



Detección de anomalías en servicio de TV-over-IP mediante autoencoder LSTM

Autor:

Ing. Christopher Charaf

Director:

Título y Nombre del director (buscando)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 24 de junio de 2025 y el 20 de Octubre de 2025.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar	5
2. Identificación y análisis de los interesados	7
3. Propósito del proyecto	7
4. Alcance del proyecto	8
5. Supuestos del proyecto.	8
6. Requerimientos	9
7. Historias de usuarios (<i>Product backlog</i>).	10
8. Entregables principales del proyecto	11
9. Desglose del trabajo en tareas	12
10. Diagrama de Activity On Node.	13
11. Diagrama de Gantt	13
12. Presupuesto detallado del proyecto	15
13. Gestión de riesgos	15
14. Gestión de la calidad	16
15. Procesos de cierre	17

-

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	24 de junio de 2025

Acta de constitución del proyecto

Buenos Aires, 24 de junio de 2025

Por medio de la presente se acuerda con el Ing. Christopher Charaf que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Detección de anomalías en servicio de TV-over-IP mediante autoencoder LSTM” y consistirá en **la implementación de un sistema de detección de anomalías en un servicio de TV-over-IP mediante un autoencoder LSTM**. El trabajo tendrá un presupuesto preliminar estimado de 600 horas y un costo estimado de \$ 7000, con fecha de inicio el 24 de junio de 2025 y fecha de presentación pública el 15 de diciembre de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Kaltura Inc.
Kaltura Inc.

Título y Nombre del director
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

El presente proyecto tiene como objetivo desarrollar un sistema de detección de anomalías en tiempo real aplicado a un servicio de televisión por protocolo de Internet (TV-over-IP) basado en interfaces de programación de aplicaciones (API). Este servicio es monitoreado mediante métricas recolectadas a través de una herramienta de acumulación de métricas llamada Prometheus, con una frecuencia de muestreo de 30 segundos. El sistema propuesto busca identificar comportamientos inusuales o fallos en el servicio mediante la reconstrucción de secuencias de métricas utilizando un modelo Autoencoder basado en redes neuronales de memoria a largo y corto plazo (LSTM, por sus siglas en inglés: *Long Short-Term Memory*).

La empresa cliente de este proyecto es "Kaltura Inc.", la cual provee servicios de TV-over-IP sobre infraestructura en la nube de Amazon Web Services (AWS). Esta empresa se enfrenta al desafío creciente de modernizar sus herramientas de monitoreo para mejorar la capacidad de reacción ante fallos en producción, sin depender exclusivamente de soluciones externas. Por este motivo, este desarrollo responde a una necesidad concreta del área de operaciones y calidad de servicio de la organización frente a sus clientes.

Actualmente, los métodos tradicionales de monitoreo utilizan umbrales fijos o alertas manuales, lo cual resulta ineficiente frente a comportamientos no triviales o patrones de uso dinámicos. Además, muchas herramientas de monitoreo de terceros requieren la exposición de métricas internas a servicios externos, por lo que compromete la privacidad de los datos y aumenta la propensión a ataques. En contraste, este proyecto propone una solución propietaria, basada en aprendizaje automático, que permita detectar anomalías de manera autónoma, no invasiva y segura.

Desde el punto de vista técnico, el modelo recibirá como entrada ventanas deslizantes de tiempo compuestas por un número a definir de métricas técnicas que reflejan el rendimiento del servicio en tiempo real. Se utilizarán técnicas de ingeniería de características (*feature engineering*) como la codificación cíclica del horario y variables contextuales que representen ventanas de mantenimiento y fines de semana para mejorar la sensibilidad del modelo frente a patrones estacionales.

Se empleará una arquitectura LSTM Autoencoder que aprende a reconstruir secuencias temporales multivariadas. Si el error de reconstrucción excede un umbral previamente definido, se considera que ocurrió una anomalía. En caso de detección, se emitirán alertas automáticas a través de la plataforma de alertas Opsgenie y se generarán enlaces a paneles de visualización de métricas como Grafana con el contexto temporal de la anomalía.

Esta propuesta se encuentra actualmente en su etapa inicial de planeamiento, donde se están definiendo tanto la arquitectura del modelo como las herramientas de integración con el ecosistema de monitoreo ya existente. La innovación principal reside en la aplicación de redes neuronales recurrentes para detección de anomalías directamente dentro del entorno técnico de la empresa, sin depender de soluciones comerciales de terceros. Esto refuerza la privacidad de los datos sensibles del sistema y de los usuarios, y permite una mayor adaptabilidad a los cambios en el comportamiento del servicio.

En comparación con el estado del arte, esta solución se destaca por su enfoque específico a servicios basados en API en tiempo real, su integración directa con herramientas de acumulación de métricas Prometheus y Grafana y por aplicar técnicas avanzadas de aprendizaje profundo en lugar de reglas estáticas. Esto la convierte en una alternativa más precisa, segura, escalable

y personalizada que herramientas convencionales de detección de anomalías y alertas nativas de Grafana.

Cabe mencionar que no existen fuentes de financiamiento externo ni convenios públicos involucrados. El proyecto se desarrolla como parte del trabajo profesional del autor, y de acuerdo con el contrato vigente con la empresa, la propiedad intelectual de todos los entregables pertenece a la organización.

El cliente interno de este desarrollo es el área de operaciones de la empresa, que valora especialmente la capacidad de detección proactiva de incidentes, la integración fluida con su infraestructura existente y la reducción de falsos positivos. Necesita una solución confiable, sustentable y alineada con sus políticas de seguridad.

En la figura 1 se presenta el diagrama en bloques del sistema. Se observa que el flujo de datos comienza con la recolección de métricas desde Prometheus a través de su API, las cuales se almacenan en un marco de datos (*dataframe*) y luego se someten a un proceso de preprocesamiento e ingeniería de características. A continuación, estas métricas son introducidas en un modelo Autoencoder basado en redes LSTM, compuesto por un codificador (*encoder*) y un decodificador (*decoder*), que tiene como objetivo reconstruir la secuencia original de datos. El sistema evalúa el error de reconstrucción de dichas métricas y lo compara contra un umbral definido: si el error excede dicho umbral, se considera que ocurrió una anomalía. En ese caso, se dispara una alerta automática hacia Opsgenie y se genera un enlace contextual hacia un panel de visualización en Grafana, lo que permite al equipo de operaciones inspeccionar el evento detectado con información visual en tiempo real.

Si no se detecta anomalía, el sistema continúa procesando los siguientes datos de forma continua y en tiempo real.

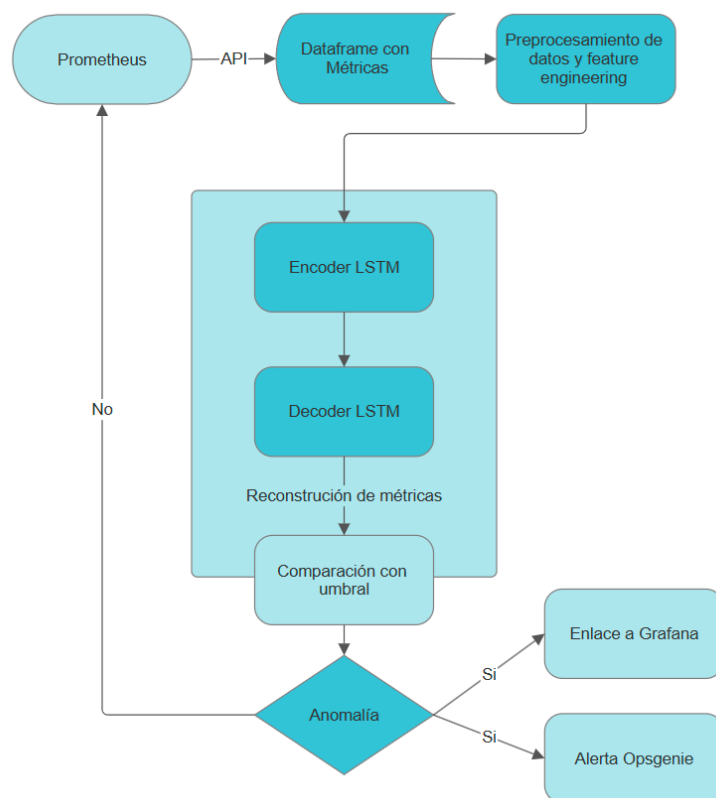


Figura 1. Diagrama en bloques del sistema.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Auspiciante	Kaltura Inc.	Kaltura Inc.	Departamento de soporte y operaciones
Responsable	Ing. Christopher Charaf	FIUBA	Alumno
Orientador	Título y Nombre del director	buscando	Director del Trabajo Final
Usuario final	Kaltura Inc.	Kaltura Inc.	Departamento de soporte y operaciones

- Orientador: a definir.
- Auspiciante: será el mismo usuario final de la solución, valora costos y organización en los procesos automatizados.

3. Propósito del proyecto

El propósito del proyecto es diseñar e implementar un sistema inteligente que permita detectar anomalías en tiempo real dentro de un servicio de televisión por protocolo de *Internet (TV-over-IP)*, basado en APIs y desplegado sobre infraestructura en la nube. Mediante el uso de

un modelo *autoencoder* basado en redes neuronales LSTM, se busca brindar a la empresa una herramienta robusta y segura para el monitoreo continuo de métricas técnicas clave; esta solución mejora su capacidad de respuesta ante incidentes, minimiza el tiempo de inactividad del servicio y garantiza una mayor calidad de experiencia para sus usuarios finales.

4. Alcance del proyecto

El proyecto **incluye**:

- La definición y análisis de métricas críticas del servicio de TV-over-IP recolectadas desde Prometheus mediante su API.
- El diseño e implementación de una serie de procesos(*pipeline*) de procesamiento de datos, incluyendo:
 - Normalización de variables.
 - *Feature engineering* (codificación cíclica de hora, día, etc.).
 - Construcción de ventanas deslizantes para series temporales.
- El desarrollo de un modelo de detección de anomalías basado en Autoencoder LSTM, entrenado para reconstruir secuencias multivariadas normales.
- La integración del sistema con herramientas de monitoreo existentes:
 - Generación de alertas automáticas vía Opsgenie.
 - Enlace contextual a paneles de Grafana.
- La validación del sistema en un entorno de producción con datos reales históricos.
- La elaboración de documentación técnica y funcional del prototipo.

El presente proyecto **no incluye**:

- El desarrollo de interfaces gráficas adicionales fuera de Grafana.
- La implementación de acciones correctivas automáticas posteriores a la detección de anomalías.
- La gestión directa de recursos en AWS ni tareas de infraestructura subyacente (como escalado automático, balanceo de carga, etc.).

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

1. Kaltura Inc. brindará acceso a las métricas necesarias a través de su instancia de Prometheus mediante API REST, así como también a sus paneles de visualización en Grafana.

2. El *dataframe* histórico utilizado para entrenamiento estará disponible, completo y será representativo del comportamiento normal del sistema en condiciones reales de operación.
3. Se contará con los recursos computacionales necesarios (por ejemplo, GPU opcional) para el entrenamiento del modelo LSTM Autoencoder en un entorno controlado por Kaltura Inc..
4. Las herramientas de integración como Opsgenie y Grafana ya están operativas y disponibles para pruebas dentro de la infraestructura de la empresa.
5. No se producirán cambios drásticos en el comportamiento del servicio durante el período de entrenamiento y validación que comprometan la utilidad del modelo.
6. El equipo de operaciones colaborará en la validación funcional del sistema, especialmente en la evaluación de falsos positivos y en el ajuste del umbral de alerta en base a estándares existentes en la empresa.
7. Las condiciones legales, de seguridad y de confidencialidad establecidas en el contrato del autor con la empresa se mantendrán vigentes y permitirán el desarrollo del proyecto sin restricciones externas.

6. Requerimientos

1. **Requerimientos funcionales:**
 - 1.1. El sistema debe ser capaz de recolectar métricas técnicas desde Prometheus a través de su API REST. (**Alta prioridad**)
 - 1.2. El sistema debe construir ventanas deslizantes de 20 pasos temporales (10 minutos) con una frecuencia de muestreo de 30 segundos. (**Alta prioridad**)
 - 1.3. El sistema debe aplicar técnicas de normalización y codificación cíclica de variables temporales como `hour_sin` y `hour_cos`. (**Alta prioridad**)
 - 1.4. El modelo debe reconstruir secuencias multivariadas. (**Alta prioridad**)
 - 1.5. El sistema debe detectar anomalías cuando el error de reconstrucción supere un umbral configurable. (**Alta prioridad**)
 - 1.6. En caso de anomalía, el sistema debe enviar una alerta automática a través de Opsgenie. (**Media prioridad**)
 - 1.7. El sistema debe incluir un enlace a Grafana que muestre el contexto temporal de la anomalía detectada. (**Media prioridad**)
2. **Requerimientos de documentación:**
 - 2.1. Se debe entregar documentación técnica del modelo, su entrenamiento, configuración y uso. (**Alta prioridad**)
 - 2.2. Se debe entregar una memoria del trabajo final con descripción funcional, arquitectura y resultados. (**Alta prioridad**)
 - 2.3. Se debe incluir una guía básica de despliegue e integración del sistema en entornos compatibles. (**Media prioridad**)
3. **Requerimientos de *testing* y validación:**
 - 3.1. El sistema debe ser validado sobre datos históricos que representen el comportamiento normal del sistema. (**Alta prioridad**)

3.2. Se deben generar métricas de desempeño como tasa de falsos positivos y precisión de detección. (**Alta prioridad**)

3.3. El umbral de error debe ser calibrado en conjunto con el equipo de operaciones. (**Media prioridad**)

4. Requerimientos de interfaz:

4.1. El sistema no debe contar con una interfaz gráfica propia, pero debe integrarse con visualizaciones (*dashboards*) existentes en Grafana. (**Alta prioridad**)

4.2. El mensaje de alerta debe incluir timestamp, métricas anómalas, y visualización de valores reales vs reconstruidos. (**Media prioridad**)

5. Requerimientos de interoperatividad e integración:

5.1. El sistema debe integrarse con la instancia de Prometheus de la empresa. (**Alta prioridad**)

5.2. El sistema debe ser compatible con Grafana para visualización. (**Alta prioridad**)

5.3. El sistema debe ser capaz de emitir alertas en el formato requerido por Opsgenie. (**Media prioridad**)

6. Requerimientos normativos y de seguridad:

6.1. El sistema no debe enviar datos a servicios externos no autorizados. (**Alta prioridad**)

6.2. Todo el procesamiento debe realizarse dentro de la red interna de la empresa o su infraestructura autorizada en AWS. (**Alta prioridad**)

6.3. El sistema debe respetar las políticas internas de privacidad y seguridad de la información establecidas por la empresa. (**Alta prioridad**)

7. Requerimientos opcionales:

7.1. Incluir variables adicionales como `weekday`, `is_weekend`, `is_night` en el *feature engineering*. (**Opcional**)

7.2. Permitir ajuste dinámico del umbral de detección desde una configuración externa. (**Opcional**)

7. Historias de usuarios (*Product backlog*)

Los **story points** se asignaron considerando tres factores:

- Complejidad técnica (1 a 5)
- Dificultad de implementación (1 a 5)
- Grado de incertidumbre (1 a 5)

Fórmula: **SP = Complejidad + Dificultad + Incertidumbre**

1. “Como ingeniero de monitoreo quiero recibir alertas automáticas cuando se detecten anomalías para poder reaccionar rápidamente y minimizar el impacto en el servicio.”

Story points: 8 (Complejidad: 3, Dificultad: 2, Incertidumbre: 3)

2. **“Como desarrollador de datos quiero procesar las métricas recolectadas y convertirlas en secuencias temporales con variables adicionales para entrenar el modelo.”**

Story points: 7 (Complejidad: 3, Dificultad: 2, Incertidumbre: 2)

3. **“Como operador del sistema quiero que cada alerta contenga un enlace directo a un panel de Grafana para visualizar rápidamente la anomalía detectada.”**

Story points: 5 (Complejidad: 2, Dificultad: 1, Incertidumbre: 2)

4. **“Como responsable de infraestructura quiero que el sistema funcione dentro de nuestra red privada para evitar exponer datos sensibles a servicios externos.”**

Story points: 6 (Complejidad: 2, Dificultad: 2, Incertidumbre: 2)

5. **“Como científico de datos quiero entrenar el modelo sobre datos históricos para que aprenda el comportamiento normal del sistema.”**

Story points: 6 (Complejidad: 2, Dificultad: 2, Incertidumbre: 2)

6. **“Como técnico de soporte quiero acceder a un log con las métricas reales y reconstruidas al momento de la anomalía para facilitar el diagnóstico del incidente.”**

Story points: 5 (Complejidad: 2, Dificultad: 2, Incertidumbre: 1)

8. Entregables principales del proyecto

- **Documentación técnica del sistema:**

Contendrá la descripción de la arquitectura del modelo, el flujo de procesamiento de datos, los componentes implementados y las herramientas utilizadas (Prometheus, Grafana, Opsgenie, entre otras).

- **Memoria del trabajo final:**

Documento completo que incluye la descripción técnica-conceptual, estado del arte, objetivos, metodología, implementación, resultados obtenidos y conclusiones.

- **Código fuente del sistema:**

Scripts y módulos desarrollados para el procesamiento de datos, entrenamiento del modelo, inferencia y generación de alertas. El código será entregado con instrucciones de ejecución y dependencias requeridas.

- **Modelo LSTM Autoencoder entrenado:**

Archivo con los pesos y la estructura del modelo entrenado, listo para su despliegue en entorno de validación o producción.

- **Pipeline de procesamiento de datos:**

Conjunto de scripts que automatizan las etapas de recolección, transformación, normalización y estructuración de datos históricos en secuencias para el modelo.

- **Configuración de integración con Opsgenie y Grafana:**

Archivos de configuración, ejemplos y plantillas para la conexión del sistema con los mecanismos de alerta y visualización en tiempo real utilizados por la empresa.

9. Desglose del trabajo en tareas

1. Análisis y definición del sistema (40 h)
 - 1.1. Revisión del estado del arte y casos similares (8 h)
 - 1.2. Análisis de métricas disponibles en Prometheus (12 h)
 - 1.3. Definición de requerimientos funcionales y no funcionales (10 h)
 - 1.4. Diseño preliminar del sistema y definición del pipeline (10 h)
2. Desarrollo del pipeline de datos (50 h)
 - 2.1. Implementación del recolector de métricas desde Prometheus (10 h)
 - 2.2. Desarrollo de preprocesamiento y normalización (12 h)
 - 2.3. Implementación de ingeniería de variables temporales (10 h)
 - 2.4. Construcción de ventanas deslizantes para series temporales (10 h)
 - 2.5. Validación del pipeline con datos históricos (8 h)
3. Desarrollo y entrenamiento del modelo (60 h)
 - 3.1. Definición de arquitectura LSTM Autoencoder (10 h)
 - 3.2. Implementación del modelo en Keras/TensorFlow (12 h)
 - 3.3. Preparación del conjunto de entrenamiento y validación (10 h)
 - 3.4. Entrenamiento del modelo con tuning de hiperparámetros (16 h)
 - 3.5. Evaluación del desempeño con métricas objetivas (12 h)
4. Integración con sistema de monitoreo (Grafana y Opsgenie) (35 h)
 - 4.1. Diseño del mecanismo de alerta con umbral configurable (10 h)
 - 4.2. Integración con Opsgenie para envío de alertas (8 h)
 - 4.3. Generación de enlaces hacia dashboards de Grafana (7 h)
 - 4.4. Pruebas de integración en entorno controlado (10 h)
5. Validación, documentación y entregables (40 h)
 - 5.1. Validación funcional con el equipo de operaciones (10 h)
 - 5.2. Elaboración del informe de resultados y validación (8 h)
 - 5.3. Redacción de la memoria técnica del proyecto (12 h)
 - 5.4. Preparación de documentación de uso e integración (10 h)

Cantidad total de horas: 225 h.

10. Diagrama de Activity On Node

Color	Grupo de tareas	Descripción
Amarillo claro	Grupo 1: Análisis y definición del sistema	Tareas relacionadas con el estudio preliminar, análisis de métricas, requisitos y diseño inicial.
Celeste claro	Grupo 2: Desarrollo del pipeline de datos	Implementación del recolector de métricas, preprocesamiento, normalización y ventanas deslizantes.
Violeta claro	Grupo 3: Desarrollo y entrenamiento del modelo	Definición de la arquitectura, implementación y evaluación del modelo LSTM Autoencoder.
Rosado claro	Grupo 4: Integración con sistema de monitoreo	Tareas de conexión con Opsgenie, configuración de alertas y enlaces hacia dashboards de Grafana.
Verde claro	Grupo 5: Validación, documentación y entregables	Actividades de verificación, redacción del informe técnico y documentación final del proyecto.
Blanco	Inicio / Fin	Representa los nodos de comienzo y cierre del proyecto.

Cuadro 1. Cuadro indicativo de colores del Diagrama AoN

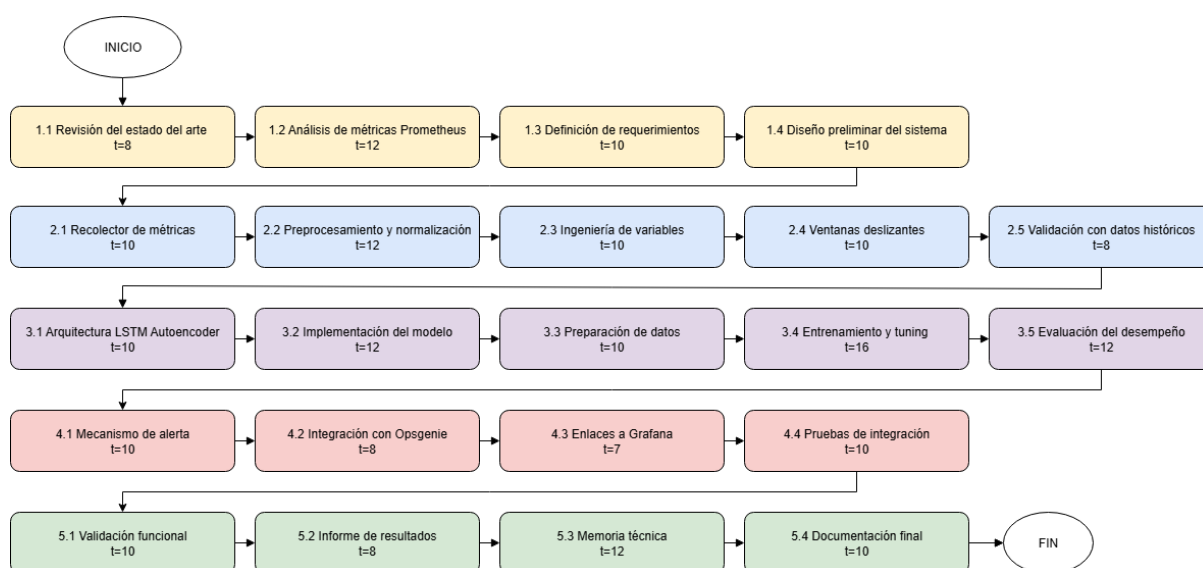


Figura 2. Diagrama de *Activity on Node*.

11. Diagrama de Gantt

A continuación se presenta el diagrama de Gantt correspondiente al cronograma del proyecto. El mismo se muestra en formato apaisado con el objetivo de facilitar su lectura y visualización, permitiendo apreciar con claridad la secuencia temporal de las actividades planificadas, su duración estimada y su relación con el resto de las tareas. Este diagrama fue elaborado en base al desglose de tareas definido en el WBS (punto 9) y refleja la distribución efectiva del trabajo a lo largo del tiempo, considerando una carga de 3 horas diarias, días no laborables y períodos de vacaciones.

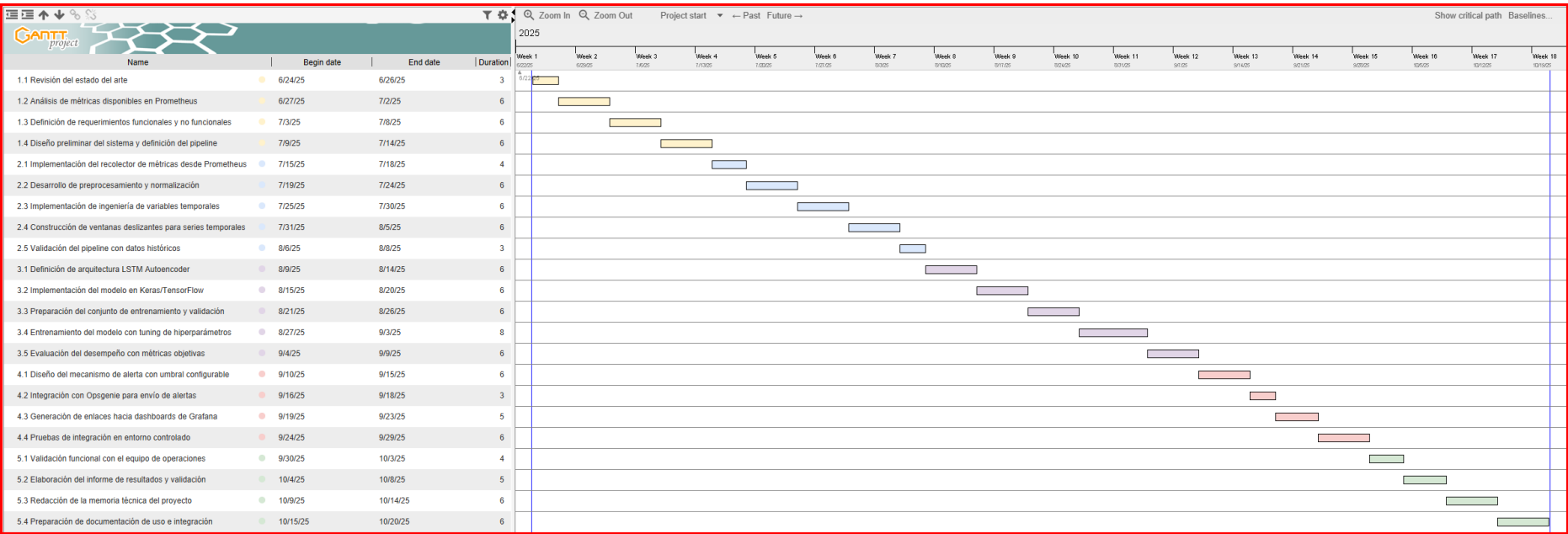


Figura 3. Diagrama de Gantt (apaisado).

12. Presupuesto detallado del proyecto

Descripción	Cantidad (h)	Valor unitario (ARS)	Valor total (ARS)
Análisis y definición del sistema	40	6.000	240.000
Desarrollo del pipeline de datos	50	6.000	300.000
Desarrollo y entrenamiento del modelo	60	6.000	360.000
Integración con herramientas de monitoreo	35	6.000	210.000
Validación, documentación y entregables	40	6.000	240.000
Subtotal costos directos	225		1.350.000

Cuadro 2. Presupuesto de costos directos del proyecto

Descripción	Cantidad	Valor unitario (ARS)	Valor total (ARS)
Electricidad y conectividad (mensual estimado)	3 meses	15.000	45.000
Licencias de software y servicios en la nube(AWS)	1 uso	25.000	25.000
Subtotal costos indirectos			70.000

Cuadro 3. Presupuesto de costos indirectos del proyecto

Costo total del proyecto: ₡ 1.420.000 ARS

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...

- Ocurriencia (O): Y.
Justificación...

Riesgo 3:

- Severidad (S): X.
Justificación...
- Ocurriencia (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. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento con su correspondiente número.

- Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar.
- Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar.

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc.

Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno.

En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, los problemas que surgieron y cómo se solucionaron:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
 - Indicar esto y quién financiará los gastos correspondientes.