



Asignación de Características Musicales con IA para Ambientación Comercial

Autor:

Ing. Carlos Alberto Rivas Araque

Director:

Ing. Esp. Martín Moreyra (GatekeeperX)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 26 de agosto de 2025 y el 14 de octubre de 2025.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar.	5
2. Identificación y análisis de los interesados	6
3. Propósito del proyecto	7
4. Alcance del proyecto	7
5. Supuestos del proyecto.	8
6. Product Backlog	9
7. Criterios de aceptación de historias de usuario	11
8. Fases de CRISP-DM	13
9. Desglose del trabajo en tareas	13
10. Planificación de Sprints	14
11. Diagrama de Gantt (sprints)	16
12. Normativa y cumplimiento de datos (gobernanza)	16
13. Gestión de riesgos	17
14. Sprint Review	18
15. Sprint Retrospective	19

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	26 de agosto de 2025
1	Se completa hasta el punto 5 inclusive	9 de septiembre de 2025

Acta de constitución del proyecto

Buenos Aires, 26 de agosto de 2025

Por medio de la presente se acuerda con el Ing. Carlos Alberto Rivas Araque que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Asignación de Características Musicales con IA para Ambientación Comercial” y consistirá en la implementación de un sistema de generación de características musicales a partir de canciones en el contexto de una aplicación de ambientación musical. El trabajo tendrá un presupuesto preliminar estimado de 600 horas y un costo estimado de 18750 dólares estadounidenses, con fecha de inicio el 26 de agosto de 2025 y fecha de presentación pública en junio de 2026.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Carolina Arbelaez
Plusyc Live SAS

Ing. Esp. Martín Moreyra
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

El proyecto se hace para la empresa Plusyc Live SAS, dedicada a la creación de experiencias musicales personalizadas para locales comerciales a través de una aplicación móvil. La aplicación utiliza un *pool* de canciones propio, y un *backend* que crea listas de reproducción. Las listas se generan gracias a características musicales, tales como el género, ánimo, energía y tempo de cada canción. Actualmente, la asignación de características musicales depende del acceso a interfaces de programación de aplicaciones (API) de terceros, lo que genera restricciones de disponibilidad y costos. Esto afecta la escalabilidad y autonomía del sistema, y en consecuencia, la calidad de la experiencia ofrecida a los clientes.

Se propone el desarrollo de un sistema de inteligencia artificial (IA) que reciba como entrada canciones y genere como salida sus características musicales. Se entrenarán y evaluarán modelos de aprendizaje supervisado, a partir de técnicas y representaciones numéricas del sonido, fuentes de datos privadas de la empresa, y datos públicos disponibles. Y se podrán inferir las características de nuevas canciones sin depender de proveedores externos.

En el ámbito de la inteligencia artificial aplicada a la música, existen herramientas que permiten extraer información musical a bajo nivel, y herramientas en línea que permiten extraer algunas características específicas. Sin embargo, no se cuenta con una solución que provea resultados completos y resuelva el problema de la disponibilidad.

La propuesta de valor radica en otorgar a Plusyc Live SAS una solución completa, integrada y altamente disponible, en pro de la escalabilidad y la independencia tecnológica. Los comercios tendrán una renovación en la experiencia de sus consumidores. El sistema será además reutilizable en otros contextos que podremos implementar en la empresa.

A continuación, en la figura 1, se presenta el diagrama de bloques de alto nivel de la aplicación. A la derecha del diagrama se observa el sistema basado en IA, que es el foco del proyecto.

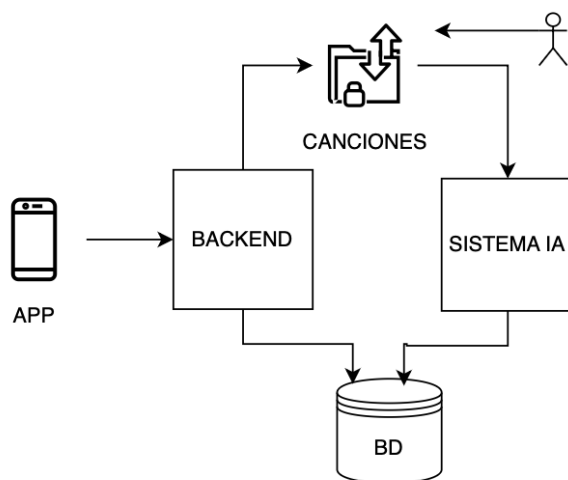


Figura 1. Diagrama en bloques del sistema.

En la figura se puede observar un actor que provee los archivos de las canciones que son las entradas del sistema IA, donde se procesa cada canción para extraer sus características musicales. Las características son las salidas, y son almacenadas en una base de datos que alimenta el *backend* de la aplicación. Ahora, en la figura 2, se muestra un diagrama de bloques más detallado del sistema IA.

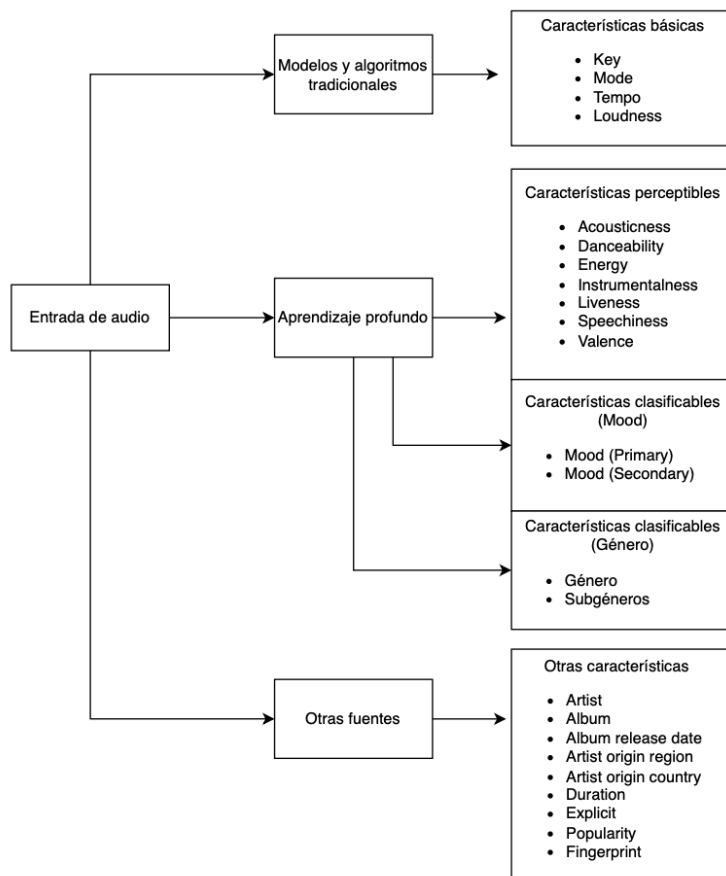


Figura 2. Diagrama de bloques del sistema IA.

En la figura se observa que el sistema IA utiliza tres módulos principales: “Modelos y algoritmos tradicionales”, “Aprendizaje profundo” y “Otras fuentes”. Cada módulo procesa el archivo de audio y se encarga de inferir o consultar un subconjunto específico de características musicales.

2. Identificación y análisis de los interesados

- Cliente: Carolina Arbelaez, es la encargada de aprobar los entregables.
- Impulsor: El Ing. Andrés Saldarriaga, es el responsable de brindar acceso a los datos del cliente, y las herramientas tecnológicas necesarias en producción, como credenciales, y proveedores de infraestructura en la nube.
- Responsable: Ing. Carlos Alberto Rivas Araque, es quien llevará a cabo el desarrollo del proyecto.

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Carolina Arbelaez	Plusyc Live SAS	C.E.O.
Impulsor	Ing. Andrés Saldarriaga	Plusyc Live SAS	C.T.O.
Responsable	Ing. Carlos Alberto Rivas Araque	FIUBA	Alumno
Orientador	Ing. Esp. Martín Moreyra	GatekeeperX	Director del Trabajo Final
Usuario final	-	-	Locales Comerciales

- Orientador: Ing. Esp. Martín Moreyra, es experto en solución de problemas de inteligencia artificial y va a ser consultor y ayuda para resolver problemas puntuales en caso de *blockers* técnicos.
- Usuario final: El usuario final son los clientes de Plusyc Live SAS, que utilizan la aplicación para ambientar su comercio con listas personalizadas.

3. Propósito del proyecto

Desarrollar un sistema de inteligencia artificial capaz de caracterizar automáticamente canciones a partir de su archivo de audio, con el fin de reducir la dependencia de servicios externos y garantizar la escalabilidad, autonomía y confiabilidad de la aplicación de Plusyc Live SAS. Con ello, se busca fortalecer la experiencia musical personalizada que la empresa ofrece a los comercios y abrir la posibilidad de aplicar esta tecnología en otros contextos dentro del sector musical.

4. Alcance del proyecto

Se definen los siguientes subconjuntos de características musicales:

- Básicas: {*Key, Mode, Tempo, Loudness*}.
- Perceptibles: {*Danceability, Energy, Speechiness, Acousticness, Instrumentalness, Liveness, Valence*}.
- Clasificables por ánimo (*Mood*): {*Primary Mood, Secondary Mood*}.
- Clasificables por género: {*Genre, Subgenres*}.
- Otras (Aquellas que no requieren de IA para su asignación): {*Artist, Album, Album release date, Artist origin region, Artist origin country, Duration, Explicit, Popularity, Fingerprint*}.

El proyecto incluye la asignación de las características básicas, perceptibles y las clasificables por ánimo y género. Para lograr esto es necesario:

- Revisión del estado del arte sobre *datasets* musicales disponibles para modelos de aprendizaje supervisado (algoritmos tradicionales) y aprendizaje profundo.

- Evaluación de los datos del cliente para verificar volumen, calidad de asignaciones y posibilidad de conformar un *dataset* propio.
- Análisis de audio, representaciones numéricas de señales de audio para depurar los *datasets*.
 - Extracción de representaciones tradicionales para los modelos supervisados.
 - Generación de representaciones intermedias para aprendizaje profundo.
- Depuración de los *datasets* para los modelos de aprendizaje supervisado tradicionales.
- Elección de modelos de aprendizaje supervisado y algoritmos tradicionales para la predicción de las características básicas.
- Evaluación de los *datasets* para modelos de aprendizaje profundo.
- Selección de modelos de aprendizaje profundo y modelos preentrenados para la inferencia de características perceptibles, y clasificables por ánimo y género.
- Implementación de los modelos seleccionados.
- Construcción de un pipeline para la asignación de características mediante el uso de los modelos seleccionados como los mejores.
- Almacenamiento de las características de salida en la base de datos.

El presente proyecto no incluye:

- Asignación del subconjunto de otras características que no requieren de IA para su asignación.
- Desarrollo de una interfaz para la recepción de un nuevo archivo de audio.
- Desarrollo de interfaces gráficas de usuario.
- Puesta en producción del sistema en un entorno del cliente.
- Mantenimiento y monitoreo a largo plazo.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone:

- Disponibilidad de recursos de *hardware*: Se dispondría de acceso a instancias de procesamiento locales, y en la nube con los suficientes recursos para las fases de entrenamiento y validación de los modelos.
- Disponibilidad de datos para el entrenamiento del modelo de procesamiento de audio: Se tendría acceso a un conjunto diverso de datos de audio que cubrirán una amplia gama de géneros y estilos.
- Factibilidad técnica: Se asume que las tecnologías actuales de procesamiento de audio son lo suficientemente avanzadas para implementar el proyecto. Y en particular.

- Las características básicas se pueden generar usando modelos de aprendizaje de máquina y algoritmos tradicionales.
- Las perceptibles y las clasificables por ánimo y género se pueden predecir usando un modelos de aprendizaje profundo de múltiples salidas.
- Tiempo: Se estima que las 600 horas asignadas para el desarrollo del proyecto serán suficientes para completar todas las etapas, incluyendo la planeación, análisis y diseño, desarrollo, pruebas y potenciales correcciones del sistema.
- Asistencia de los interesados: Se contaría con la ayuda del cliente ante *blockers* técnicos, y dudas sobre los requisitos del sistema.

6. Product Backlog

Se definen los siguientes roles:

- Usuario: el sistema esta diseñado para ser utilizado en ambientación musical en comercios. Los encargados de los comercios no necesitan conocer el funcionamiento interno del sistema, solo brindan sus preferencias y determinan si las canciones son adecuadas o no para sus comercios, lo cual no solo depende de las características musicales sino también de la configuración del comercio en la aplicación, por eso, en este caso, el usuario final es una persona del cliente con conocimientos cuantitativos que verifica que las características son asignadas correctamente de acuerdo a sus conocimientos musicales.
- Desarrollador de software con conocimiento en inteligencia artificial: la persona ocupando este rol deberá tener conocimiento suficiente para entrenar modelos de procesamiento de audio y generación de características musicales utilizando inteligencia artificial.
- Cliente: es el encargado de verificar que el proyecto cumple con las especificaciones y dar la aprobación final del sistema.

Épicas, historias de usuario (y spikes).

- Exploración y validación de *datasets* para modelos tradicionales
 - **SPIKE1 (3 SP, prioridad máxima):** Como desarrollador de IA, quiero revisar *datasets* musicales públicos y técnicas de representación de audio para aprendizaje supervisado, para identificar qué datos podrían usarse en el entrenamiento de modelos tradicionales.
 - **SPIKE2 (5 SP, prioridad máxima):** Como desarrollador de IA, quiero evaluar los datos del cliente para verificar volumen y calidad, para determinar si se puede conformar un *dataset* propio.
- Desarrollo y evaluación de modelos tradicionales
 - **HU1 (8 SP, prioridad alta):** Como desarrollador de IA, quiero entrenar y comparar modelos supervisados tradicionales (SVM, RF, k-NN), para seleccionar el mejor en la predicción de la característica A.

- **HU2 (8 SP, prioridad alta):** Como desarrollador de IA, quiero entrenar y comparar modelos supervisados tradicionales, para seleccionar el mejor en la predicción de la característica B.
 - **HU3 (8 SP, prioridad alta):** Como desarrollador de IA, quiero entrenar y comparar modelos supervisados tradicionales, para seleccionar el mejor en la predicción de la característica C.
 - .
 - .
 - .
 - **HUM Opcional (8 SP, prioridad baja):** Como desarrollador de IA, quiero entrenar y comparar modelos supervisados tradicionales, para seleccionar el mejor en la predicción de la característica M.
- **Desarrollo y evaluación de modelos de aprendizaje profundo**
- **SPIKE3 (3 SP, prioridad máxima):** Como desarrollador de IA, quiero revisar datasets musicales públicos y las técnicas de representación de audio para aprendizaje profundo, para identificar qué datos podrían usarse en el entrenamiento de modelos de deep learning.
 - **HUM+1 (8 SP, prioridad alta):** Como desarrollador de IA, quiero entrenar y evaluar modelos de deep learning (CNN, RNN, Transformers de audio, modelos preentrenados), para seleccionar el más adecuado en la predicción de la característica M+1.
 - **HUM+2 (8 SP, prioridad alta):** Como desarrollador de IA, quiero entrenar y evaluar modelos de deep learning, para seleccionar el más adecuado en la predicción de la característica M+2.
 - .
 - .
 - .
 - **HUN Opcional (8 SP, prioridad baja):** Como desarrollador de IA, quiero entrenar y evaluar modelos de deep learning, para seleccionar el más adecuado en la predicción de la característica N.
- **Implementación del pipeline de procesamiento y despliegue**
- **HUN+1 Opcional (5 SP, prioridad baja):** Como desarrollador, quiero una interfaz para la recepción de un nuevo archivo de audio, para alimentar el pipeline de inteligencia artificial que genere las características musicales.
 - **HUN+2 (13 SP, prioridad alta):** Como desarrollador, quiero un pipeline que procese los modelos desarrollados y asigne características al audio, para automatizar la generación de características musicales del audio y almacene los resultados en una base de datos.
 - **HUN+3 (13 SP, prioridad baja):** Como cliente, quiero desplegar el sistema en el *backend*, para automatizar la generación de características musicales del audio en producción.
 - **HUN+4 (1 SP, prioridad baja):** Como usuario, quiero tener acceso de lectura a la base de datos para poder validar la calidad de las características generadas.

7. Criterios de aceptación de historias de usuario

■ Exploración y validación de *datasets* para modelos tradicionales

● SPIKE1

- Se escribe un documento que incluye el análisis realizado de los *datasets* públicos, y las representaciones de audio disponibles.
- Se listan los *datasets* seleccionados para el conjunto de características A, B, C, \dots y las razones de selección.
- Se detalla el procesamiento necesario a realizar en los datos por cada uno de los *dataset* seleccionados.

● SPIKE2

- Se edita el documento anterior con el objetivo de incluir el inventario de los datos del cliente, incluyendo la cantidad de audios y características.
- Se describe la calidad de los datos en términos de la consistencia de las características asignadas.
- Se describen las razones que determinan la viabilidad de construir un *dataset* propio.

■ Desarrollo y evaluación de modelos tradicionales

● HU1 (Ejecución del ciclo de vida de IA para predicciones de la característica A)

- Se obtiene un *dataset* depurado para la característica a partir de los documentos generados en los spikes, para la característica A.
- Se constata la aplicación de técnicas de feature engineering.
- Se constata el entrenamiento de más de un modelo, según aplique.
- Se entrega el mejor modelo resultante de la evaluación de las métricas aplicadas al caso específico (precisión, recall, F1 score, test speedup).
- Se itera hasta obtener resultados aceptables. (70 % de F1, balance de precisión y recall, y el mejor speedup).

● HU2 (Ejecución del ciclo de vida de IA para la característica B)

- Se obtiene un *dataset* o se reutiliza un *dataset* ya depurado a partir de los documentos generados en los spikes para la característica B.
- Se constata la aplicación de técnicas de feature engineering.
- Se constata el entrenamiento de más de un modelo, según aplique.
- Se entrega el mejor modelo resultante de la evaluación de las métricas aplicadas al caso específico (precisión, recall, F1 score, test speedup).
- Se itera hasta obtener resultados aceptables. (70 % de F1, balance de precisión y recall, y el mejor speedup).

● HU3 (característica C)

- Se obtiene un *dataset* o se reutiliza un *dataset* ya depurado a partir de los documentos generados en los spikes para la característica C.
- Se constata la aplicación de técnicas de feature engineering.
- Se constata el entrenamiento de más de un modelo, según aplique.
- Se entrega el mejor modelo resultante de la evaluación de las métricas aplicadas al caso específico (precisión, recall, F1 score, test speedup).

- Se itera hasta obtener resultados aceptables. (70 % de F1, balance de precisión y recall, y el mejor speedup).
- .
- .
- .
- **Desarrollo y evaluación de modelos de aprendizaje profundo**
 - **SPIKE3**
 - Se edita el documento resultante del SPIKE1 incluyendo el análisis realizado de los *datasets* públicos, y las representaciones de audio aplicables al aprendizaje profundo.
 - Se listan los *datasets* seleccionados para el conjunto de características $M + 1, M + 2, \dots$ y las razones de selección.
 - Se detalla el procesamiento necesario a realizar en los datos por cada uno de los *dataset* seleccionados.
 - **HUM+1 (característica M+1)**
 - Se obtiene un *dataset* depurado para la característica a partir de los documentos generados en el SPIKE3 para la característica M+1.
 - Se constata la aplicación de técnicas relacionadas con el **ciclo de vida del aprendizaje profundo**.
 - Se constata el entrenamiento de más de un modelo de tipo CNN, RNN, Transformer o preentrenado, según aplique.
 - Se entrega el mejor modelo resultante de la evaluación de las métricas aplicadas al caso específico. **Revisar los conceptos de DL**.
 - Se itera hasta obtener resultados aceptables.
 - **HUM+2 (característica M+2)**
 - Se obtiene un *dataset* depurado para la característica a partir de los documentos generados en el SPIKE3 para la característica M+2.
 - Se constata la aplicación de técnicas relacionadas con el **ciclo de vida del aprendizaje profundo**.
 - Se constata el entrenamiento de más de un modelo, según aplique.
 - Se entrega el mejor modelo resultante de la evaluación de las métricas aplicadas al caso específico. **Revisar los conceptos de DL**.
 - Se itera hasta obtener resultados aceptables.
 - .
 - .
 - .
- **Implementación del pipeline de procesamiento y despliegue**
 - **HUN+1 Opcional (interfaz de recepción)**
 - La interfaz permite subir archivos de audio en formato estándar (**.mp3 o .wav**).
 - El archivo queda en una cola de procesamiento para el pipeline.
 - El sistema devuelve confirmación de recepción y ejecuta el pipeline por archivo.
 - **HUN+2 (pipeline de procesamiento)**
 - El pipeline procesa automáticamente un archivo nuevo.

- Se ejecutan los modelos entrenados y se asignan las características al audio.
- Los resultados se almacenan en la base de datos.
- **HUN+3 (despliegue en *backend*)**
 - El sistema está disponible en un entorno de prueba accesible por el cliente.
 - El backend permite ejecutar al menos (x archivos consecutivos sin fallo.
 - La documentación de despliegue está disponible para reproducir la instalación.
- **HUN+4 (validación por usuario)**
 - El usuario puede acceder para consultar las características de todas las canciones procesadas en la base de datos.
 - El usuario puede filtrar las características canción por canción.
 - El usuario puede verificar los resultados y reportar cualquier incidente al cliente y al desarrollador.

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 Historia de Usuario (HU) definida en la sección 6, descomponer el trabajo en tareas técnicas concretas, medibles y acotadas en el tiempo.

- Seleccionar entre 2 y 3 tareas por cada historia de usuario.
- Cada tarea debe estar claramente formulada, ser técnica, accionable y con una estimación horaria entre 2 y 8 horas.
- Evitar tareas genéricas (como "desarrollar funcionalidad") o demasiado amplias.
- Si una tarea supera las 8 horas, debe dividirse en subtareas.
- Indicar la prioridad relativa de cada tarea (Alta, Media o Baja).

Criterios para estimar tiempos:

- Considerar la complejidad técnica, el nivel de incertidumbre y la experiencia previa.
- Evitar subestimar el esfuerzo: estimar el tiempo realista que llevaría implementar, testear y documentar cada tarea.

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
...

- 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:

- -Enfocarse en tareas que surgen directamente de las HU planteadas.
- No es necesario cubrir las 600 horas del proyecto en esta sección: el foco está en el desglose de funcionalidades clave.
- Este trabajo será la base para organizar algunos de los sprints y elaborar el cronograma del proyecto, por lo que debe ser claro y realista.

10. 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 el tutor/cliente	5 h	Alumno	50 %
Sprint 0	Planificación	Ajuste de los 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 de la 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.

11. Diagrama de Gantt (sprints)

Visualizar en un diagrama de Gantt la planificación temporal del proyecto, tomando como base los sprints definidos en la sección anterior. Debe contemplar todas las horas del proyecto.

Consigna:

- Elaborar un diagrama de Gantt que muestre la secuencia temporal de los sprints.
- Cada fila debe representar un sprint (con su número o nombre), y el eje horizontal debe indicar el tiempo (en semanas o fechas concretas).
- Las tareas técnicas derivadas de HU deben diferenciarse visualmente (por ejemplo, con un color distinto) de las tareas no técnicas (planificación, redacción, defensa).
- Incluir todas las tareas estimadas en cada sprint.

Recomendaciones para el Gantt:

- Podés usar herramientas gratuitas como TeamGantt, ClickUp, GanttProject, [Google Sheets], [Trello + Planyway], entre otras.
- Ordená los sprints de forma cronológica, comenzando con Sprint 0 (planificación) y finalizando con el sprint de defensa.
- Asegurate de reflejar la duración realista de cada sprint según tu disponibilidad y el cronograma general del posgrado.
- Incluí hitos importantes: reuniones, entregas parciales, defensa.

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

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:

Sprint tipo y N°	¿Qué hacer más?	¿Qué hacer menos?	¿Qué mantener?	¿Qué empezar a hacer?	¿Qué dejar de hacer?
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