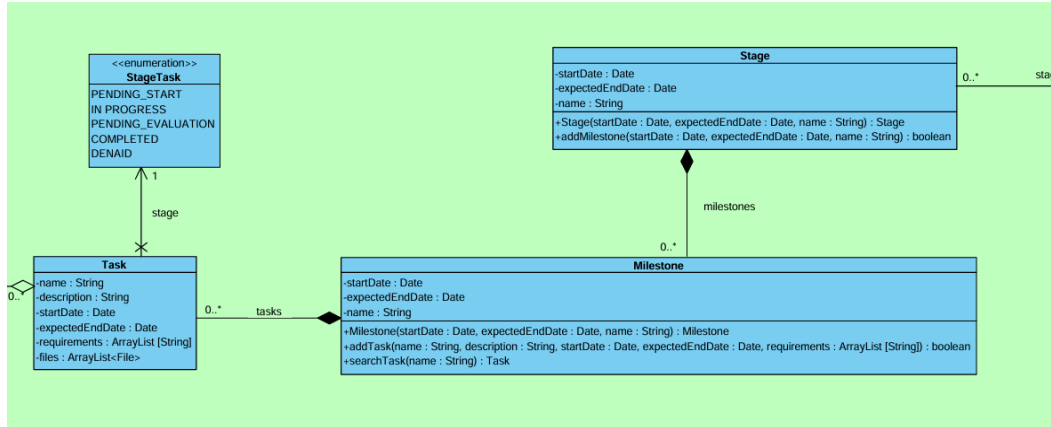
En el diagrama de clases se representan todas las entidades necesarias para la realización del programa, sus relaciones, métodos y atributos.

Controller
<pre> +Controller() : Controller +registerFreelancer(firstName : String, lastName : String, email : String, id : String, phone : String, profilePhoto : Image, cityOfResidence : String, country : String, typeQuestion : String, answerQuestion : String) : boolean +registerClient(RepresentativesFirstName : String, RepresentativesLastName : String, CompaniesName : String, email : String, companyID : String, phone : String, profilePhoto : Image, country : String, bussinessBranch : String, image : Image) : boolean +initiateWithGmail(email : String) : boolean +resetPasswordCode(email : String, answerCode : String) : boolean +resetPasswordQuestion(answer : String) : boolean +checkSkillAlreadyExist(skillName : String) : boolean +addSkill(skillName : String) : boolean +addNewProject(name : String, description : String, budget : double, startDate : Date, expectedEndDate : Date, requirements : String, ubicationLink : String) : boolean +searchProjectByBudget(minBudget : double, maxBudget : double, myProject : boolean) : ArrayList&lt;Project&gt; +watchTimeLineProjectFreelancer(nameProject : String) : Stage +watchTimeLineProjectClient(nameProject : String) : Stage +selectProject(name : String) : Project +changeTaskStatus(projectName : String, statusName : String, newStatus : int) : boolean +startChat(otherUserName : String) : boolean +searchUser(userName : String) : User +sendMessage(destinataryName : String, text : String, file : File) : boolean +searchTextInChats(text : String) : Chat +searchChat(otherUserName : String) : Chat +eraseChat(otherUserName : String) : boolean +createProject(name : String, description : String, budget : double, startDate : Date, expectedEndDate : Date, requirements : String, ubicationLink : String) : boolean +checkLabelExist(name : String) : boolean +addLabelToProject(project : Project, nameLabel : String) : boolean +searchProject(nameProject : String) : Project +addMilestones(projectName : String, startDate : Date, expectedEndDate : Date, name : String) : boolean +addTask(milestone : Milestone, name : String, description : String, startDate : Date, expectedEndDate : Date, requirements : ArrayList [String]) : boolean +assignTaskToUser(freelancerName : String, task : Task) : boolean +generateInformProject(nameProject : String) : File +validateTaskAsDone(nameProject : String, nameTask : String, aprove : boolean) : boolean +generateLinkToVideoCall() : String +addPayment(projectName : String, destinatary : User, image : Image, title : String, amount : double, expectedDate : Date, frequency : String) : boolean +pay(payment : Payment) : boolean +addReview(userName : String, text : String, stars : int, date : Date, publicName : String, email : String) : boolean +SearchProject(name : String) : boolean +addStage(startDate : Date, expectedEndDate : Date, name : String) : boolean </pre>

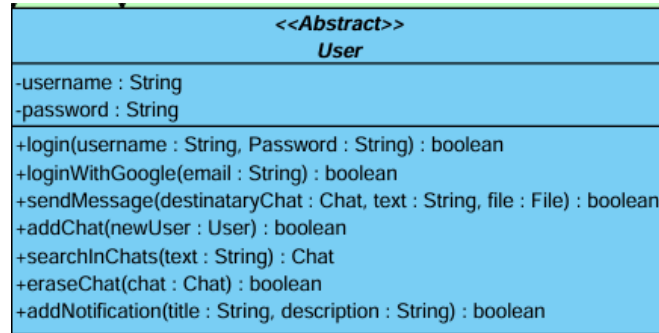
### Entre las clases están:

- Controladora
- Proyecto
- Usuario (abstracto)
- Freelancer
- Cliente
- Notificación
- Reseña (del freelancer)
- Pago
- Chat
- Mensaje
- Etapa (del proyecto)
- Hito
- Tarea
- Precedente (después se explica a que hace referencia)

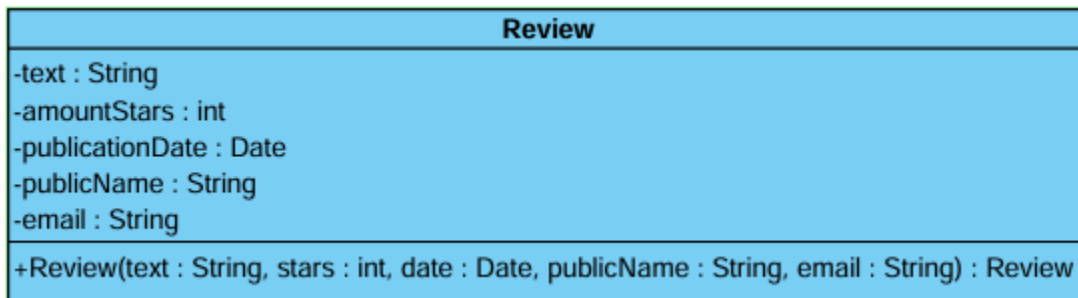
Adicionalmente se ven representadas la relación con las apis necesarias.



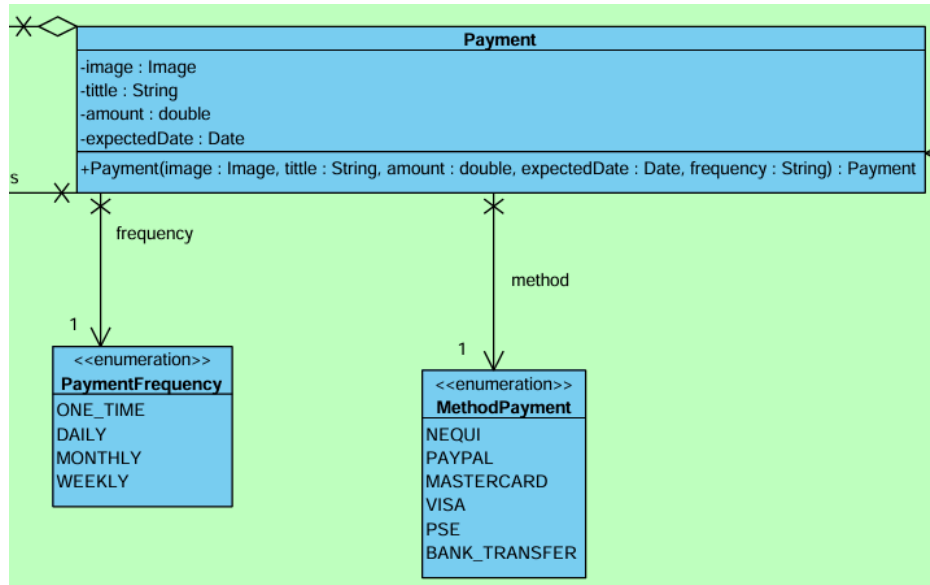
El hito se representa como un objetivo que se realiza mediante la finalización de sus actividades, un ejemplo sería, en un proceso de software, la etapa de diseño, una de diagramado: parte en la que se realizan el diagrama de clases, de secuencia y caso de uso. Cada uno es una tarea pero juntas forman un hito.



El método para realizar el login se realiza en la clase abstracta usuario, se creó esta clase en caso tal de que en un futuro existan más tipos de usuario.



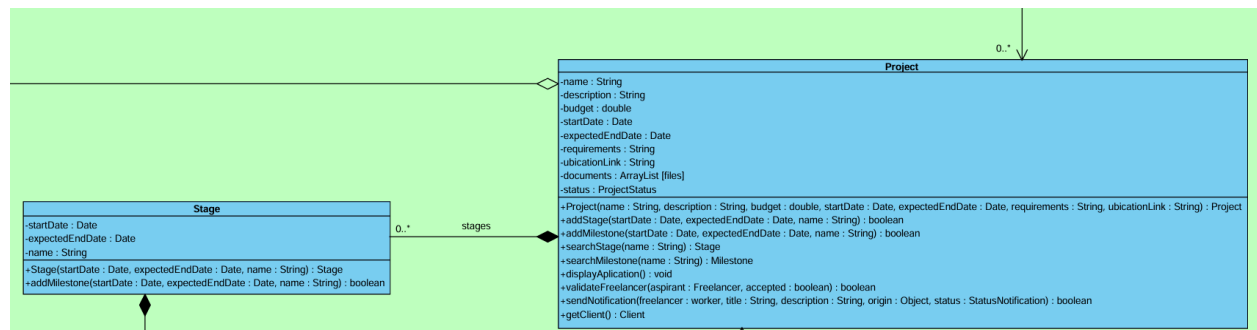
Las reseñas, aparte de incluir un sistema de 5 estrella, se realizan de manera anónima por lo que cualquiera podrá realizar una reseña sin dejar rastro por lo que no se guarda como tal la referencia al usuario que escribió la reseña, inclusive tiene un nombre “anónimo” que es el que se muestra en la reseña.



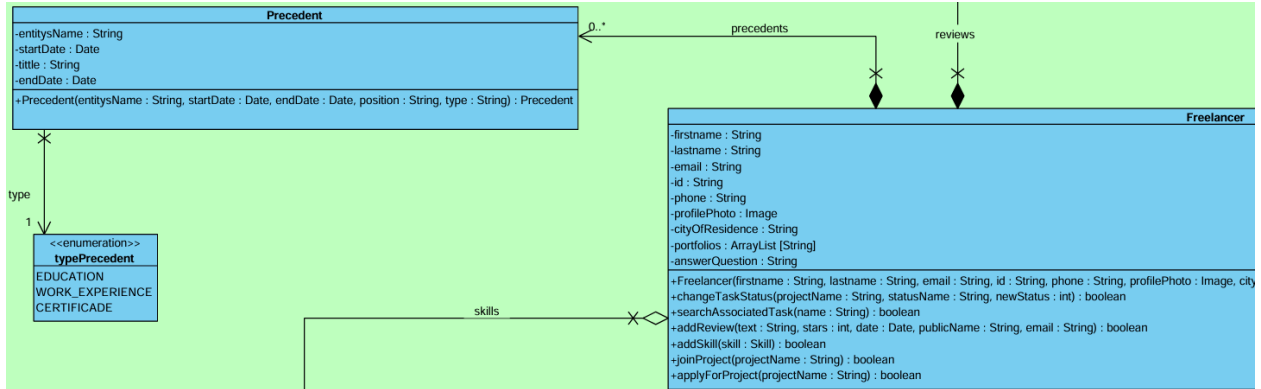
Los métodos de pago son los mismos que en el SRC: Nequi, transferencia bancaria, pse, tarjeta visa, tarjeta mastercard y Paypal. Al registrar un pago, se tiene la opción de crear un pago “periódico”, esto es que se crea un pago y se vuelve a crear cada cierto tiempo. Esta funcionalidad se espera que se use con frecuencia, en especial en caso de salarios.

Para guardar el rol de cada freelancer en cada proyecto se creó una clase intermedia entre el proyecto y el freelancer que se llama “Worker” que guarda el rol y los permisos que éste posee.

Para acceder más rápidamente a la información de cada usuario y la facilidad de cada método se creó un atributo en la controladora que se llama “usuario actual”, así no es necesario realizar logins, verificaciones, o búsquedas cada vez que se llame un método.



La etapa del proyecto son simples divisiones temporales, por nombre, o por fase, cada cliente decide cómo dividirlo.



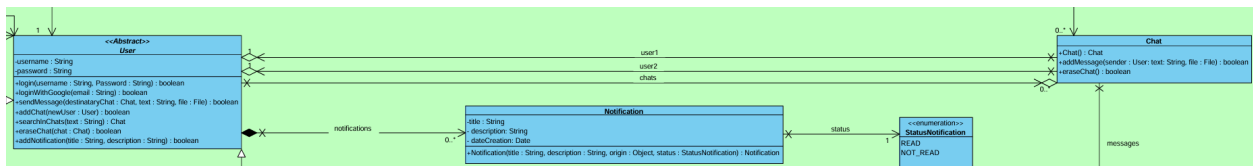
Se generalizó todos los elementos que pueden hacer a un freelancer más “llamativo” para el cliente. Ya sea un nivel académico, una experiencia laboral, un certificado, será llamado un precedente. Cada precedente tiene fecha de origen, de fin, el nombre de la entidad, el nombre del rol o título que obtuviste, y el tipo de precedente

Para la parte de recuperar la contraseña del usuario mediante una pregunta de seguridad, se tienen tres opciones preliminares:

- Nombre de la primera mascota
- Nombre del primer auto
- Fecha de nacimiento

Se tiene previsto que haya más opciones en un futuro.

Algunas enumeraciones como el país de residencia, el país de operaciones de la empresa, los lenguajes del freelancer y ramas de negocio. No se especificaron, o bien, por la gran cantidad de opciones, y en especial el último también por indecisión a la hora de escoger qué categorías son diferentes a otras.



El chat tiene la referencia a ambos usuarios para evitar tener que realizar la búsqueda en todo momento. El mensaje posee el atributo “archivo” por lo que se pueden enviar archivos mediante el chat.

## Modelo de Datos: Diagramas Entidad Relación y Modelo Relacional

### DataBaseIntegrador.pdf

Para el diseño de la Base de Datos de NexTask, se ha decidido realizar un modelo relacional en tercera forma normal, asegurando que la estructura de la base de datos minimiza la redundancia entre la información y se mantenga una integridad referencial, optimizando el almacenamiento y eficiencia de las consultas. El esquema incluye entidades principales como users, freelancer, projects, proposals, activities, y milestones, cada una definida con su respectiva clave primaria.

Las relaciones entre estas entidades están claramente delineadas mediante claves foráneas(FK), permitiendo conexiones como las de freelancers con proposals y users con companies.

Además, se integran tablas adicionales como skills y notifications para enriquecer la funcionalidad de la plataforma, ofreciendo personalización en perfiles y una gestión de comunicaciones eficiente. Este diseño robusto soporta la interacción dinámica entre freelancers, clientes y sus proyectos, asegurando escalabilidad y rendimiento en el sistema.

**Mockups:**

**Enlace del figma:**

<https://www.figma.com/design/kIjJ3CXCTbSzv6icgBqcKB/SOPAS-PROYECTO-INTEGRADOR?node-id=21-1197&t=xFdMOHI27axZFNf6-1>

**Diagramas de Casos de uso**

## **1. Registro y Acceso de Usuarios**

Este diagrama refleja las operaciones de registro y acceso que facilitan a los freelancers y clientes crear y gestionar sus cuentas en la plataforma. Los usuarios pueden registrarse con datos personales o usando su cuenta de Google, y tienen la opción de recuperar sus contraseñas mediante preguntas de seguridad o correo electrónico. El sistema asegura que todos los datos se almacenen de manera segura, soportando integraciones como la API de Google para una experiencia de acceso simplificada.

## **2. Gestión de Perfiles de Usuario**

Este diagrama ilustra cómo los freelancers configuran y actualizan sus perfiles, incluyendo la gestión de habilidades, experiencias laborales y portafolios. Los clientes pueden ver estos perfiles al seleccionar freelancers para sus proyectos. El diagrama también muestra cómo los freelancers pueden ajustar sus perfiles para reflejar nuevas calificaciones y mejorar sus oportunidades dentro de la plataforma.

## **3. Publicación de Proyectos**

El diagrama describe las funcionalidades que permiten a los clientes crear y gestionar proyectos dentro de la plataforma. Los clientes pueden definir presupuestos, establecer plazos y actualizar la información del proyecto. Los freelancers pueden buscar y filtrar proyectos según criterios como el presupuesto y la relevancia, y gestionar sus postulaciones de manera eficiente. El sistema asegura que todos los participantes estén al tanto del estado actual de los proyectos.

## 4. Sistema de Mensajería

Este diagrama detalla cómo los usuarios se comunican entre ellos mediante el sistema de mensajería de la plataforma. Los usuarios pueden iniciar conversaciones, intercambiar mensajes en tiempo real y compartir archivos importantes. El sistema organiza el historial de chats y notifica a los usuarios sobre nuevos mensajes, facilitando una comunicación fluida y efectiva entre freelancers y clientes.

## 5. Gestión de Proyectos

Este diagrama se enfoca en cómo los usuarios supervisan y gestionan proyectos a través de la plataforma. Los clientes pueden definir y modificar hitos, asignar tareas y monitorear el avance general del proyecto. Los freelancers pueden marcar tareas como completadas o solicitar más tiempo cuando sea necesario. El sistema refleja estos cambios en la línea de tiempo del proyecto, manteniendo actualizada la información sobre los avances y retrasos.

## 6. Procesamiento de Pagos

El diagrama aborda las funcionalidades relacionadas con los pagos dentro de la plataforma. Los clientes pueden efectuar pagos de manera segura a través de varias opciones, como tarjetas de crédito o plataformas de pago digital. Además, el sistema permite configurar pagos recurrentes y notifica tanto a clientes como freelancers sobre el estado de las transacciones, asegurando que se cumplan los estándares de seguridad y privacidad de los datos financieros.

## 7. Notificaciones

Este diagrama explora cómo los usuarios reciben notificaciones sobre eventos importantes en la plataforma, como nuevos proyectos o actualizaciones de tareas. Los usuarios pueden personalizar sus preferencias de notificación y consultar un historial detallado para revisar alertas previas. El sistema garantiza que las notificaciones lleguen a tiempo y en el formato más conveniente para cada usuario.

Enlace Casos de uso: [CASOS DE USO](#)

Diagrama de Secuencias: 4 Procesos Relevantes de NexTask: Creación de Usuario Freelancer, Creación de Nuevos Proyectos,

[FreelancerUserCreation.pdf](#)

Se explica la secuencia del proceso de registro de un nuevo usuario freelancer en la plataforma. El flujo comienza con la llamada al método `registerFreelancer()`, que verifica si el usuario ya existe mediante `searchUser()`. Si el usuario ya está registrado, se envía un mensaje de error, "User Already Exists". En caso contrario, se procede a crear un nuevo objeto `Freelancer`. Posteriormente, el sistema verifica si las

habilidades del freelancer ya existen utilizando `checkSkillsAlreadyExist()`. Dependiendo del resultado, se envía un mensaje de confirmación o error ("Skill successfully created" o "Skill already exists") antes de agregar la habilidad con `addSkill()`. Este diagrama resalta la importancia de validar la existencia de usuarios y habilidades antes de realizar cualquier operación de creación, asegurando así la integridad y consistencia de los datos en la plataforma.

#### ■ NewProjectsCreation.pdf

Se ejemplifica el flujo para la creación de un nuevo proyecto y la asignación de etapas (Stage) y hitos (Milestone). El proceso se inicia con `createProject()`, que verifica la existencia del proyecto mediante `searchProject()`. Si el proyecto ya existe, se emite un mensaje de error ("Project Already Exists"). Si no existe, se crea un nuevo proyecto. Posteriormente, se añade una etapa al proyecto con `addStage()` tras confirmar su existencia con `searchProject()`. Finalmente, se agrega un hito (Milestone) asociado a una etapa del proyecto utilizando `addMilestone()`, y se valida la existencia del proyecto y la etapa correspondientes. Este diagrama enfatiza la secuencia lógica de creación y verificación necesaria para garantizar que los proyectos, etapas y hitos se gestionen adecuadamente, asegurando que todas las operaciones se realicen en el contexto correcto.

#### ■ CreateTask.pdf


Este diagrama de secuencia describe el flujo de acciones necesarias para la creación de una nueva tarea dentro de un proyecto. El proceso comienza con la visualización de un hito existente (`displayMilestone view`) y la posibilidad de editar los hitos (`editMilestones()`). Una vez que los hitos están definidos, se procede a añadir nuevos hitos al proyecto (`addMilestones()`), seguido de la actualización de la vista de hitos (`newMilestones view`). A continuación, se crea una nueva tarea (`newTask()`) mediante la función `addTask()`. Una vez que la tarea es creada, se notifica al freelancer asignado utilizando `notifyFreelancer()`, que envía una notificación con `sendNotification()`. El flujo termina con la actualización de la vista para reflejar los cambios realizados (`updateView`). Este diagrama destaca la importancia de la secuenciación lógica en la creación de tareas, asegurando que todas las partes interesadas estén al tanto de los cambios y que la interfaz de usuario refleje adecuadamente el estado actual del proyecto.

#### ■ HireFreelancers.pdf

Este diagrama de secuencia muestra el proceso de contratación de freelancers a través de la plataforma. El flujo comienza con la solicitud de aplicación a un proyecto (`Request application`) y la visualización de aplicaciones existentes (`displayApplications`). Una vez que un freelancer decide unirse a un proyecto, inicia el proceso de aplicación con `joinProject()`, lo que genera una nueva aplicación (`newApplication()`). Esta aplicación es validada (`validate()`) y, dependiendo del resultado, puede ser guardada (`saveApplication()`) o generar un mensaje de error. Si la aplicación es válida, se envía una notificación al cliente mediante `sendNotification()`, y el cliente es informado del resultado de la aplicación (`clientNotified`). Este diagrama enfatiza la necesidad de un proceso de validación robusto para las aplicaciones y la importancia de la comunicación clara con el cliente durante el proceso de contratación.



## Documento de Seguimiento Diligenciado (dailys)

 Informe de seguimiento del Proyecto-Daily.docx

El Documento de Seguimiento Dailys, permite visualizar el progreso del equipo SOPAS durante el Sprint 0, que tuvo lugar del 5 de agosto al 1 de septiembre del 2024. Dentro del informe, los miembros del equipo detallan las actividades diarias de cada miembro del equipo, sus tareas planificadas, las acciones realizadas y los impedimentos encontrados.

### Principales puntos del informe:

1. **Planificación y Ejecución de Tareas:** Cada miembro del equipo reportó sus actividades diarias, que incluyeron la lectura y análisis del documento de requisitos (SRC), la creación de historias de usuario, la utilización de herramientas como Figma para el diseño de interfaces, y la planificación de prototipos y diagramas de clases.
2. **Retos y Obstáculos:** Los desafíos comunes enfrentados por el equipo incluyeron la falta de claridad en el documento de requisitos, dificultades técnicas con herramientas como Figma, problemas de tiempo debido a otras asignaturas, y la necesidad de aclaraciones adicionales por parte del cliente para algunos requerimientos.
3. **Colaboración y Comunicación:** El equipo destacó la colaboración efectiva entre los miembros para superar los desafíos, como ayudar en el diseño de diagramas o intercambiar ideas sobre la comprensión de los requisitos del proyecto. También se mencionaron discusiones sobre la mejora de la experiencia del usuario en los prototipos diseñados.
4. **Mejoras Necesarias:** Se identificó la necesidad de mejorar el manejo de ciertas herramientas y técnicas, como Figma y el diseño de aplicaciones monolíticas web, para la próxima fase del proyecto. También se sugirió una mejor organización y planificación de las tareas para evitar redundancias y mejorar la eficiencia.

## Retrospectiva Sprint 0

 SPRINT RETROESPECTIVE

Durante el Sprint, varios aspectos positivos fueron destacados por los miembros del equipo SOPAS. La colaboración efectiva y la comunicación asertiva fueron clave para la completitud de las tareas. Los miembros del equipo no solo cumplieron con sus responsabilidades asignadas, sino que también ofrecieron apoyo a otros compañeros cuando fue necesario. El equipo mostró un buen manejo tanto

teórico como práctico de las habilidades requeridas en el sprint, lo cual fue evidente en la entrega de incrementos de producto satisfactorios para la fase de diseño. La adaptabilidad y respuesta rápida a los cambios solicitados por el cliente también fueron aspectos positivos destacados, ya que se pudieron debatir a tiempo entre los integrantes del equipo desarrollador.

## Plan de Pruebas

📄 Plan de pruebas - SOPAS.pdf

El **Plan de Pruebas** para el proyecto "NextTask: Marketplace Freelancer" del equipo SOPAS se centra en garantizar la calidad, funcionalidad y seguridad de la plataforma a través de pruebas exhaustivas en varias etapas: planificación, pruebas por módulo, integración, experiencia de usuario, rendimiento, y seguridad. Utilizando herramientas como Selenium, Jira, y Django, el equipo de pruebas ejecutará casos de prueba detallados, gestionará incidentes y realizará un seguimiento constante del progreso. Las responsabilidades se distribuyen entre el Scrum Master, Product Owner y los Ingenieros de Pruebas, quienes colaboran para cumplir con los objetivos del plan, documentando resultados y asegurando que todos los aspectos críticos del sistema sean validados antes del despliegue.

## Conclusiones y Aprendizajes

A través de este sprint “0”, el equipo obtuvo una comprensión integral de las etapas iniciales de documentación necesarias para un desarrollo exitoso bajo la metodología SCRUM. Se desarrollaron competencias clave en la creación y gestión de documentación detallada, esencial para la planificación y ejecución de futuros sprints. Asimismo, el equipo mejoró su habilidad en la definición de Historias de Usuario, lo que es crucial para alinear las expectativas del proyecto con las necesidades reales del mercado. Se identificaron los siguientes puntos de mejora:

- Optimizar la estimación de tiempos y recursos, utilizando las experiencias de este sprint para mejorar las estimaciones en futuros sprints, garantizando una mejor asignación de recursos y tiempo.
- Continuar fomentando la comunicación abierta y la colaboración dentro del equipo para mantener un alto nivel de productividad y calidad en el trabajo entregado.
- Mejorar la calidad y el detalle de la documentación inicial, asegurando que todos los documentos, sean exhaustivos y claros para evitar ambigüedades durante las fases de desarrollo.

## Anexos

**Link del Jira:**

<https://oscarmunoz.atlassian.net/jira/software/projects/SPII/boards/69/backlog>