Autoencoders para reforzar modelos de prevención de fraude

Autor:
Ariel Salassa

Director:
M. Sc. Lcdo. Franco Arito (Mercado Libre)

Esta planificación fue realizada en el curso de Gestión de proyectos entre el 24 de junio de 2021 y el 19 de agosto de 2021.

Índice

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0.0	Creación del documento	24/06/2021
1.0	Se completa hasta la sección 5 inclusive	07/07/2021
1.1	Correcciones de la versión 1.0	08/07/2021
2.0	Se completa hasta la sección 9 inclusive	14/07/2021
2.1	Correcciones de la versión 1.0	22/07/2021
3.0	Se completa hasta la sección 12 inclusive	29/07/2021
4.0	Se completa hasta la sección 15 inclusive	05/08/2021
4.1	Correcciones de la versión 4.0	10/08/2021

Acta de constitución del proyecto

Buenos Aires, 24 de junio de 2021

Por medio de la presente se acuerda con el Ing. Ariel Salassa que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará "Autoencoders para reforzar modelos de prevención de fraude" y consistirá esencialmente en un sistema que aporte información complementaria para el entrenamiento de futuros modelos de prevención de fraude de pagos electrónicos. El Trabajo Final tendrá un presupuesto preliminar estimado de 630 hs de trabajo y recursos económicos brindados por la empresa Mercado Libre, con fecha de inicio 24 de junio de 2021 y fecha de presentación pública 15 de junio de 2022.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg
Director posgrado FIUBA

M. Sc. Lcdo. Franco Arito
Mercado Libre

M. Sc. Lcdo. Franco Arito
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

Mercado Pago es la plataforma fintech dentro del ecosistema de Mercado Libre (Meli). En esta plataforma, que tiene millones de usuarios activos, se ofrecen distintas soluciones tecnológicas que hace posible pagar y cobrar en forma online. Algunos ejemplos de dichas soluciones son: abono de servicios, recargas de teléfono o pases de transporte, pagos y cobros con códigos QR, envíos de dinero, entre otros.

Dada la gran cantidad de usuarios y transacciones, y la variedad de estas, ha sido necesario desarrollar modelos de machine learning que sean capaces de detectar transacciones fraudulentas que perjudican reputacional y económicamente a la empresa, como se puede visualizar en la Figura 1.

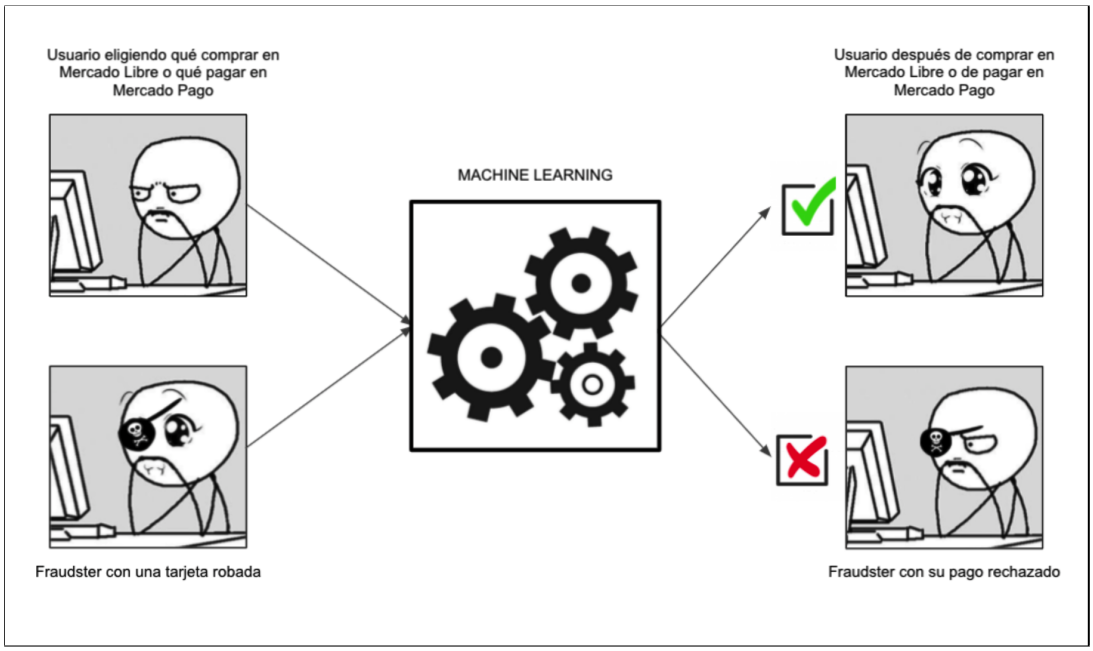


Figura 1. Esquema del comportamiento esperado de los usuarios dentro de las plataformas de Meli.

El entrenamiento de modelos de machine learning para este tipo de aplicaciones conlleva una dificultad adicional: los datos de entrenamiento están fuertemente desbalanceados, es decir, la cantidad de pagos no fraudulentos es mucho mayor que la cantidad de pagos fraudulentos.

Para hacerle frente a este problema, lo que se propone hacer es entrenar un autoencoder, cuya arquitectura esquemática se muestra en la Figura 2, sólo con pagos fraudulentos. De esta manera será posible enriquecer los registros de pagos que en el pasado fueron rechazados por ser riesgosos para poder tomarlos en cuenta en futuros entrenamientos.

Por otro lado, el autoencoder presenta en su capa central o capa latente una representación reducida y codificada de la entrada. Es de esperarse que a partir de ella puedan visualizarse patrones de fraudes conocidos y, eventualmente, sacar conclusiones o indicios de patrones de fraudes sin conocer. El potencial de dicha representación puede verse en la Figura 3.

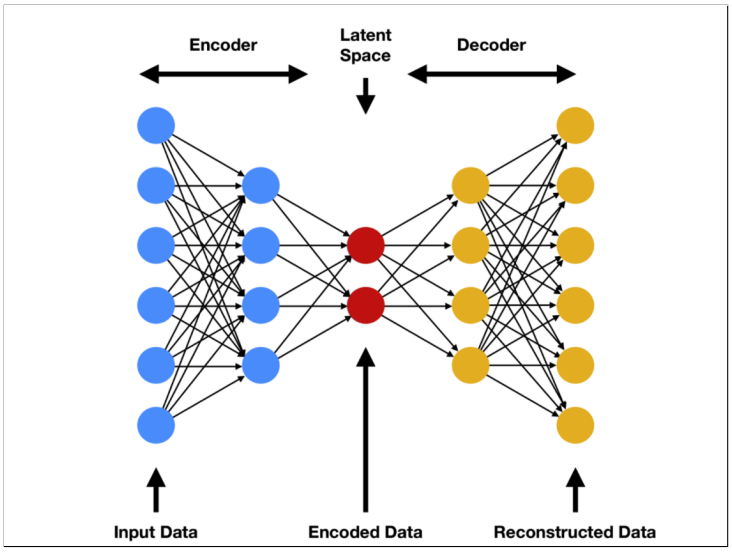
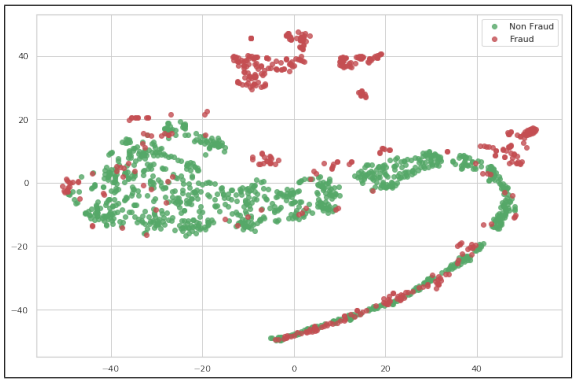
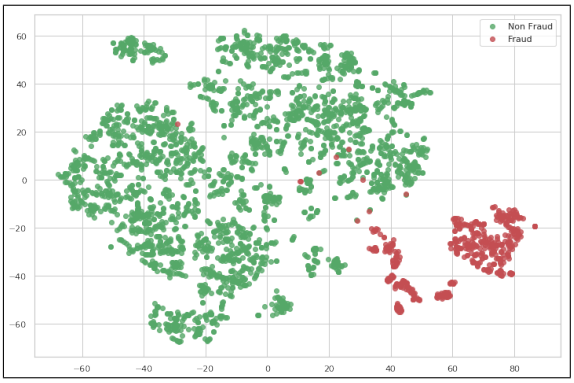


Figura 2. Arquitectura representativa de un autoencoder.



(a) Descomposición de datos de entrada.



(b) Descomposición de datos latentes.

Figura 3. Imagen ilustrativa del artículo *'Semi Supervised Classification using AutoEnconders'* de Kaggle usando el método de descomposición T-SNE (*t-Distributed Stochastic Neighbor Embedding*) aplicado a los datos.

En la Figura 4 se observa un diagrama de bloques que ilustra cómo sería el funcionamiento del sistema en producción. En primer lugar, un usuario realizaría un pago en línea. Inmediatamente, el pago entra al sistema y se obtiene una representación vectorizada del mismo con los atributos de interés. Esta codificación del pago pasa por la red neuronal del motor de fraude y se obtiene una probabilidad de que el pago sea fraudulento (predicción). En función de la probabilidad de fraude y otros factores se decide si el pago se rechaza o se aprueba y se guardan todos los valores en una base de datos. En caso de que el pago sea rechazado, el mismo se envía al autoencoder de Fraude. Del autoencoder se obtendrá una puntuación de fraude asociada al pago rechazado que también se guardará en la base de datos.

La puntuación de fraude será una medida de la capacidad de reconstrucción del autoencoder y representará qué tan similar a un fraude real es el pago rechazado. Al momento de entrenar nuevos modelos de red o reentrenar modelos existentes, los pagos con alta puntuación podrán ser considerados en el dataset de entrenamiento.

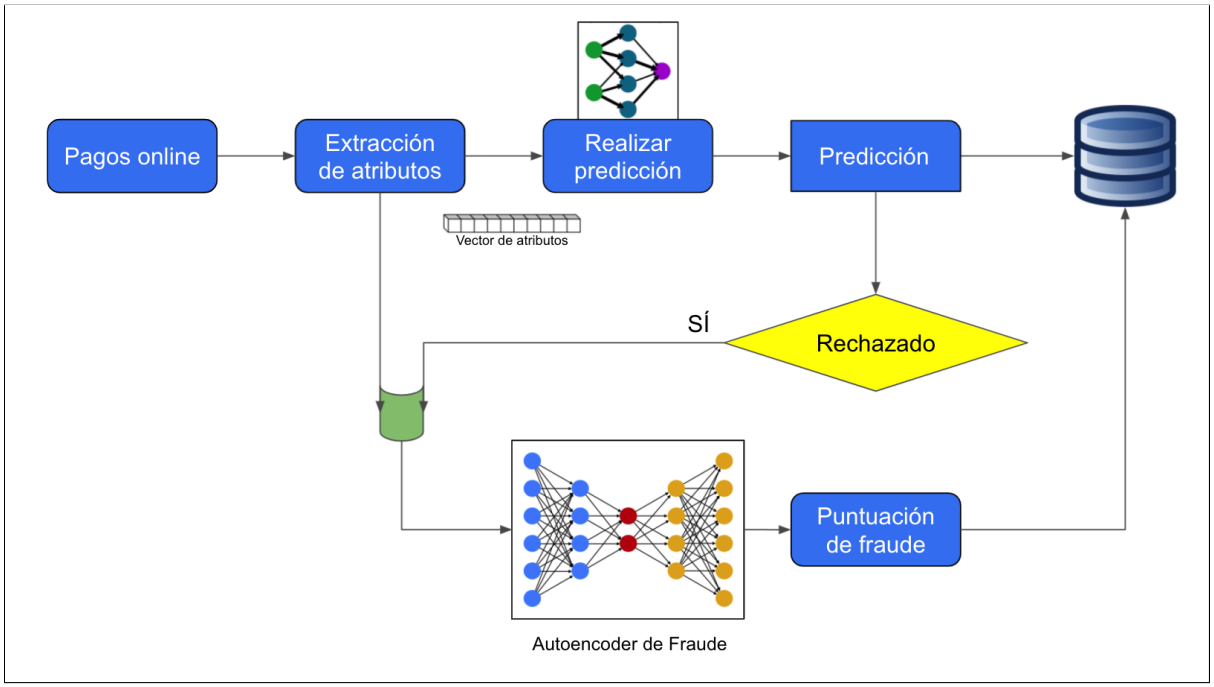


Figura 4. Diagrama de bloques del funcionamiento del sistema.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Responsable	Ariel Salassa	Mercado Libre	ML Engineer Alumno
Colaboradores	Ing. Paz Martin Lcdo. Joaquín Loyola Ing. Enrique Serdio	Mercado Libre	Sr. Data Scientist Sr. Data Engineer Sr. ML Engineer
Orientador	M. Sc. Lcdo. Franco Arito	Mercado Libre	Sr. ML Expert Director Trabajo final
Usuario final	Desarrolladores de Machine Learning	Mercado Libre	Data Scientists ML Engineers

Cuadro 1. Identificación de los interesados.

- Responsable: Ariel Salassa, es la persona que desarrollará el proyecto.
- Colaboradores:
  - Paz Martín: es líder y referente técnica del equipo de científicos de datos donde se desempeña el responsable. Validará la gestión del tiempo y será capaz de orientar en el desarrollo si el responsable lo requiriese.
  - Joaquín Loyola: es líder y referente técnico del equipo de ingeniería de datos. Su colaboración pasará por asistir al referente en cuestiones ligadas a los datos de entrenamiento, si fuese necesario.
  - Enrique Serdio: es referente técnico del equipo de ML Ops. Su colaboración se centrará, si fuese necesario, en asistir al responsable en cuestiones ligadas a la infraestructura de los modelos de machine learning en la nube.

- Orientador: Franco Arito es el director del presente proyecto y líder técnico de múltiples equipos de Mercado Libre. Su función será orientar al responsable a lo largo de la realización del proyecto.
- Desarrolladores de Machine Learning: son los usuarios finales que podrán hacer uso del sistema para enriquecer sus modelos.

3. Propósito del proyecto

El propósito de este proyecto es poner en valor los pagos que son rechazados por el motor de fraude y que tienen potencial de ser utilizados en futuros entrenamientos de redes neuronales de manera tal de reducir el desbalance de los datasets de entrenamiento y validación. Además, se espera que la representación en la capa latente permita evaluar oportunidades para determinar perfiles de fraude. Con una representación como esta, los equipos de prevención tendrán a su disposición una herramienta que les permitirá ser más reactivos ante posibles ataques.

4. Alcance del proyecto

El proyecto comprenderá las siguientes etapas:

- Planificación de tareas.
- Formación en TensorFlow.
- Investigación de autoencoders aplicados a la prevención de fraude.
- Selección y extracción del dataset para realizar prueba de concepto del modelo.
- Análisis de datos del dataset.
- Pruebas de arquitectura de red.
- Visualización y análisis de datos de la capa latente utilizando el método de descomposición T-SNE.
- Evaluación de distintas formas de hacer etiquetado (labeling).
- Evaluación de la performance del sistema comparado con otras soluciones.
- Evaluación del modelo con otros datasets.

El presente proyecto no incluye:

- Aplicación de algoritmos de clustering para los datos codificados a partir de la capa latente.
- Despliegue del modelo y puesta en producción.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- El responsable dispondrá de suficiente cantidad de tiempo para encarar los problemas que se presenten en el desarrollo del proyecto.
- El responsable tendrá a su disposición a su director y/o colaboradores cuando sea pertinente.
- TensorFlow es el framework de cálculo numérico que dispone de todas las herramientas necesarias para encarar este proyecto.
- El autoencoder entrenado solamente con pagos fraudulentos tendrá buen ratio de reconstrucción de datos a la hora de evaluar pagos rechazados por alto riesgo.
- La puntuación de fraude (asociada con la medida de reconstrucción de un pago) será un dato de tipo flotante, o bien, un dato de tipo categórico basado en ciertos valores de corte (thresholds).
- Es posible aplicar el método de descomposición T-SNE a los datos codificados y, a partir de su representación en dos o tres dimensiones, se podrán realizar nuevos análisis, por ejemplo, la identificación de clusters de fraudes.
- Una vez que el autoencoder esté entrenado y validado con un set de pagos, su aplicación podrá generalizarse.
- El comportamiento de los usuarios que provocan el fraude no mutará mientras tiene lugar el desarrollo de este proyecto.

6. Requerimientos

1. Requerimientos de documentación
   1.1. Toda documentación compartida debe mantenerse dentro de un acuerdo de confidencialidad.
   1.2. El trabajo debe ser continuamente documentado y se presentarán informes de avance una vez cada tres semanas al director.
   1.3. Los informes de avance pueden ser presentados como código correctamente documentado con los resultados correspondientes.
2. Requerimientos de forma trabajo
   2.1. Se utilizará una metodología de trabajo ágil e iterativa con mucha interacción entre el responsable, el director y los colaboradores.
3. Requerimientos de lenguajes y frameworks
   3.1. Los datos deberán ser consultados a base de datos relacionales. El lenguaje para estas transacciones debe ser SQL y debe ser lo más agnóstico posible intentando de no usar funciones que sean específicas de uno y otro proveedor.
   3.2. El framework utilizado debe ser Tensor Flow en su versión V2.0 o superior en Python.

3.3. Todo análisis debe realizarse en código Python dentro de Jupyter labs y utilizando librerías standard (numpy, pandas, matplotlib, seaborn, etcétera).

4. Requerimientos de infraestructura

   4.1. Las queries de extracción de datos deben ser compatibles con Amazon Redshift y/o Google BigQuery.

   4.2. Los datasets deben ser guardados en Amazon S3 o Google Cloud Storage como archivos con extensión *.csv*.

   4.3. En caso de ser necesario un hardware específico de entrenamiento, deberán usarse los servicios de Google Cloud Platform (GCP).

5. Requerimientos funcionales

   5.1. La extracción de datos no puede demorar más de 24 hs.

   5.2. El modelo entrenado debe tener una precisión de al menos 85 %.

   5.3. El modelo debe ser entrenado con al menos diez mil registros.

   5.4. El entrenamiento del modelo no puede demorar más de 24 hs.

   5.5. El tipo de dato que represente la puntuación de fraude debe ser categórico o flotante.

   5.6. Las representaciones resultantes de la descomposición deben poder visualizarse en dos o tres dimensiones.

6. Requerimientos de testing y evaluación

   6.1. La efectividad de la puntuación de fraude debe ser evaluada contra una marca dada por una heurística conocida y la efectividad de la puntuación debe ser mejor que la efectividad de la marca.

7. Historias de usuarios (*Product backlog*)

La medida del trabajo a efectuar para cumplir con cada una de las historias de usuarios estará dada por *story points*. Para ponderar los esfuerzos se utilizará la serie de Fibonacci con valores: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc. Cuando el esfuerzo se considere alto los *story points* tomarán valores entre 0 y 3 inclusive. Cuando se considere medio, entre 5 y 13 inclusive. Cuando el esfuerzo se considere alto el valor de los *story points* será igual o mayor a 21.

- Como analista y desarrollador quiero una métrica de fraude para reutilizar pagos rechazados por alto riesgo en futuros entrenamientos y futuros análisis.

  Dificultad: Alta (34) – Complejidad: Alta (34) – Incertidumbre: Alta (34)

  Story Points: $34 + 34 + 34 = 102 \rightarrow 89$

- Como cliente quiero tener un modelo de red realizado con tecnologías standards y *open source* para poder mantenerlo a futuro con el menor esfuerzo posible.

  Dificultad: Media (8) – Complejidad: Media (8) – Incertidumbre: Baja (1)

  Story Points: $8 + 8 + 1 = 17 \rightarrow 21$

- Como analista de datos quiero tener una representación simplificada del fraude para entender posibles ataques y ser reactivo en consecuencia.

  Dificultad: Alta (21) – Complejidad: Alta (21) – Incertidumbre: Alta (21)

  Story Points: $21 + 21 + 21 = 63 \rightarrow 55$

- Como cliente quiero tener las bases de un modelo de red preciso para servirlo en producción adaptándose a flujos preexistentes.

  Dificultad: Media (8) – Complejidad: Media (8) – Incertidumbre: Alta (21)

  Story Points: $8 + 8 + 21 = 37 \rightarrow 34$

- Como cliente quiero recibir la documentación del trabajo realizado para que pueda servir como base de futuros desarrollos de la empresa.

  Dificultad: Media (13) – Complejidad: Baja (2) – Incertidumbre: Baja (1)

  Story Points: $13 + 2 + 1 = 16 \rightarrow 13$

- Como cliente quiero tener una presentación resumida del trabajo para mostrar sus resultados y su potencial potencial a todos los interesados.

  Dificultad: Baja (3) – Complejidad: Baja (2) – Incertidumbre: Baja (2)

  Story Points: $3 + 2 + 2 = 7 \rightarrow 8$

8. Entregables principales del proyecto

Los entregables del proyecto que conservará la empresa donde trabaja el responsable y el director son:

- Informe final.
- Presentación final.
- Datasets utilizados.
- Queries de extracción documentadas.
- Jupyter labs de análisis documentados.
- Informe de avance.

Además, se entregará a los docentes responsables de la Carrera de Especialización en Inteligencia Artificial de la UBA informe de avance y el informe final del proyecto con firma previa de los documentos de confidencialidad.

9. Desglose del trabajo en tareas

1. Planificación. (60 hs)
   1.1. Estudio de necesidades. (9 hs)
   1.2. Análisis de factibilidad. (3 hs)
   1.3. Definición de requerimientos. (9 hs)
   1.4. Confección de documento de planificación. (39 hs)
2. Investigación y capacitación. (114 hs)
   2.1. Estudio de distintos tipos de autoencoders. (9 hs)

2.2. Estudio de autoencoders aplicados a la prevención de fraude. (3 hs)
2.3. Capacitación en Tensor Flow 2. (15 hs)
2.4. Capacitación en *feature preprocessing* y *feature engineering*. (24 hs)
2.5. Capacitación en *feature selection*. (15 hs)
2.6. Capacitación en SQL aplicado a Amazon Redshift y Google BigQuery. (9 hs)
2.7. Capacitación en *Machine Learning pipelines* en GCP. (15 hs)
2.8. Estudio de algoritmo T-SNE. (9 hs)
2.9. Elaboración de códigos de ejemplos básicos. (15 hs)

3. Confección dataset de prueba. (45 hs)
   3.1. Exploración y elección tablas. (9 hs)
   3.2. Análisis de completitud de datos. (9 hs)
   3.3. Confección de query de extracción. (9 hs)
   3.4. Extracción de datos. (3 hs)
   3.5. Selección de features. (15 hs)
4. Entrenamiento. (78 hs)
   4.1. Aplicación de *feature preprocessing*. (15 hs)
   4.2. Prueba de distintas arquitecturas de red con distintas configuraciones. (39 hs)
   4.3. Evaluación y ajuste del modelo. (24 hs)
5. Pruebas de validación. (63 hs)
   5.1. Evaluación de distintas estrategias de *labeling*. (39 hs)
   5.2. Comparación de resultados en función de heurísitcas conocidas. (24 hs)
6. Representación reducida. (63 hs)
   6.1. Visualización de datos de la capa de entrada utilizando el método de descomposición T-SNE. (15 hs)
   6.2. Visualización de datos de la capa latente utilizando el método de descomposición T-SNE. (15 hs)
   6.3. Visualización segmentada de los datos en función de features de interés del modelo. (33 hs)
7. Generalización. (87 hs)
   7.1. Extracción de datos correspondientes a otro flujo de datos. (24 hs)
   7.2. Análisis de completitud de datos. (15 hs)
   7.3. Comparación y análisis de resultados. (24 hs)
   7.4. Ajustes finales. (24 hs)
8. Documentación y presentación final. (90 hs)
   8.1. Elaborar informe de avance proyecto. (40 hs)
   8.2. Elaborar informe final del proyecto. (40 hs)
   8.3. Preparación de presentación final. (30 hs)

Cantidad total de horas: (630 hs)

10. Diagrama de Activity On Node

En la Figura 5 se muestra el diagrama de *Activity on Node* del proyecto. Las flechas resaltadas en negro ilustran el camino crítico del proyecto. También se puede ver que los hitos marcan el fin de las distintas fases del proyecto, las cuales, a su vez, están representadas en distintos colores.

La primera fase está comprendida por tareas previas al desarrollo en código del proyecto. En esta fase se considera planificación y capacitación.

La segunda fase involucra todo el *machine learning pipeline* necesario para evaluar entrenar un modelo y dejarlo preparado para en un futuro servirlo en producción.

La tercera fase toma como punto de partida un modelo de calidad ya entrenado e intenta agregarle valor al mismo mediante la generalización de resultados y la representación reducida de los pagos para intentar obtener otras posibles conclusiones adicionales relacionadas con potenciales perfiles de fraude.

Finalmente, la última fase, involucra toda la realización de toda la documentación del trabajo realizado.

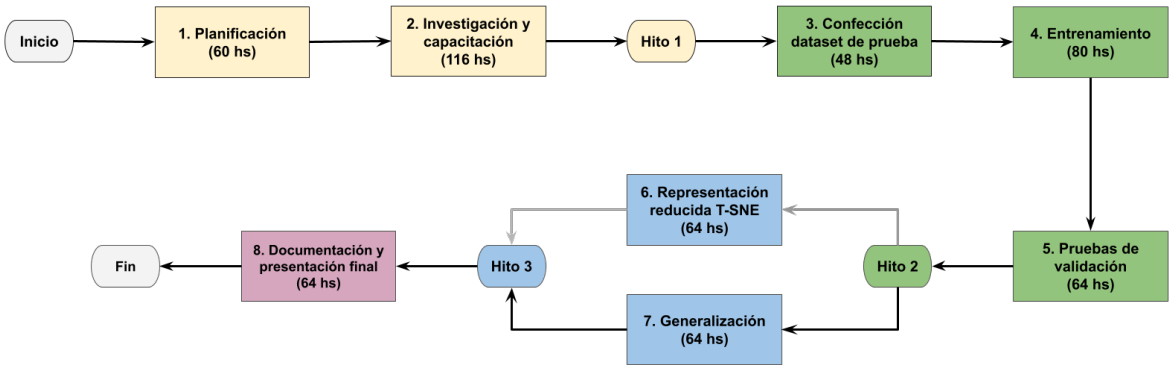


Figura 5. Diagrama en *Activity on Node.*

11. Diagrama de Gantt

A continuación se muestra el diagrama de Gant del presente proyecto. Se consideró la jornada laboral de 3 horas de trabajo desde la fecha de inicio del curso hasta finales de abril del próximo año. En la figura 6 y 7 se muestra el diagrama de Gantt de forma compacta y de forma desglosada respectivamente, tal como se enumeró en la sección 9.

#	Title	Duration	Start	End
0	CEIA - Gantt		24/6/21	25/4/22
1	Inicio		24/6/21	24/6/21
2	**Planificación**		**24/6/21**	**20/7/21**
7	**Investigación y capacitación**		**21/7/21**	**24/9/21**
17	Hito 1		24/9/21	24/9/21
18	**Confección dataset de prueba**		**25/9/21**	**5/11/21**
24	**Entrenamiento**		**8/11/21**	**13/12/21**
28	**Pruebas de validación**		**14/12/21**	**11/1/22**
31	Hito 2		11/1/22	11/1/22
32	**Representación reducida**		**12/1/22**	**9/2/22**
36	**Generalización**		**10/2/22**	**22/3/22**
41	Hito 3		22/3/22	22/3/22
42	**Documentación y presentación final**		**23/3/22**	**25/4/22**
45	Fin		25/4/22	25/4/22

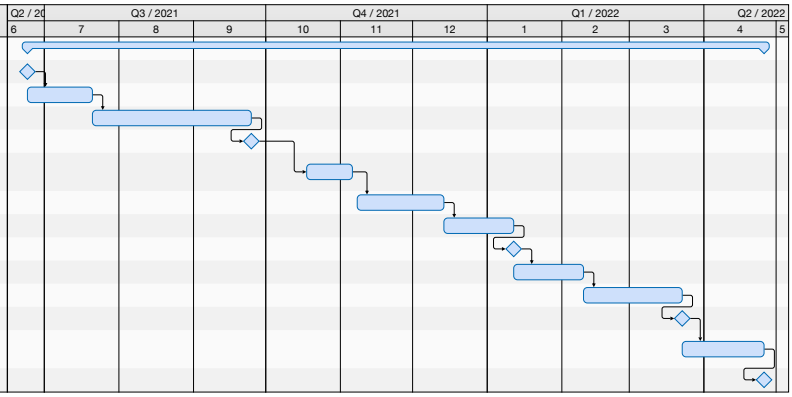
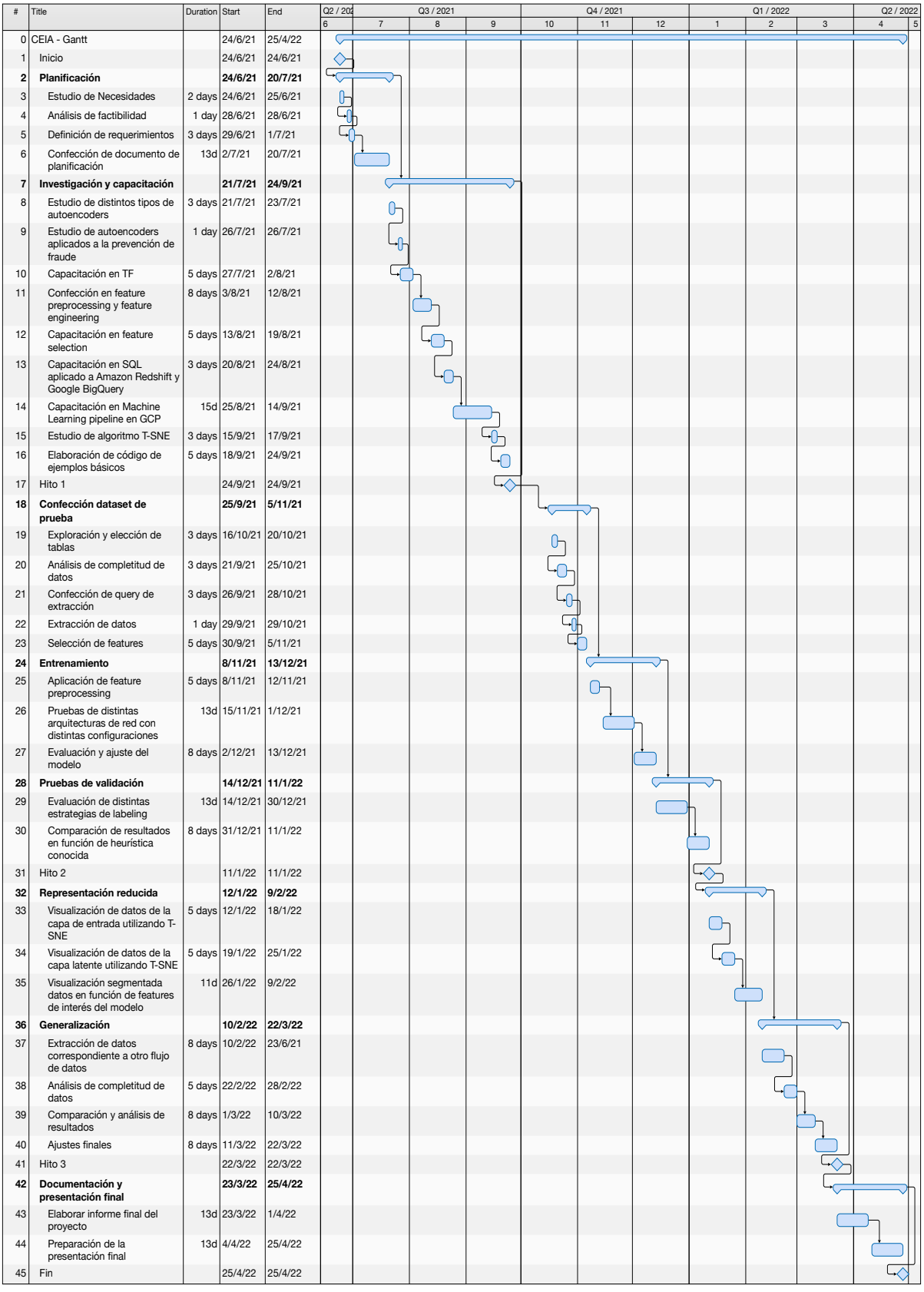


Figura 6. Diagrama de Gantt reducido.



#	Title	Duration	Start	End
0	CEIA - Gantt		24/6/21	25/4/22
1	Inicio		24/6/21	24/6/21
2	**Planificación**		**24/6/21**	**20/7/21**
3	Estudio de Necesidades	2 days	24/6/21	25/6/21
4	Análisis de factibilidad	1 day	28/6/21	28/6/21
5	Definición de requerimientos	3 days	29/6/21	1/7/21
6	Confección de documento de planificación	13d	2/7/21	20/7/21
7	**Investigación y capacitación**		**21/7/21**	**24/9/21**
8	Estudio de distintos tipos de autoencoders	3 days	21/7/21	23/7/21
9	Estudio de autoencoders aplicados a la prevención de fraude	1 day	26/7/21	26/7/21
10	Capacitación en TF	5 days	27/7/21	2/8/21
11	Confección en feature preprocessing y feature engineering	8 days	3/8/21	12/8/21
12	Capacitación en feature selection	5 days	13/8/21	19/8/21
13	Capacitación en SQL aplicado a Amazon Redshift y Google BigQuery	3 days	20/8/21	24/8/21
14	Capacitación en Machine Learning pipeline en GCP	15d	25/8/21	14/9/21
15	Estudio de algoritmo T-SNE	3 days	15/9/21	17/9/21
16	Elaboración de código de ejemplos básicos	5 days	18/9/21	24/9/21
17	Hito 1		24/9/21	24/9/21
18	**Confección dataset de prueba**		**25/9/21**	**5/11/21**
19	Exploración y elección de tablas	3 days	16/10/21	20/10/21
20	Análisis de completitud de datos	3 days	21/9/21	25/10/21
21	Confección de query de extracción	3 days	26/9/21	28/10/21
22	Extracción de datos	1 day	29/9/21	29/10/21
23	Selección de features	5 days	30/9/21	5/11/21
24	**Entrenamiento**		**8/11/21**	**13/12/21**
25	Aplicación de feature preprocessing	5 days	8/11/21	12/11/21
26	Pruebas de distintas arquitecturas de red con distintas configuraciones	13d	15/11/21	1/12/21
27	Evaluación y ajuste del modelo	8 days	2/12/21	13/12/21
28	**Pruebas de validación**		**14/12/21**	**11/1/22**
29	Evaluación de distintas estrategias de labeling	13d	14/12/21	30/12/21
30	Comparación de resultados en función de heurística conocida	8 days	31/12/21	11/1/22
31	Hito 2		11/1/22	11/1/22
32	**Representación reducida**		**12/1/22**	**9/2/22**
33	Visualización de datos de la capa de entrada utilizando T-SNE	5 days	12/1/22	18/1/22
34	Visualización de datos de la capa latente utilizando T-SNE	5 days	19/1/22	25/1/22
35	Visualización segmentada datos en función de features de interés del modelo	11d	26/1/22	9/2/22
36	**Generalización**		**10/2/22**	**22/3/22**
37	Extracción de datos correspondiente a otro flujo de datos	8 days	10/2/22	23/6/21
38	Análisis de completitud de datos	5 days	22/2/22	28/2/22
39	Comparación y análisis de resultados	8 days	1/3/22	10/3/22
40	Ajustes finales	8 days	11/3/22	22/3/22
41	Hito 3		22/3/22	22/3/22
42	**Documentación y presentación final**		**23/3/22**	**25/4/22**
43	Elaborar informe final del proyecto	13d	23/3/22	1/4/22
44	Preparación de la presentación final	13d	4/4/22	25/4/22
45	Fin		25/4/22	25/4/22

Figura 7. Diagrama de Gantt desglosado en tareas.

12. Presupuesto detallado del proyecto

En esta sección se detallan los gastos del proyecto. Las unidades del valor unitario están dadas en dólares estadounidenses.

Las cantidades del uso de los servicios son estimadas. En el caso de Amazon Redshift y Google BigQuery se ha considerado el gasto mensual fijo del uso del servicio que paga la empresa. En el caso de Amazon Redshift se ha estimado que se almacenará 1 Terabyte de información durante 10 meses. Finalmente, para el caso de entrenamientos, se ha considerado que los entrenamientos de los experimentos durarán 50 horas y tendrán lugar durante un mes.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total [USD]
Mano de obra	600 horas	10 USD/hora	6300
Google BigQuery	1 mes	1700 USD/mes	1700
Amazon Redshift	1 mes	1380 USD/mes	1380
Amazon S3	1 TB 10 meses	0.0.23 USD/GB/mes	235.52
Google AI Platform	1 mes	61.05 USD/mes	61.05
SUBTOTAL			9676.57
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total [USD]
30 % del costo directo	-	-	2812.97
SUBTOTAL			2902.97
TOTAL			12579.54

13. Gestión de riesgos

a) Identificación de los riesgos del proyecto:

Riesgo 1: falta de disponibilidad de los colaboradores.

- Severidad (3): ante la ausencia de alguno de los colaboradores, se hará más largo el desarrollo en los cuellos de botella del proyecto.
- Probabilidad de ocurrencia (7): hay mucha rotación de empleados en la empresa y colaborador podría dejarla durante el transcurso del proyecto.

Riesgo 2: falta de tiempo del responsable.

- Severidad (5): se atrasarían las tareas, no se cumpliría con la planificación y, en el peor de los casos, no se lograría completar el proyecto para la fecha de presentación establecida.
- Probabilidad de ocurrencia (4): en la empresa donde trabaja el responsable surgen contingencias que son prioritarias.

Riesgo 3: no llegar a un 85 % de precisión en el entrenamiento.

- Severidad (8): no se contaría con una red entrenada de calidad y los resultados de la misma no serían tan fiables.

- Probabilidad de ocurrencia (3): puede ocurrir que los datos seleccionados para el entrenamiento no sean lo suficientemente representativos.

Riesgo 4: comportamiento inesperado del modelo a la hora de predecir.

- Severidad (9): si el modelo tiene un buen ratio de reconstrucción para pagos fraudulentos y resulta que también tiene un buen ratio de reconstrucción para los pagos no fraudulentos no se cumpliría la hipótesis planteada en este trabajo.
- Probabilidad de ocurrencia (3): es posible que la red no aprenda los patrones que permitan distinguir un pago fraudulento de un pago no fraudulento.

Riesgo 5: efectividad de puntuación de fraude menor a marca de heurística.

- Severidad (9): si esto sucede no se cumpliría con la motivación del trabajo.
- Probabilidad de ocurrencia (4): puede que la red neuronal no aprenda patrones que permitan separar pagos fraudulentos de no fraudulentos y, por ello, la puntuación de fraude puede no ser eficiente. También puede ocurrir que el modelo se comporte de la manera esperada pero que aún así no llegue a superar la eficiencia de la marca heurística.

Riesgo 6: no llegar a distinguir patrones en la representación reducida.

- Severidad (3): no es tan grave que no se puedan distinguir patrones de fraude dentro del set de datos. No es la motivación principal de este proyecto.
- Probabilidad de ocurrencia (5): es una hipótesis donde se tiene gran incertidumbre.

b) Tabla de gestión de riesgos: (El RPN se calcula como RPN=SxO)

Riesgo	S	O	RPN	S*	O*	RPN*
Falta de disponibilidad de los colaboradores	3	7	21	-	-	21
Falta de tiempo del responsable	5	4	20	-	-	20
No llegar a un 85 % de precisión en el entrenamiento	8	3	24	6	2	12
Comportamiento inesperado del modelo a la hora de predecir	9	3	27	8	2	16
Efectividad de puntuación de fraude menor a marca de heurística	9	4	36	8	2	16
No llegar a distinguir patrones en la representación reducida	3	5	15	-	-	15

Criterio adoptado: se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 21.

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 3: no llegar a un 85 % de precisión en el entrenamiento.

- Plan de mitigación: en primer lugar se atacará la red. Se seguirán probando distintas arquitecturas y distintos hiperparámetros. Si eso no da resultado se buscarán más datos y se hará una nueva extracción con más registros y más *features*.
- Severidad (6): cada uno de los planes de mitigación conlleva un tiempo adicional no menor por lo que la planificación del proyecto podría verse atrasada.
- Probabilidad de ocurrencia (2): haciendo todos los ajustes posibles debería alcanzarse la precisión deseada.

Riesgo 4: comportamiento inesperado del modelo a la hora de predecir.

- Plan de mitigación: se invertirá el paradigma del trabajo. El objetivo se planteó de manera tal de entrenar un autoencoder con pagos fraudulentos para estimar, dentro de los pagos rechazados, cuántos estuvieron bien rechazados. La mitigación consistirá en entrenar el autoencoder con pagos no fraudulentos para medir, dentro de los rechazados, cuántos estuvieron mal rechazados. Esto puede hacerse porque la cantidad de pagos no fraudulentos es mucho mayor que la cantidad de pagos fraudulentos y, además, la dinámica del comportamiento del usuario no fraudulento es mucho menor que la dinámica de comportamiento del usuario no fraudulento.
- Severidad (8): tomar el enfoque del plan de mitigación seguramente daría buenos resultados pero se perdería mucho tiempo en nuevas extracciones y pre procesamiento de datos. Además la representación reducida de dichos pagos no tendría mucho valor.
- Probabilidad de ocurrencia (2): es muy improbable que con la gran cantidad de datos disponibles no se pueda ajustar la red para que pueda predecir correctamente.

Riesgo 5: efectividad de puntuación de fraude menor a marca de heurística.

- Plan de mitigación: se convocará al director o a alguno de los colaboradores para que participen activamente en la mejora.
- Severidad (8): poner mucho recurso humano a mitigar el problema seguramente impactará en grandes retrasos en la planificación.
- Probabilidad de ocurrencia (2): sería raro que surgiese este problema sin que se hubiese advertido y eventualmente, mitigado, en pasos anteriores.

14. Gestión de la calidad

- Req 1.1: toda documentación compartida debe mantenerse dentro de un acuerdo de confidencialidad.
  - Verificación: se firmará acuerdo de confidencialidad entre las partes.
  - Validación: revisión y conformidad por parte del director.
- Req 1.2: el trabajo debe ser continuamente documentado y se presentarán informes de avance una vez cada tres semanas al director.

  - Verificación: se controlará que cada avance tenga la documentación necesaria de manera que alguno de los colaboradores pueda entender el avance sin entrar en detalles de implementación.
  - Validación: revisión y conformidad por parte del director.

- Req 1.3: los informes de avance pueden ser presentados como código correctamente documentado con los resultados correspondientes.
  - Verificación: se controlará que cada avance tenga la documentación necesaria de manera que alguno de los colaboradores pueda entender el avance sin entrar en detalles de implementación.
  - Validación: revisión y conformidad por parte del director.

- Req 2.1: se utilizará una metodología de trabajo ágil e iterativa con mucha interacción entre el responsable, el director y los colaboradores.
  - Verificación: se realizarán todas las ceremonias que correspondan y se comentarán los problemas y/o avances en las *dailys* que correspondan.
  - Validación: N/A (requerimiento interno).

- Req 3.1: los datos deberán ser consultados a base de datos relacionales. El lenguaje para estas transacciones debe ser SQL y debe ser lo más agnóstico posible (intentar no usar funciones que sean específicas de uno y otro proveedor).
  - Verificación: se probarán las queries en distintos proveedores para asegurarse de que corra en todos.
  - Validación: N/A (requerimiento interno).

- Req 3.2: el framework utilizado debe ser Tensor Flow en su versión V2.0 o superior en Python.
  - Verificación: al comienzo de cada sesión de trabajo se utilizará el comando *pip show tesorflow* para verificar la versión utilizada.
  - Validación: N/A (requerimiento interno).

- Req 3.3: todo análisis debe realizarse en código Python dentro de Jupyter labs y utilizando librerías standard (numpy, pandas, matplotlib, seaborn, etcétera).
  - Verificación: todo el código escrito se volcará en una máquina virtual provista de un Jupyter Lab con todas las librerías standard pre instaladas.
  - Validación: N/A (requerimiento interno).

- Req 4.1: las queries de extracción de datos deben ser compatibles con Amazon Redshift y/o Google BigQuery.
  - Verificación: se probarán las queries en uno y otro proveedor con registros limitados.
  - Validación: N/A (requerimiento interno).

- Req 4.2: los datasets deben ser guardados en Amazon S3 o Google Cloud Storage como archivos con extensión .csv.
  - Verificación: se podrán consultar cada uno de los archivos en el bucket que corresponda.
  - Validación: N/A (requerimiento interno).

- Req 4.3: en caso de ser necesario un hardware específico de entrenamiento, deberán usarse los servicios de Google Cloud Platform (GCP).
  - Verificación: se harán ensayos de entrenamiento en una máquina local y en máquinas remotas que a lo sumo estarán provistas de GPU. En caso de no cumplir con los requerimientos de entrenamiento, se solicitará usar una máquina virtual provista de TPU.
  - Validación: revisión y conformidad por parte del director.
- Req 5.1: la extracción de datos no puede demorar más de 24 hs.
  - Verificación: se harán extracciones de prueba con datos reducidos. En función de los resultados se hará una estimación de cuánto puede llegar a demorar una extracción de calidad para cumplir con el requerimiento.
  - Validación: la extracción se configurará para parar por *timeout* a las 24 hs.
- Req 5.2: el modelo entrenado debe tener una precisión de al menos 85 %.
  - Verificación: se tendrán los resultados en entrenamiento en las tablas de métricas de Tensor Flow para consultar y verificar.
  - Validación: cuando el sistema se ponga en producción se monitoreará para validar la métrica antes mencionada.
- Req 5.3: el modelo debe ser entrenado con al menos diez mil registros.
  - Verificación: se tendrá un contador de registros presentes en la extracción resultante.
  - Validación: N/A (requerimiento interno).
- Req 5.4: el entrenamiento del modelo no puede demorar más de 24 hs.
  - Verificación: se harán entrenamientos de prueba con modelos sencillos para ver cuánto dura un entrenamiento de este estilo. En función de los resultados se hará una estimación de cuánto puede llegar a complejizarse el modelo para cumplir con el requerimiento.
  - Validación: el entrenamiento se configurará para parar por *timeout* a las 24 hs.
- Req 5.5: el tipo de dato que represente la puntuación de fraude debe ser categórico o flotante.
  - Verificación: se realizarán distintas pruebas analizando la dispersión de los resultados obtenidos para saber si es posible simplificar el grado de reconstrucción a una variable categórica binaria.
  - Validación: revisión y conformidad por parte del director.
- Req 5.6: las representaciones resultantes de la descomposición deben poder visualizarse en dos o tres dimensiones.
  - Verificación: una vez que el modelo del autoencoder esté entrenado el modelo de la red se seccionará de forma tal de tener como capa de salida la capa latente. Cada uno de los pagos que forman parte de los set de entrenamiento, validación y test pasarán por esta red de codificación. Una vez que se tienen los datos codificados o latentes, se le aplicará el algoritmo T-SNE y los resultados se graficarán usando matplotlib.
  - Validación: N/A (requerimiento interno).

- Req 6.1: la efectividad de la puntuación de fraude debe ser evaluada contra una marca dada por una heurística conocida y la efectividad de la puntuación debe ser mejor que la efectividad de la marca.
    - Verificación: se tomará el set de validación, se le asignará cada una de las puntuaciones y se verificará la eficiencia de una y otra métrica.
    - Validación: cuando el sistema se ponga en producción se monitoreará una y otra marca para corroborar las pruebas de verificación.

15. Procesos de cierre

Una vez finalizado el proyecto, se procederá a su cierre para lo cual se contemplarán las siguientes actividades, cada una de ellas a cargo del responsable Ariel Salassa:

- Análisis del cumplimiento del Plan de Proyecto original:
    - Se compararán las horas dedicadas a cada tarea con las horas planificadas en el inicio.
    - Se analizará el porcentaje de requerimientos cumplidos.
    - Se evaluará el desempeño de la solución y la satisfacción del equipo de trabajo.
- Identificación de las técnicas y procedimientos útiles e inútiles que se utilizaron, así como los problemas que surgieron y cómo se solucionaron.
    - Se dejará asentado en los informes al director qué fue lo qué se probó, resaltando lo que dio resultado y lo que no, los inconvenientes que surgieron y cómo se solucionaron.
    - Se dejarán escritas las posibles oportunidades de mejora que presente la solución.
- Presentación oral virtual del proyecto y acto de agradecimiento a todos los interesados.
    - Se organizará una sesión virtual donde se presentará el proyecto, su utilidad y su estado al concluir la especialización.