



# **Predicción de Riesgos Financieros y Estimaciones Presupuestarias en Construcción Inmobiliaria con Modelos de Aprendizaje Automático**

Autor:

Ing. Danilo Simón Reitano Andrades

Director:

A definir (A definir)

*Esta planificación fue realizada en el curso de Gestión de proyectos  
entre el 1 de mayo de 2025 y el 30 de junio de 2025.*

## Índice

1. Descripción técnica-conceptual del proyecto a realizar. . . . .	5
2. Identificación y análisis de los interesados . . . . .	8
3. Propósito del proyecto . . . . .	8
4. Alcance del proyecto . . . . .	8
5. Supuestos del proyecto. . . . .	9
6. Product Backlog . . . . .	10
7. Criterios de aceptación de historias de usuario . . . . .	11
8. Fases de CRISP-DM . . . . .	12
9. Desglose del trabajo en tareas . . . . .	12
10. Diagrama de Gantt . . . . .	13
11. Planificación de Sprints . . . . .	14
12. Normativa y cumplimiento de datos (gobernanza) . . . . .	15
13. Gestión de riesgos . . . . .	16
14. Sprint Review . . . . .	17
15. Sprint Retrospective . . . . .	18

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	1 de mayo de 2025
1	Se completa hasta el punto 5 inclusive	12 de mayo de 2025

## Acta de constitución del proyecto

Mendoza, 1 de mayo de 2025

Por medio de la presente se acuerda con el Ing. Danilo Simón Reitano Andrades que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Predicción de Riesgos Financieros y Estimaciones Presupuestarias en Construcción Inmobiliaria con Modelos de Aprendizaje Automático” y consistirá en el desarrollo un modelo de predicción de préstamos y presupuestos de construcción. El trabajo tendrá un presupuesto preliminar estimado de 600 horas y un costo estimado de \$ 1500, con fecha de inicio el 1 de mayo de 2025 y fecha de presentación pública el 30 de noviembre de 2024.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg  
Director posgrado FIUBA

Martin Brambati  
Built Technologies

A definir  
Director del Trabajo Final

## 1. Descripción técnica-conceptual del proyecto a realizar

### Introducción, contexto y propuesta de solución

El presente trabajo surge a partir de una necesidad concreta identificada en el entorno laboral del autor en Built Technologies, empresa con sede en Nashville, Tennessee, Estados Unidos. Dicha empresa desarrolla una plataforma SaaS líder en la gestión de préstamos para la construcción. Trabaja con cientos de bancos y entidades financieras que otorgan financiamiento a personas buscando construir su casa, desarrolladores y contratistas para la ejecución de proyectos residenciales y comerciales. La plataforma permite monitorear el uso del préstamo, la ejecución del presupuesto y los avances del proyecto de forma integrada.

Uno de los principales desafíos operativos que enfrenta la empresa, y en general el sector de préstamos para construcción, es la dificultad para validar en etapas tempranas si el monto del préstamo solicitado por el cliente será suficiente para cubrir los costos del proyecto. Actualmente, esta validación se realiza mediante revisiones manuales de los presupuestos enviados, los cuales suelen estar desagregados en múltiples partidas (ej. cimentación, materiales, permisos, mano de obra, pisos, techos, terminaciones, etc.). Este proceso es altamente dependiente del criterio del analista y de las experiencias pasadas, sin una herramienta sistemática que permita predecir desvíos, subestimaciones o riesgos de insuficiencia presupuestaria. A menudo, estas deficiencias recién se manifiestan cuando el proyecto ya está en ejecución, generando sobrecostos, atrasos, renegociaciones contractuales o incluso abandono de obra.

En este contexto, la propuesta de esta tesis es el desarrollo de un sistema predictivo basado en modelos de aprendizaje automático que permita asistir de manera inteligente a los analistas de crédito en dos aspectos clave:

1. Estimar, a partir de la información disponible al momento de analizar el préstamo, si el monto solicitado será suficiente para cubrir el presupuesto completo del proyecto.
2. Predecir los costos esperados para cada una de las partidas presupuestarias del proyecto (por ejemplo: excavación, estructura, instalaciones eléctricas, techado, acabados), teniendo en cuenta características del proyecto como su ubicación, tipo, proveedor, constructor, superficie y otros factores históricos.

Este doble enfoque con distintas capas de análisis permitirá no solo detectar situaciones de riesgo financiero de manera temprana, sino también ofrecer recomendaciones concretas sobre los costos esperados. Todo esto en función de los datos históricos, ajustados al contexto de cada proyecto.

### Contexto y condiciones particulares del proyecto

Este proyecto se desarrollará en estrecha colaboración con el equipo de datos de Built Technologies. Se cuenta con acceso autorizado a un dataset anonimizado que incluye información detallada de más de 10 años de historial de proyectos, con datos por rubro presupuestario, tipo de préstamo, ubicación, resultados de ejecución y características del contratista y del proveedor. Por cuestiones de privacidad y cumplimiento normativo (SOC 2 y GDPR), los datos no contienen información sensible de clientes, y el trabajo se limita exclusivamente a información estructurada, descartando el uso de documentos escaneados o imágenes.

## Estado del arte y diferenciación de la solución

En términos generales, existen diversas aplicaciones de modelos de machine learning en el ámbito financiero, particularmente en la evaluación de crédito, detección de fraude y puntuación de clientes. Sin embargo, el uso de aprendizaje automático para prever costos de construcción y validar la suficiencia de préstamos basándonos en presupuestos históricos es aún una línea de investigación y desarrollo incipiente.

La mayoría de las soluciones actuales en el sector se apoyan en heurísticas basadas en precios unitarios, bases de datos estáticas por región o benchmarking manual entre proyectos. Esto tiene limitaciones evidentes: no contempla el contexto completo del proyecto ni aprende de los patrones reales de ejecución observados en los últimos años. Además, a medida que los proyectos modernos incrementaron en escala y complejidad, los métodos convencionales resultan insuficientes para capturar todas las variables que afectan los costos.

La regresión multivariada ha sido un enfoque base para estimar costos de construcción. Estos modelos asumen relaciones lineales entre los factores (como tamaño de la obra, calidad de los materiales, etc.) y el costo total. En escenarios relativamente simples, las regresiones pueden brindar estimaciones razonables: típicamente logran una precisión del orden de 75 %-80 % [1]. No obstante, en la mayoría de proyectos existen relaciones no lineales y dependencias complejas entre variables (economías de escala, influencias del mercado, interacciones entre diseño y método constructivo) que limitan la capacidad predictiva.

También se ha explorado la utilización de otro tipo de modelos como Random Forest y XGBoost. Estos enfoques han mostrado mejoras significativas en precisión frente a métodos tradicionales[2]. Por ejemplo, un estudio recopiló datos de 95 proyectos de edificios y implementó un modelo de Random Forest para predecir riesgos de sobrecostos.[3] Otro ejemplo relevante es un trabajo que utiliza XGBoost para seleccionar las variables más influyentes y estimar el costo de proyectos de edificación. [4]

En otros casos, se ha explorado la posibilidad de utilizar modelos de deep learning para estimación de costos en el desarrollo de proyectos. Algunas investigaciones han resultado en precisiones del 85 %-90 %, superior a la obtenida por modelos más simples. Esto se traduce en menores errores de predicción para proyectos complejos, aunque a costa de una mayor demanda de datos y capacidad computacional.

Al leer los resultados de dichos proyectos, se puede ver que las variables de mayor influencia son la escala del proyecto (metros cuadrados construidos, número de plantas), funcionalidad (edificio residencial, comercial, salud, etc.), así como especificaciones técnicas principales (tipo de climatización, sistema estructural, acabados exteriores e interiores, presencia de elementos especiales como ascensores, etc.), ubicación geográfica, año o época de construcción y tipo de contrato, entre otros.

La solución propuesta se diferencia en que:

- Utiliza modelos de aprendizaje supervisado entrenados sobre datos reales de ejecución y financiamiento, ajustando las predicciones al comportamiento histórico.
- Integra múltiples variables categóricas y numéricas, como ubicación geográfica, superficie construida, tipo de contratista y composición del presupuesto, ofreciendo una predicción contextualizada.
- Utiliza un volumen de datos mayor a los proyectos realizados hasta el momento.

- Incorpora técnicas de explicabilidad como SHAP (Shapley Additive Explanations) para interpretar por qué el modelo predice insuficiencia o sobre costo, facilitando su adopción por parte de analistas humanos.
- Ofrece tanto una salida binaria (¿es suficiente el préstamo?) como una salida regresiva multirrubro (estimación de costos por partida).

## Propuesta de valor e impacto esperado

La implementación de este sistema podría generar múltiples beneficios para Built Technologies y sus socios financieros:

- Reducir el porcentaje de proyectos con préstamos insuficientes, disminuyendo renegociaciones y sobre costos.
- Mejorar la eficiencia del análisis crediticio, apoyando a los analistas con una herramienta predictiva basada en datos.
- Aumentar la satisfacción del cliente final, evitando interrupciones en la ejecución del proyecto por errores de estimación.
- Detectar patrones de subestimación crónica en ciertos rubros o regiones, lo que podría informar futuras políticas de originación de préstamos.

El sistema se implementará inicialmente como un prototipo funcional con salida en formato tabular y visual, para luego ser integrado en el flujo de trabajo de análisis crediticio mediante una API o módulo dentro de la plataforma existente.

## Descripción funcional de la solución

La solución se compone de los siguientes bloques funcionales:

- **Módulo de extracción de datos:** Identificación, scrapping y unificación de datos.
- **Módulo de procesamiento de datos:** Limpieza, transformación y validación de los datos históricos estructurados.
- **Módulo de entrenamiento:** Entrenamiento de dos modelos: uno de clasificación binaria (suficiencia del préstamo) y otro de regresión multirrubro (predicción de costos por partida).
- **Módulo de inferencia:** Dado un nuevo proyecto con sus características, el sistema entrega predicciones de suficiencia y una tabla con los valores esperados por rubro presupuestario.
- **Módulo de visualización:** Presenta los resultados a través de gráficos, tablas y explicaciones interpretables, facilitando el uso por parte del equipo financiero.

En la figura 1 se presenta el diagrama en bloques del sistema. Se observa el flujo de datos desde la base histórica, pasando por las etapas de preprocesamiento, entrenamiento e inferencia, hasta la visualización de resultados por parte del analista financiero.

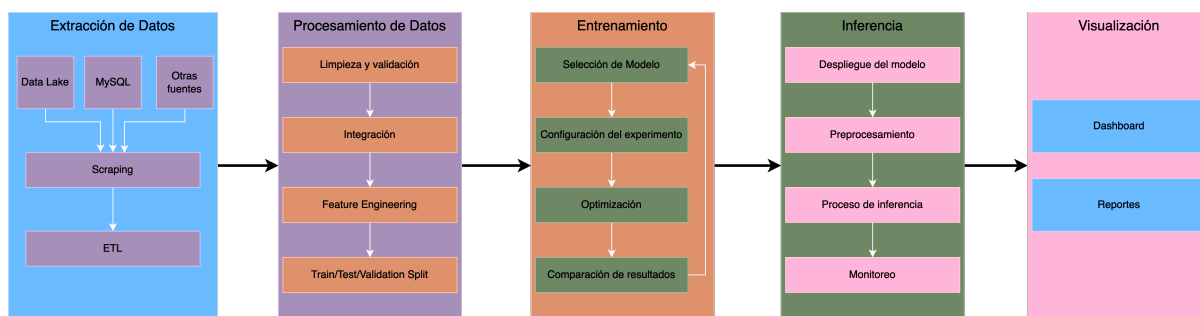


Figura 1. Diagrama en bloques del sistema.

## 2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Martin Brambati	Built Technologies	Engineering Manager
Responsable	Ing. Danilo Simón Reitano Andrades	FIUBA	Alumno
Colaboradores	Thomas Schlegel	Built Technologies	Distinguished Engineer
Orientador	A definir	A definir	Director del Trabajo Final
Opositores	-	Procore Technologies	-
Usuario final	-	Cientes de Built	-

- Orientador: a definir.
- Cliente: Martin Brambati tiene muchos años de experiencia en el puesto de manager y director de ingeniería. Su experiencia puede ser muy útil para el desarrollo del proyecto.
- Colaboradores: Thomas Schlegel tiene más de 8 años de experiencia a cargo del área de datos en Built Technologies, lo cual será de mucha ayuda a la hora de entender el dataset.
- Opositores: Procore Technologies proporciona una plataforma unificada de gestión financiera de proyectos de construcción. La empresa ya ha incorporado herramientas de inteligencia artificial, por la que tendría intenciones de que este proyecto no llegue a concluirse.

## 3. Propósito del proyecto

Desarrollar una solución basada en modelos de aprendizaje automático que permita asistir a los analistas financieros de Built Technologies en la evaluación de la suficiencia de préstamos otorgados para proyectos de construcción, así como en la estimación detallada de los costos asociados a cada una de las partidas presupuestarias. La solución busca anticipar riesgos de subfinanciamiento y sobrecostos mediante el análisis de datos históricos anonimizados, promoviendo decisiones más informadas, eficientes y escalables dentro del flujo de originación de créditos.

## 4. Alcance del proyecto

El presente proyecto incluye:



- Relevamiento y comprensión del problema de negocio asociado a la evaluación de suficiencia de préstamos y estimación presupuestaria en proyectos de construcción.
- Exploración, limpieza y transformación del conjunto de datos históricos anonimizados proporcionado por Built Technologies.
- Desarrollo de un pipeline de preprocesamiento de datos, que incluirá:
  - Tratamiento de valores faltantes.
  - Codificación de variables categóricas.
  - Normalización y estandarización de variables numéricas.
  - Balanceo de clases en caso de desbalance significativo en la variable objetivo (para el modelo de clasificación).
- Entrenamiento y validación de:
  - Un modelo de clasificación binaria para predecir si el préstamo solicitado es suficiente para cubrir el presupuesto total.
  - Un modelo de regresión multivariada para estimar los costos esperados por categoría presupuestaria (por ejemplo: cimentación, electricidad, techado, etc.).
- Evaluación de los modelos desarrollados mediante métricas adecuadas (F1-score, AUC-ROC, MAE, RMSE).
- Generación de explicaciones interpretables de las predicciones usando técnicas como SHAP o Feature Importance.
- Documentación del proceso completo y presentación de los resultados en un formato replicable y académico.

El presente proyecto no incluye:

- El desarrollo de una interfaz gráfica de usuario.
- La integración de los modelos desarrollados en los sistemas de producción de Built Technologies.
- La automatización completa del pipeline en un entorno de producción.
- La toma de decisiones finales sobre políticas de crédito, las cuales quedarán en manos del equipo financiero de la empresa.
- El análisis de documentos no estructurados.

## 5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- El dataset estructurado y anonimizado proporcionado por Built Technologies estará disponible desde el inicio del proyecto, y contará con la calidad y cantidad suficientes para el entrenamiento de modelos de aprendizaje automático.

- No se requerirá solicitar acceso a datos sensibles o confidenciales.
- El alcance del proyecto se mantendrá centrado en el desarrollo de modelos predictivos (clasificación y regresión), sin requerir su integración en producción o implementación de interfaces visuales para usuarios finales.
- Se dispondrá de al menos 8 horas semanales para dedicar al proyecto a lo largo del año calendario 2025, equilibrando responsabilidades laborales con los tiempos del posgrado.
- Se contará con acceso continuo a recursos computacionales adecuados (principalmente Jupyter notebooks, Python y bibliotecas de ML como scikit-learn, XGBoost, pandas y SHAP), sin necesidad de infraestructura especializada ni servicios cloud pagos adicionales.
- Se contará con la disponibilidad del director del trabajo final para realizar revisiones metodológicas periódicas y seguimiento académico del progreso del proyecto.

## 6. Product Backlog

El Product Backlog debe organizarse en cuatro **épicas** fundamentales del proyecto. Cada épica debe contener al menos dos historias de usuario que describan funcionalidades clave.

El Product Backlog debe permitir interpretar cómo será el proyecto y su funcionalidad. Se deben indicar claramente las prioridades entre las historias de usuario y si hay alguna opcional.

Las historias de usuario deben ser breves, claras y medibles, expresando el rol, la necesidad y el propósito de cada funcionalidad. También deben tener una prioridad definida para facilitar la planificación de los sprints.

Cada historia de usuario debe incluir una ponderación en *Story Points*, un número entero que representa el tamaño relativo de la historia. El criterio para calcular los Story Points debe indicarse explícitamente.

Las historias deben seguir el formato: “*Como [rol], quiero [tal cosa] para [tal otra cosa]*”.

Las épicas deben estructurarse de la siguiente forma:

- **Épica 1**
  - HU1
  - HU2
- **Épica 2**
  - HU3
  - HU4
- **Épica 3**
  - HU5
  - HU6
- **Épica 4**

- HU7
- HU8

### **Reglas para definir historias de usuario:**

- Ser concisas y claras.
- Expresarlas en términos cuantificables y medibles.
- No dejar margen para interpretaciones ambiguas.
- Indicar claramente su prioridad y si son opcionales.
- Considerar regulaciones y normas vigentes.

## **7. Criterios de aceptación de historias de usuario**

Los criterios de aceptación deben establecerse para cada historia de usuario, asegurando que se cumplan las condiciones necesarias para que la funcionalidad sea validada correctamente.

Cada historia debe tener criterios medibles, específicos y verificables. Deben permitir validar que se cumple con las necesidades del usuario.

Se estructuran de forma análoga a las épicas del backlog:

- **Épica 1**
  - Criterios de aceptación HU1
  - Criterios de aceptación HU2
- **Épica 2**
  - Criterios de aceptación HU3
  - Criterios de aceptación HU4
- **Épica 3**
  - Criterios de aceptación HU5
  - Criterios de aceptación HU6
- **Épica 4**
  - Criterios de aceptación HU7
  - Criterios de aceptación HU8

### **Reglas para definir criterios de aceptación:**

- Medibles y verificables.
- Especificar cuándo una historia se considera completada.

- Incluir condiciones específicas.
- No ambiguos.
- Probables de testear funcional o técnicamente.
- Mínimo 3 criterios por HU.

## 8. Fases de CRISP-DM

1. **Comprensión del negocio:** objetivo, valor agregado de IA, métricas de éxito.
2. **Comprensión de los datos:** tipo, origen, cantidad, calidad.
3. **Preparación de los datos:** características clave, transformaciones necesarias.
4. **Modelado:** tipo de problema, algoritmos posibles.
5. **Evaluación del modelo:** métricas de rendimiento.
6. **Despliegue del modelo (opcional):** tipo de despliegue y herramientas.

## 9. Desglose del trabajo en tareas

A partir de cada HU, descomponer en tareas concretas, técnicas y medibles:

- Duración estimada: entre 2 y 8 h. Evitar tareas genéricas.
- Si una tarea excede 8 h, dividirla.
- Indicar prioridad relativa (Alta, Media, Baja).

Historia de usuario	Tarea técnica	Estimación	Prioridad
HU1	Tarea 1 HU1	6 h	Alta
HU1	Tarea 2 HU1	8 h	Alta
HU2	Tarea 1 HU2	5 h	Media
HU2	Tarea 2 HU2	6 h	Alta
...	...	...	...

### Criterios para estimar tiempos:

- Considerar dificultad, complejidad técnica y grado de incertidumbre de cada tarea.
- Evitar subestimar el esfuerzo requerido. Si una tarea supera las 8 h, dividirla en subtareas.
- Basar la estimación en la experiencia propia o en referencias de tareas similares.

### Sobre la prioridad:

- Asignar una prioridad relativa (Alta, Media o Baja) según la relevancia funcional de la tarea y su impacto en los entregables.

- Priorizar tareas que estén vinculadas a criterios de aceptación de las HU o que sean necesarias para desbloquear otras.
- Incluir tareas opcionales solo si están bien justificadas.

### **Recomendaciones generales:**

- Incluir al menos dos tareas por historia de usuario.
- El total estimado debe ser coherente con la planificación global de unas 600 horas (la cantidad de horas sugeridas para un trabajo de posgrado).
- Este desglose se utilizará en las secciones siguientes para armar el diagrama de Gantt (sección 10) y la planificación de sprints (sección 11).
- Recordar que la calidad del desglose impacta directamente en la calidad de la planificación del proyecto.

## **10. Diagrama de Gantt**

El diagrama de Gantt debe representar de forma visual y cronológica todas las tareas del proyecto, abarcando aproximadamente 600 horas totales, de las cuales entre 480 y 500 deben destinarse a tareas técnicas (desarrollo, pruebas, implementación) y entre 100 y 120 a tareas no técnicas (planificación, documentación, escritura de memoria y preparación de la defensa).

### **Consignas y recomendaciones:**

- Incluir tanto tareas técnicas derivadas de las HU como tareas no técnicas generales del proyecto.
- El eje vertical debe listar las tareas y el eje horizontal representar el tiempo en semanas o fechas.
- Utilizar colores diferenciados para distinguir tareas técnicas y no técnicas.
- Las tareas deben estar ordenadas cronológicamente y reflejar todo el ciclo del proyecto.
- Iniciar con la planificación del proyecto (coincidente con el inicio de Gestión de Proyectos) y finalizar con la defensa, próxima a la fecha de cierre del trabajo.
- Configurar el software para mostrar los códigos del desglose de tareas y los nombres junto a cada barra.
- Asegurarse de que la fecha final coincida con la del Acta Constitutiva.
- Evitar tareas genéricas o ambiguas y asegurar una secuencia lógica y realista.
- Las fechas pueden ser aproximadas; ajustar el ancho del diagrama según el texto y el parámetro `x unit`. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

### **Herramientas sugeridas:**

- Planner, GanttProject, Trello + plugins  
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately (colaborativa online)  
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- LaTeX con pgfgantt:  
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Incluir una imagen legible del diagrama de Gantt. Si es muy ancho, presentar primero la tabla y luego el gráfico de barras.

## 11. Planificación de Sprints

Organizar las tareas técnicas del proyecto en sprints de trabajo que permitan distribuir de forma equilibrada la carga horaria total, estimada en 600 horas.

### Consigna:

- Completar una tabla que relacione sprints con HU y tareas técnicas correspondientes.
- Incluir estimación en horas para cada tarea.
- Indicar responsable y porcentaje de avance estimado o completado.
- Contemplar también tareas de planificación, documentación, redacción de memoria y preparación de defensa.

### Conceptos clave:

- Una épica es una unidad funcional amplia; una historia de usuario es una funcionalidad concreta; un sprint es una unidad de tiempo donde se ejecutan tareas.
- Las tareas son el nivel más desagregado: permiten estimar tiempos, asignar responsables y monitorear progreso.

### Duración sugerida:

- Para un proyecto de 600 h, se recomienda planificar entre 10 y 12 sprints de aproximadamente 2 semanas cada uno.
- Asignar entre 45 y 50 horas efectivas por sprint a tareas técnicas.
- Reservar 100 a 120 h para actividades no técnicas (planificación, escritura, reuniones, defensa).

### Importante:

- En proyectos individuales, el responsable suele ser el propio autor.

- Aun así, desagregar tareas facilita el seguimiento y mejora continua.

### Conversión opcional de Story Points a horas:

- 1 SP  $\approx$  2 h como referencia flexible.
- Tener en cuenta aproximaciones tipo Fibonacci.

Cuadro 1. Formato sugerido

Sprint	HU o fase	Tarea	Horas / SP	Responsable	% Completado
Sprint 0	Planificación	Definir alcance y cronograma	10 h	Alumno	100 %
Sprint 0	Planificación	Reunión con tutor/cliente	5 h	Alumno	50 %
Sprint 0	Planificación	Ajuste de entregables	6 h	Alumno	25 %
Sprint 1	HU1	Tarea 1 HU1	6 h / 3 SP	Alumno	0 %
Sprint 1	HU1	Tarea 2 HU1	10 h / 5 SP	Alumno	0 %
Sprint 2	HU2	Tarea 1 HU2	7 h / 5 SP	Alumno	0 %
...	...	...	...	...	...
Sprint 5	Escritura	Redacción memoria	50 h / 34 SP	Alumno	0 %
Sprint 6	Defensa	Preparación exposición	20 h / 13 SP	Alumno	0 %

### Recomendaciones:

- Verificar que la carga horaria por sprint sea equilibrada.
- Usar sprints de 1 a 3 semanas, acordes al cronograma general.
- Actualizar el % completado durante el seguimiento del proyecto.
- Considerar un sprint final exclusivo para pruebas, revisión y ajustes antes de la defensa.

## 12. Normativa y cumplimiento de datos (gobernanza)

En esta sección se debe analizar si los datos utilizados en el proyecto están sujetos a normativas de protección de datos y privacidad, y en qué condiciones se pueden emplear.

### Aspectos a considerar:

- Evaluar si los datos están regulados por normativas como GDPR, Ley 25.326 de Protección de Datos Personales en Argentina, HIPAA u otras según jurisdicción y temática.
- Determinar si el uso de los datos requiere consentimiento explícito de los usuarios involucrados.
- Indicar si existen restricciones legales, técnicas o contractuales sobre el uso, compartición o publicación de los datos.

- Aclarar si los datos provienen de fuentes licenciadas, de acceso público o bajo algún tipo de autorización especial.
- Analizar la viabilidad del proyecto desde el punto de vista legal y ético, considerando la gobernanza de los datos.

Este análisis es clave para garantizar el cumplimiento normativo y evitar conflictos legales durante el desarrollo y publicación del proyecto.

### 13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.  
Justificación...
- Ocurrencia (O): Y.  
Justificación...

Riesgo 3:

- Severidad (S): X.  
Justificación...
- Ocurrencia (O): Y.  
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como  $RPN=S \times O$ )

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado:



Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (\*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).  
Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S\*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O\*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

## 14. Sprint Review

La revisión de sprint (*Sprint Review*) es una práctica fundamental en metodologías ágiles. Consiste en revisar y evaluar lo que se ha completado al finalizar un sprint. En esta instancia, se presentan los avances y se verifica si las funcionalidades cumplen con los criterios de aceptación establecidos. También se identifican entregables parciales y se consideran ajustes si es necesario.

Aunque el proyecto aún se encuentre en etapa de planificación, esta sección permite proyectar cómo se evaluarán las funcionalidades más importantes del backlog. Esta mirada anticipada favorece la planificación enfocada en valor y permite reflexionar sobre posibles obstáculos.

**Objetivo:** anticipar cómo se evaluará el avance del proyecto a medida que se desarrollen las funcionalidades, utilizando como base al menos cuatro historias de usuario del *Product Backlog*.

Seleccionar al menos 4 HU del Product Backlog. Para cada una, completar la siguiente tabla de revisión proyectada:

**Formato sugerido:**

HU seleccionada	Tareas asociadas	Entregable esperado	¿Cómo sabrás que está cumplida?	Observaciones o riesgos
HU1	Tarea 1	Módulo funcional	Cumple criterios de aceptación definidos	Falta validar con el tutor
	Tarea 2			
HU3	Tarea 1	Reporte generado	Exportación disponible y clara	Requiere datos reales
	Tarea 2			
HU5	Tarea 1	Panel de gestión	Roles diferenciados operativos	Riesgo en integración
	Tarea 2			
HU7	Tarea 1	Informe trimestral	PDF con gráficos y evolución	Puede faltar tiempo para ajustes
	Tarea 2			

## 15. Sprint Retrospective

La retrospectiva de sprint es una práctica orientada a la mejora continua. Al finalizar un sprint, el equipo (o el alumno, si trabaja de forma individual) reflexiona sobre lo que funcionó bien, lo que puede mejorarse y qué acciones concretas pueden implementarse para trabajar mejor en el futuro.

Durante la cursada se propuso el uso de la **Estrella de la Retrospectiva**, que organiza la reflexión en torno a cinco ejes:

- ¿Qué hacer más?
- ¿Qué hacer menos?
- ¿Qué mantener?
- ¿Qué empezar a hacer?
- ¿Qué dejar de hacer?

Aun en una etapa temprana, esta herramienta permite que el alumno planifique su forma de trabajar, identifique anticipadamente posibles dificultades y diseñe estrategias de organización personal.

**Objetivo:** reflexionar sobre las condiciones iniciales del proyecto, identificando fortalezas, posibles dificultades y estrategias de mejora, incluso antes del inicio del desarrollo.

Completar la siguiente tabla tomando como referencia los cinco ejes de la Estrella de la Retrospectiva (*Starfish* o estrella de mar). Esta instancia te ayudará a definir buenas prácticas desde el inicio y prepararte para enfrentar el trabajo de forma organizada y flexible. Se deberá completar la tabla al menos para 3 sprints técnicos y 1 no técnico.

**Formato sugerido:**

<b>Sprint tipo y N°</b>	<b>¿Qué hacer más?</b>	<b>¿Qué hacer menos?</b>	<b>¿Qué mantener?</b>	<b>¿Qué empezar a hacer?</b>	<b>¿Qué dejar de hacer?</b>
Sprint técnico - 1	Validaciones continuas con el alumno	Cambios sin versión registrada	Pruebas con datos simulados	Documentar cambios propuestos	Ajustes sin análisis de impacto
Sprint técnico - 2	Verificar configuraciones en múltiples escenarios	Modificar parámetros sin guardar historial	Perfiles reutilizables	Usar logs para configuración	Repetir pruebas manuales innecesarias
Sprint técnico - 8	Comparar correlaciones con casos previos	Cambiar parámetros sin justificar	Revisión cruzada de métricas	Anotar configuraciones usadas	Trabajar sin respaldo de datos
Sprint no técnico - 12 (por ej.: “Defensa”)	Ensayos orales con feedback	Cambiar contenidos en la memoria	Material visual claro	Dividir la presentación por bloques	Agregar gráficos difíciles de explicar