



V1.0.0 (22-08-2024) - Milton Loaiza

Historia de usuario: Como comunidad, queremos una plataforma donde se pueda publicar propiedades en arriendo, y los posibles arrendatarios puedan solicitar citas para verlas o alquilarlas. El objetivo es dar una herramienta optimizada para una gestión eficiente del proceso de arriendo, superando las limitaciones actuales de plataformas como Facebook Marketplace, no diseñadas específicamente para ello, o del tradicional voz a voz. El objetivo es que dicha solución se despliegue y cumpla con el ciclo de vida de desarrollo de software (Planeación, Diseño, Implementación, QA, Despliegue).

Para esto se requiere crear un conjunto de APIs en la tecnología NestJS que permitan:

- **API usuarios:**

Me debe permitir administrar usuarios, sus datos personales, roles (admin, cliente y dueño, un usuario puede tener múltiples roles). **Plus:** Validación de humanidad de los usuarios.

- **API solicitudes:**

Debe permitir generar una solicitud de ver o arrendar el inmueble, permitir al dueño aceptar la cita, rechazarla o proponer un nuevo horario, el cual debe notificarle al cliente y permitirle aprobar, rechazar o proponer otro nuevo y notificar a las partes, 2 horas antes de la cita pactada. **Plus:** Gestionar lista de muebles del cliente y calcular la idoneidad según el metraje del inmueble.

- **API chat:**

Me debe permitir establecer una comunicación entre cliente y dueños. **Plus:** El chat se debe integrar con IA para responder las preguntas comunes como tamaño, ubicación y demás, cuando la IA no sea capaz de responder debe enviarle notificación al dueño para que agregue dicha información.

- **API inmuebles:**

Me debe permitir gestionar los inmuebles con la mayor cantidad de datos relevantes posibles, incluido la multimedia (cargar archivos de imágenes y videos.)



Plus: Integrar con IA para llenar automáticamente campos como beneficios del sitio.

- **API notificaciones:**

Me debe permitir enviar emails y enviar mensajes por telegram que las otras APIs requieran, se debe encargar de gestionar las plantillas html y formatos de los mensajes. **Plus:** Crear Bot de telegram que publique en telegram los inmuebles menos vistos.

- **API integración lugares:**

Me debe permitir integrarme con la API de Google de places para obtener información de sitios cercanos al inmueble como colegios, restaurantes y demás.

Plus: Obtener otra metadata sobre el lugar de otras fuentes de información como noticias, eventos, etc.

- **API auth:**

Me debe permitir gestionar la autenticación de los usuarios a través de jwt. **Plus:** Implementar validación de x-api-key por cada plataforma backend y frontend.

- **API contratos:**

Me debe permitir generar un pdf con el contrato de arrendamiento predefinido y debe permitir realizar firma digital por los implicados. **Plus:** permitir al dueño del inmueble cargar su propio formato de contrato y validarlo con una IA si es aplicable a la ley del país.

Entregar evidencias progresivamente del proyecto:

a. Planeación y diseño: hasta el lunes 26 de agosto en la noche.

1. Diagramas UML (flujo y casos de uso)
2. Historias de usuario estimadas, refinadas y asignadas en Jira y que involucren tareas no solo de desarrollo sino de QA y demás.
3. Diagramas de arquitectura



4. Diagramas de modelos de datos (ER u otros)

5. Documentación en Confluence.

b. Implementación: hasta el viernes 30 de agosto en la noche

1. Uso de gitflow

2. Buen uso de NestJS

3. Desarrollo completo de HUs relacionadas a codificación

c. QA: hasta 03 de septiembre

1. Pruebas funcionales documentadas en Confluence

2. Pruebas unitarias con coverage de al menos el 30% (Plus no obligatorio)

d. Despliegue: hasta el 06 de septiembre

1. Despliegue en la nube de aplicativos.

2. Despliegue en la nube de persistencia.

3. Integración real con los demás microservicios.

Equipos: Cada equipo debe asignar un líder y avisarle al TL a más tardar hoy 23 de agosto por discord.

Api usuarios	Alejandro Ramírez Patiño	Api inmuebles	Alexis Soto Barreiro	Api auth	Andrés Felipe Hernández Córdoba
	Carlos Andrés Durán Bustamante		Damián Alejandro García Agudelo		David Alejandro Tirado Ledesma
	Diego Jaramillo Pérez		Hernán Darío Vergara Viana		María José Olivero Carrascal
	Erick Santiago Giraldo Pérez		José Manuel Bernal Yepes		Samuel Córdoba Henao
	Doris Daniela Mendoza Arbeláez		Mahicol Alejandro Gallego Valencia		Samuel Eliab Sotelo Parra
Api solicitudes	Alejandro Rivas López	Api notificaciones	Ana Isabel Alzate Henao	Api contratos	Christian Camilo Pabón Villanueva
	Christian Felipe López Sarrazola		Daniel Alejandro López Naranjo		Elver Noe Tunubala Yalanda
	Edson Esteban García Cuesta		Jhonatan Stiven Vergara		Emanuel Estrada Muñoz
	Francisco Alberto Aguirre Agudelo		Marlon Stiven Peña Naranjo		Cristhian Vásquez
	Isabela Ortega Vasco				
Api chat	Alejandro Vergara Zapata	Api integración lugares	Andrés David González Vanegas		
	Cristian Gabriel Ortega Saldarriaga		Daniel Alejandro Sánchez Ángel		
	Harold Medrano Emiliani		Julián Andrés Villacís Rodríguez		
	Juan Felipe Pulgarín Hernández		Lucas Aristizábal Giraldo		
	Luis Eugenio Rubio Romero		Matías Mesa		

Metodología ágil para utilizar:



www.riwi.io



301 732 53 27



Cl. 16 # 55 - 129



Scrum: Garantizar al menos ceremonias de planning, daily, sprint review y retrospective en todos los equipos.

Notas:

*Cada equipo que lo requiera deberá implementar el frontend que le corresponda, sin embargo, si no lo alcanzan a implementar entonces garantizar a través de documentación postman las pruebas de uso.

** El objetivo es que este proyecto salga a la realidad, después de finalizado realizaremos un sondeo para determinar la viabilidad, infraestructura y demás para que salga a producción y se pueda utilizar por la comunidad.

*** No pueden pasar a implementación sin que el TL aprueba el diagrama de arquitectura, el diagrama de modelo de datos, para ello deben aprovechar y reservar los espacios de entrenamiento.