

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
SANTIAGO - CHILE



“PROCEDIMIENTO ETL SOBRE LOS REGISTROS DE  
ACTIVIDAD DE LOS TELESCOPIOS UT DEL  
OBSERVATORIO PARANAL PARA LA GENERACIÓN DE  
DATASETS DEL SISTEMA DE ÓPTICA ACTIVA DE LOS  
ESPEJOS M1.”

IGNACIO ANDRÉS ORTIZ VALDEBENITO

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Pedro Toledo Correa  
Profesor Correferente: Jose Luis Martí Lara

Octubre - 2025



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción):     Memoria o trabajo de título     Tesis de Postgrado

Título del trabajo: PROCEDIMIENTO ETL SOBRE LOS REGISTROS DE ACTIVIDAD DE LOS TELESCOPIOS UT DEL OBSERVATORIO PARANAL PARA LA GENERACIÓN DE DATASETS DEL SISTEMA DE ÓPTICA ACTIVA DE LOS ESPEJOS M1

Nombre del candidato(a): Ignacio Andres Ortiz Valdebenito

Carrera / Grado: Ingeniería Civil Informática

Campus: San Joaquín Departamento: Informática

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Pedro Felipe Toledo Correa, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

### 3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses     12 meses     2 años     3 años     5 años     10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

---

---

---

### 4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 29/11/2025 Firma:  29/11/2025

Estudiante o Candidato(a):

Fecha: 27/11/2025 Firma:  \_\_\_\_\_

*Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.*

## **DEDICATORIA**

Dedicado a mi mamá, a mi papá, a mi hermana, a mi abuela y a mi tata, a mi tío, a Anibal, a Matías Lagos, a Matías Gutierrez, a Matías Dawson.

Por siempre estar ahí, por todo lo que me han enseñado, por todos los buenos y malos ratos, por escucharme y entenderme, por ser ejemplos de que el esfuerzo y la perseverancia construyen caminos, por todo el cariño y por moldearme a lo que soy hoy en día.

Muchas gracias por todo.

## AGRADECIMIENTOS

Primero que todo, agradezco a mi familia, por todo el apoyo incondicional que me dieron siempre.

Luego, agradecer a mi profesor Pedro Toledo, por mostrarme la realidad de la ingeniería informática, por las oportunidades y por ayudarme a descubrir mi vocación.

Agradecer a Rodrigo Parra y Claudio Agurto, por abrirme las puertas del observatorio Paranal, por recibirme con los brazos abiertos y mostrarme todo lo que necesitaba saber y más.

Agradecer a Gonzalo Fernández, por haber hecho posible el viaje al observatorio Paranal, de forma incondicional y desinteresada.

Agradecer a Andres Caraballo, Bastían Fernández, Vicente Llanos e Ignacio Jobse, por sus consejos y su sabiduría.

## RESUMEN

**Resumen**— El sistema de Óptica Activa es un proceso fundamental para los telescopios del observatorio Paranal; por este motivo se ha discutido la implementación de técnicas modernas de soft-computing para el análisis de los datos de log de Óptica Activa. En este trabajo, se desarrolló un procedimiento que permite la extracción de datos de los logs de la Óptica Activa y la disposición de estos datos en datasets estructurados con tal de permitir la adopción de las técnicas anteriormente descritas. El procedimiento retorna datos sobre las iteraciones de corrección, las distribuciones de fuerzas de actuadores bajo el espejo del telescopio y las imágenes tomadas; almacenados en dataframes dedicados y relacionados, tomados bajo la certeza del correcto funcionamiento del telescopio. La cantidad de datos almacenados es baja en comparación a la cantidad de líneas en los log. Se concluye que los datos obtenidos cumplen con las expectativas de calidad, y destina el trabajo a futuro en aumentar el número de datos obtenidos.

**Palabras Clave**— Ingeniería de Datos; UT; Óptica Activa; Análisis de logs, Extracción de datos de logs.

## ABSTRACT

**Abstract**— The Active Optics system is a fundamental process for the telescopes in Paranal Observatory; for these reason, the implementation of modern soft-computing techniques for Active Optic's log data analysis has been discussed. In this paper, a procedure was developed, that allows for Active Optic's log data extraction and saving in structured datasets that grants the adoption of previously described techniques. The procedure returns data about correction iterations, force distributions of actuators under the telescope mirror, and images taken; stored in dedicated and related dataframes, taken under the certainty of the telescope's correct operation. The number of data stored is lower than the number of log lines. In conclusion, the data obtained match quality expectations, and future works is aimed towards increasing the number of data obtained.

**Keywords**— Data Engineering; UT; Active Optics; Log Analysis; Log Data Extraction.

## **GLOSARIO**

VLT: Very Large Telescope.

UT: Unitary Telescope.

AT: Auxiliary Telescope.

VLTI: Very Large Telescope Interferometer.

VISTA: Visible and Infrared Survey Telescope for Astronomy.

VST: VLT Survey Telescope.

ESO: European Southern Observatory.

M1: Espejo Primario

# ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 CONTEXTO	3
1.2 ACTORES INVOLUCRADOS	4
1.3 OBJETIVOS Y ALCANCE	4
CAPÍTULO 2: MARCO CONCEPTUAL	5
2.1 TELESCOPIO VLT	5
2.2 SISTEMA DE ÓPTICA ACTIVA	7
2.3 SENSOR DE FRENTE DE ONDA	9
2.4 LOGS	10
2.5 ANÁLISIS DE LOGS	11
2.6 ESTRUCTURACIÓN DE LOGS	12
2.7 FUENTES DE DATOS	13
2.8 PROBLEMÁTICAS RELEVANTES	14
2.8.1 Identificación de datos deseados	14
2.8.2 Almacenamiento de los datos	16
2.8.3 Formato de almacenamiento	18
2.8.4 Extracción de información de las líneas de log	19
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	23
3.1 Descripción general del proceso	23
3.2 Filtrado de líneas de log	23
3.2.1 Remoción de valores redundantes	23
3.2.2 Cruce con archivo de observaciones	24
3.3 Extracción de datos	25
3.3.1 Plantillas de líneas	26
3.3.2 Construcción de los dataframes	26
3.3.3 Carga a los dataframes	27
3.3.4 Validación de los dataframes	27
3.3.5 Relación de los dataframes	28
3.4 Guardado de los datos	28

<b>CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN</b>	<b>29</b>
4.1 Algoritmo	29
4.1.1 log_formatting()	30
4.1.2 obs_filtering()	30
4.1.3 log_parsing()	30
4.1.4 generate_dataframes()	31
4.1.5 validate_forces()	31
4.1.6 validate_images()	31
4.1.7 link_images()	32
4.1.8 validate_corrections()	32
4.1.9 link_dataframes()	32
4.2 Ejecución	32
4.3 Pruebas	33
4.3.1 Metodología	33
4.4 Resultados	34
4.4.1 Resultados para Filtrado de líneas de log	36
4.4.2 Resultados para Extracción de los datos	37
4.4.3 Resultados para Guardado de los datos	38
<b>CAPÍTULO 5: CONCLUSIONES</b>	<b>39</b>
<b>ANEXOS</b>	<b>40</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>41</b>

## ÍNDICE DE FIGURAS

1	Diagrama frontal de un UT	6
2	Diagrama lateral de un UT	7
3	UT 2 visto desde el exterior	8
4	Montura del M1	8
5	Diagrama de la distribución de los actuadores de fuerza a lo largo de la base del espejo M1	9
6	Espejo M1, con M2 siendo reflejado y M3 en el centro, en el telescopio UT 2	9
7	Espejo M2	10
8	Espejo M3	10
9	Diagrama explicativo del sistema Óptica Activa	11
10	Imágen capturada por un sensor de frente de onda Shack-Hartmann	12
11	Página web de archivo de observación de la ESC	14
12	Diagrama general de las tablas de datos	16
13	Diagrama del almacenamiento de datos directamente relacionados con la corrección de fuerzas	17
14	Diagrama de almacenamiento de datos sobre valores adicionales	18
15	Diagrama de flujo del proceso de solución	29

## ÍNDICE DE TABLAS

1	Líneas encontradas de forma manual	35
2	Líneas encontradas por el algoritmo	35
3	Ventanas de observación y tiempo total de observación	35
4	Comparación de datos extraídos manualmente con datos extraídos mediante el algoritmo desde muestras de líneas de log entre las 04:00 hrs y las 05:00 hrs	36
5	Cantidad de líneas de log filtradas	37
6	Tiempos y cantidad de datos para modos de extracción	37
7	Cantidad de líneas de log extraídas	38
8	Cantidad de datos filtrados	38

## INTRODUCCIÓN

El observatorio Paranal, ubicado en el Norte de Chile, es uno de los más importantes a nivel mundial. Su enfoque en la investigación astronómica se ve apoyado por el VLT, una de las instalaciones astronómicas más avanzadas del mundo, la cuál esta compuesta de numerosos telescopios. Estos telescopios están equipados con el sistema de Óptica Activa, que busca compensar las deformaciones que afectan al espejo primario del telescopio, causadas como producto de su operación.

El sistema de Óptica Activa logra este objetivo al controlar el espejo principal del respectivo telescopio mediante la aplicación de fuerzas sobre él. Esto se genera dentro de un ciclo cerrado, el cuál implica tomar una imagen durante una observación, analizar dicha imagen y medir sus aberraciones ópticas, para luego generar el set de fuerzas que se deben aplicar al espejo principal del telescopio con tal de corregir las perturbaciones, y por último, aplicar las fuerzas al espejo principal antes de repetir el proceso.

Durante todo este proceso se van registrando varios datos e interacciones ocurridas durante las operaciones en archivos llamados registros de log, como es común en modelos de software.

Dadas las nuevas capacidades computacionales de la actualidad, se discute la implementación de nuevas técnicas de soft-computing para analizar los datos de operación de la Óptica Activa. Sin embargo, la forma en la que estos datos son registrados en los logs no permite la aplicación directa de estas nuevas técnicas, y tampoco existe un procedimiento que permita la extracción de un dataset de estos datos para fines similares.

Considerando la problemática descrita anteriormente, el objetivo de esta memoria será el desarrollo de un procedimiento que extraiga los datos de logs generados por el sistema de Óptica Activa, los transforme de manera acorde, y los almacene en la forma de datasets estructurados.

Este documento se divide en las siguientes partes centrales:

- Primero en el Capítulo Definición del Problema, se realiza una definición formal del problema, sumando contexto y razones detrás de la problemática, actores involucrados y el alcance del proyecto.
- Luego, en el Capítulo Marco Conceptual, se investiga tanto el sistema de Óptica Activa como métodos para la extracción de datos de registros de logs, con el fin de recabar antecedentes y poder planificar el desarrollo.
- Luego, en el Capítulo Propuesta de la Solución, se describe de forma teórica los aspectos fundamentales del modelo a desarrollar, incluyendo librerías, etapas, estructura, entre otros.

- Finalmente, durante el Capítulo Validación de la Solución, se ejecuta el modelo descrito en el capítulo anterior, y se analiza la información extraída con tal de examinar si cumple con la lógica y la calidad deseadas.

## CAPÍTULO 1

### DEFINICIÓN DEL PROBLEMA

En el VLT del Observatorio Paranal, se busca disponer de datasets con información útil sobre el comportamiento del sistema de Óptica Activa, extrayéndola de sus registros de log, con el fin de implementar técnicas de soft-computing para la asistencia y mejora de las operaciones del sistema. Sin embargo, actualmente no existe un procedimiento que permita analizar los logs y extraer sets de información, en ciertos rangos de tiempo, con tal de producir los datasets deseados.

La existencia de este procedimiento es esencial para habilitar la discusión sobre la implementación de técnicas de soft-computing, lo cuál no fue prioritario durante la construcción del sistema de Óptica Activa.

La necesidad del procedimiento se debe a que los registros de log generados durante las operaciones del sistema de Óptica Activa usan un formato abstracto y poco estructurado. Sumado a lo anterior, estos logs registran todas las actividades y factores presentes durante la operación de los UTs en un lapso de 24 horas, generando archivos de gran tamaño con información muy variada.

#### 1.1. CONTEXTO

El observatorio Paranal, perteneciente a la ESO, se ubica en el cerro homónimo en el Norte de Chile, y se dedica principalmente a la búsqueda y estudio de galaxias y otras estructuras interestelares. El observatorio, para lograr estos objetivos, cuenta con el VLT, una de las instalaciones astronómicas más avanzadas del mundo [ESO, 1998].

El VLT debe cumplir con altos estándares tecnológicos, con tal de cumplir la visión de avanzar el entendimiento del Universo mediante la disposición de instalaciones de clase mundial [ESO, 1998].

Como telescopio óptico, el VLT usa una serie de espejos para conducir la luz estelar a instrumentos que capturan la misma para la toma de imágenes, análisis del espectro electromagnético e interferometría óptica; todas estas funcionalidades destinadas tanto a fines científicos como para la mejora del funcionamiento de los sistemas involucrados [ESO, 1998].

Para cumplir con estas funciones, el VLT consta principalmente de 4 telescopios primarios, denominados UT, sumado a 4 telescopios auxiliares y otros sistemas complementarios [ESO, 1998].

La forma en la que cada UT capta la luz es la siguiente:

- Primero, la luz es captada por un espejo principal de 8.2 metros de diametro, denominado M1, el cuál gracias a su forma redirige la luz a un espejo de 0.9 metros de diametro, denominado M2 [ESO, 1998].
- Luego este último vuelve a redirigir la luz a un tercer espejo elíptico, denominado M3, el cuál redirige la luz a uno de cuatro puntos focales [ESO, 1998].
- Desde cada punto focal, la luz es procesada por un sensor de frente de onda Shack-Hartmann, y luego estos datos son procesados por un computador central, el cuál interpreta estos datos como imágenes [ESO, 1998].

Las características físicas del M1 lo vuelven susceptible a sufrir deformaciones por diversos factores, como por ejemplo el efecto de la gravedad cuando se inclina. Estas deformaciones, a su vez, pueden causar aberraciones ópticas en las imágenes obtenidas [Wilson et al., 1987].

Con tal de resolver estos problemas, en los UTs se implementa un sistema de soporte bajo el espejo M1, con 150 pistones de funcionamiento hidráulico y neumático que aplican fuerza al espejo, permitiéndole conservar su forma ideal, compensando las deformaciones que pueda tener debido a factores externos [ESO, 1998].

Durante las observaciones de los UTs, sus operadores escogen una estrella guía, cuya luz es capturada y analizada por el sensor Shack-Hartmann, calculando con esta las aberraciones ópticas. En base a estos cálculos, se determina una nueva distribución de fuerzas para los actuadores, los cuáles aplican estas fuerzas en el M1, devolviendo a este último a una forma apropiada para continuar con las observaciones [ESO, 1998].

El ciclo anteriormente descrito corresponde al sistema de Óptica Activa, implementada en todos los UTs, el cuál se ejecuta repetidas veces por cada observación [ESO, 1998].

## 1.2. ACTORES INVOLUCRADOS

Los actores involucrados en el problema corresponden a los elementos participantes en el sistema de Óptica Activa del telescopio VLT, más específicamente la célula M1, los sensores de frente de onda Shack-Hartmann y el software de control que almacena y analiza los logs y los datos entregados por los sensores [ESO, 2011c].

## 1.3. OBJETIVOS Y ALCANCE

EL objetivo detrás de la solución busca desarrollar un procedimiento para extraer los datos no estructurados, transformarlos en datos estructurados, y disponerlos para distintos modelos y sistemas que busquen analizar dichos datos con diversos fines, como por ejemplo mantenimiento o técnicas de soft-computing.

## CAPÍTULO 2

### MARCO CONCEPTUAL

#### 2.1. TELESCOPIO VLT

El VLT, sigla para “Very Large Telescope”, es un telescopio ubicado en el Cerro Paranal a 2635 metros de altura, como parte de la “European Southern Observatory” (o conocida por su sigla ESO). Los fines bajo los cuáles se construyó el VLT corresponden a los siguientes [ESO, 1998]:

- El mayor área de colecta posible, según los recursos disponibles.
- La mayor cobertura de longitud de onda, con tal de explotar completamente todas las ventanas atmosféricas.
- Máxima flexibilidad y amplia diversificación instrumental, permitiendo múltiples usos de las instalaciones, incluyendo observaciones simultáneas de múltiples longitudes de onda.
- Capacidad limitada por la difracción de la mayor línea de base posible.
- Optimización de los procedimientos de operaciones científicas, con tal de permitir la explotación, completa y en tiempo real, de la calidad astronómica del sitio y garantizar un máximo retorno científico.

El VLT se compone de una red de 4 telescopios principales idénticos, denominados como “Unitary Telescopes” (o por sus siglas UT), 4 telescopios auxiliares, denominados como “Auxiliary Telescopes” (o por sus siglas AT), un interferómetro, denominado en inglés como “VLT Interferometer” (o por sus siglas en inglés VLTI) y dos telescopios de survey (denominados por sus siglas en inglés VISTA y VST).

Cada UT posee un espejo principal (denominado M1) de forma cóncava con 8,2 metros de diámetro, instalado en un montura de 22 metros de largo, 10 metros de ancho y 20 metros de alto, como se aprecia en la Figura 4. Dicha montura permite al espejo ajustar hacia donde apunta el UT, rotando horizontal y verticalmente [ESO, 1998], como se muestra en la Figura 1.

Bajo M1, se encuentran 150 actuadores de fuerza axiales; estos son, pistones hidráulicos y neumáticos. Estos actuadores están repartidos en 6 anillos concéntricos, distribuidos en forma de dona, desde donde dan a M1 una determinada forma óptica, establecida por el patrón de fuerza presentado por los actuadores [ESO, 1998]. El M1 es visible en la Figura 6.

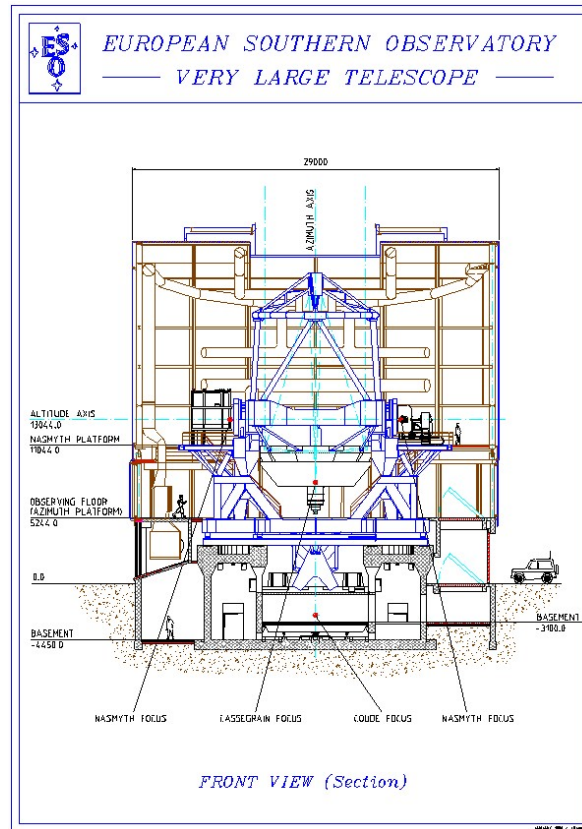


Figura 1: Diagrama frontal de un UT  
Fuente: The VLT White Book [ESO, 1998]

El M2 consiste de un espejo de 0.9 metros de diámetro, montado a una distancia de aproximadamente 12.3 metros de M1 en paralelo y centrado a M1, con el espejo de M2 apuntando a M3 [ESO, 2011b]. El M2 es visible en la Figura 7.

El M3 consiste de un espejo elíptico de 1.24 metros de diámetro mayor con 0.86 metros de diámetro menor. Este se ubica dentro de una torre posicionada en el orificio central del M1. Este puede rotar alrededor de su eje acimutal [ESO, 2011a]. El M3 es visible en la Figura 8.

Luego de que la luz sea redirigida por M3, la luz llega a uno de 3 puntos focales: Nasmyth A, Nasmyth B y Cassegrain. Desde estos puntos la luz es capturada por un sensor de frente de onda (ver Diccionario 1 del Anexo), y finalmente, es capturada por un instrumento de análisis espectrométrico, el cuál varía según el punto focal por el que la luz entró. Cada UT contiene una serie de instrumentos específicos para el análisis a realizar [ESO, 1998].

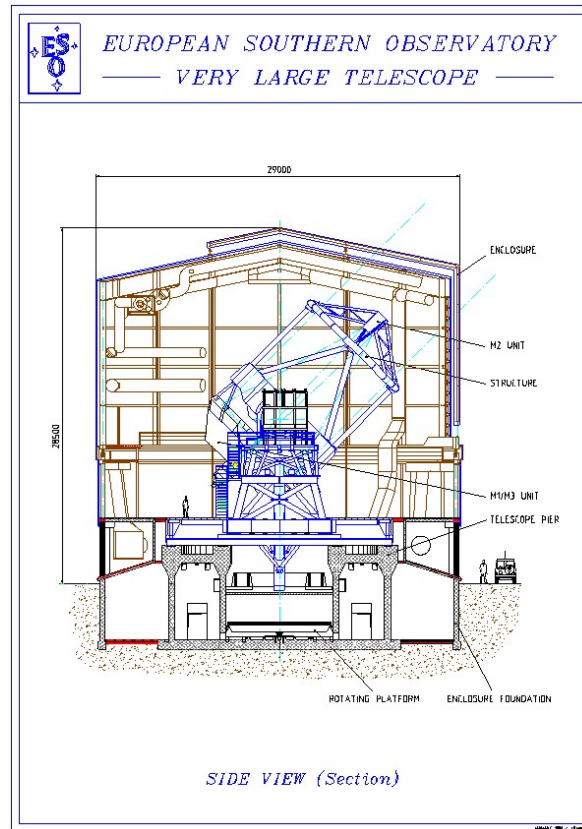


Figura 2: Diagrama lateral de un UT  
Fuente: The VLT White Book [ESO, 1998]

## 2.2. SISTEMA DE ÓPTICA ACTIVA

La Óptica Activa es un sistema integrado en el M1, encargado de corregir las aberraciones y la degradación en la calidad de imagen provocadas por las ópticas del espejo [ESO, 1998].

Estas aberraciones ópticas suelen ser causadas por la sensibilidad de este a perturbaciones ambientales, cómo las distorsiones térmicas, deformación de espejo por ráfagas de viento, errores de manufactura y mantenimiento del telescopio, entre otros. Esta sensibilidad es causada por la baja proporción entre el grosor y el diámetro del espejo, la cuál es la principal característica que permite al espejo M1 tener su gran tamaño, ya que esta proporción es la que permite ajustar el espejo [Wilson et al., 1987].

El ciclo básico del sistema de Óptica Activa se ilustra en la Figura 9. El mismo sigue el siguiente procedimiento:

1. Los operadores del UT escogen una estrella en el cielo, la cuál es designada como estrella guía, y luego toman un imagen de la estrella con el sensor de frente de onda Shack-Hartmann [ESO, 1998].



Figura 3: UT 2 visto desde el exterior  
Fuente: Elaboración propia, tomada el 28/08/2025

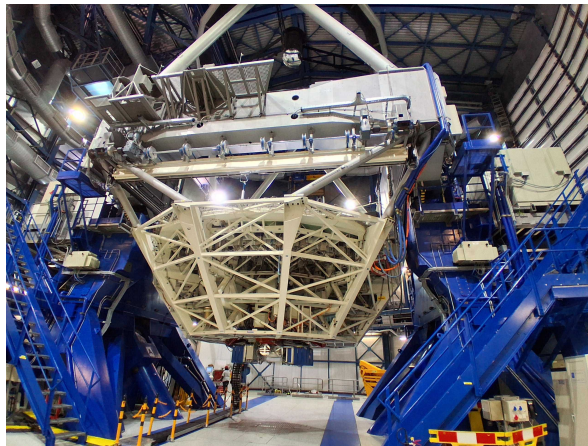


Figura 4: Montura del M1  
Fuente: Elaboración propia, tomada el 28/08/2025

2. Se analiza la imagen y se miden las perturbaciones ópticas, calculando las aberraciones [Wilson et al., 1987].
3. Se genera un set de fuerzas que se debe aplicar al M1 para corregir las perturbaciones [Wilson et al., 1987].
4. Se aplica el set de fuerzas al espejo M1 y, durante una observación, el sensor de frente de onda Shack-Hartmann toma una nueva imagen, repitiendo el proceso [Wilson et al., 1987]. Un ejemplo de imagen tomada por un sensor Shack-Hartmann se muestra en la Figura 10.
5. Finalmente, se retorna al paso 2 [Wilson et al., 1987].

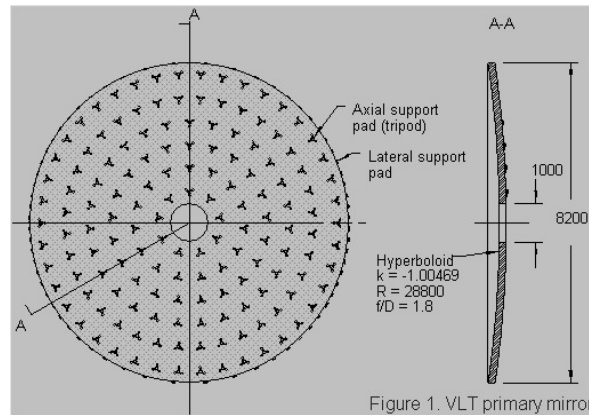


Figura 5: Diagrama de la distribución de los actuadores de fuerza a lo largo de la base del espejo M1

Fuente: The VLT primary mirrors: mirror production and measured performance [ESO, 2011d]

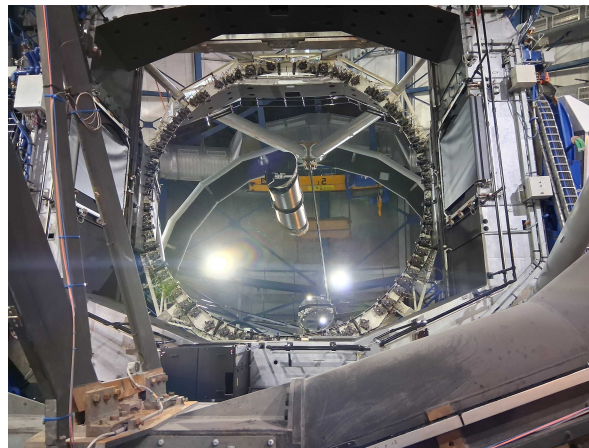


Figura 6: Espejo M1, con M2 siendo reflejado y M3 en el centro, en el telescopio UT 2  
Fuente: Elaboración propia, tomada el 28/08/2025

### 2.3. SENSOR DE FRENTE DE ONDA

Un sensor de frente de onda es un dispositivo electrónico diseñado para calcular las aberraciones ópticas presentes en una imagen tomada, ya sea en el espectro visible o en infrarrojo. El VLT usa CCD Shack-Hartmann como sensores de frente de onda [ESO, 1998].

El objetivo del uso de sensores Shack-Hartmann es la medición de distorsiones del frente de onda (ver Diccionario 2 del Anexo) capturado desde la fuente, sobre las cuáles se realiza el proceso de Óptica Activa. [ESO, 1998]



Figura 7: Espejo M2

Fuente: Elaboración propia, tomada el 28/08/2025



Figura 8: Espejo M3

Fuente: Elaboración propia, tomada el 28/08/2025

## 2.4. LOGS

Logging se refiere al proceso de registrar diferentes eventos y actividades que ocurren dentro de un sistema de software [Jayathilake, 2011]. Estos registros suelen almacenarse en archivos para su posterior análisis, tanto por parte de desarrolladores como de operadores externos.

Los logs son los únicos tipos de datos que registran información de la operación interna de un sistema de software, por lo que su rol en la industria es crucial. Debido a esto mismo, los logs pueden alcanzar tamaños en el orden de hasta el orden de millones de líneas [Ma et al., 2023].

Los logs de los telescopios UT registran un día completo de operación, donde se registra información sobre las condiciones ambientales, el funcionamiento del sistema de Óptica Activa, la toma de imágenes durante periodos de calibración y de observación, lo cuál resulta en

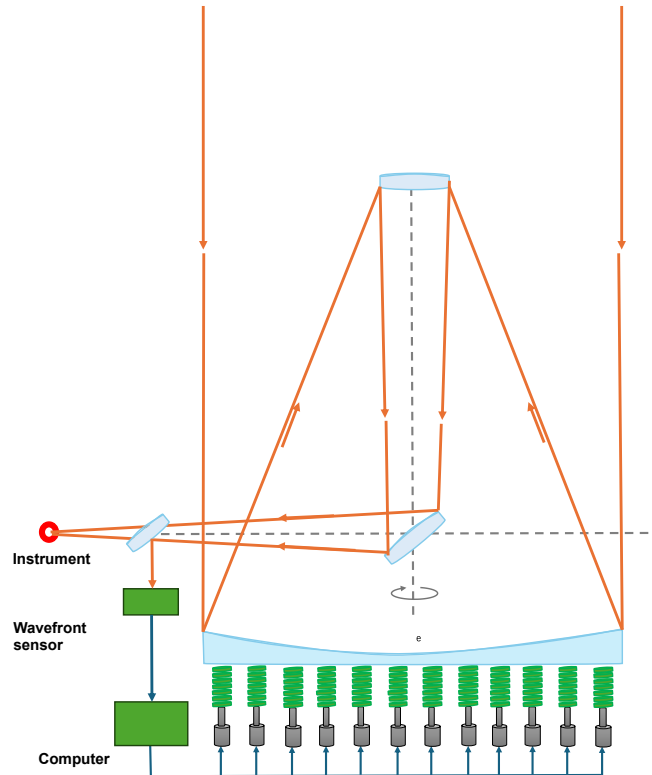


Figura 9: Diagrama explicativo del sistema Óptica Activa

Fuente: Elaboración propia, basado en una diagrama de The VLT White Book [ESO, 1998]

logs que alcanzan tamaños entre, aproximadamente, 50 a 60 MB, con un número de líneas en el orden de cientos de miles. Un ejemplo de estos logs puede verse en el Anexo 1.

Las líneas de un registro de log están compuestas de tokens estáticos y valores dinámicos. Los tokens estáticos corresponden a cadenas de texto que entregan metadata e información complementaria sobre el dato reportado en las líneas, y como tal se repiten de forma sistemática a lo largo del registro. Los valores dinámicos son cadenas de texto o números que entregan el valor del dato reportado, y por ende no se repiten con un patrón claro en el registro [Ma et al., 2023].

## 2.5. ANÁLISIS DE LOGS

Originalmente, el análisis de logs era realizado por desarrolladores con el fin de trazar el flujo de ejecución del sistema de software, identificar excepciones y potenciales errores. [Jayathilake, 2011]

Actualmente este enfoque se ha expandido a casos de uso en otros servicios en la industria

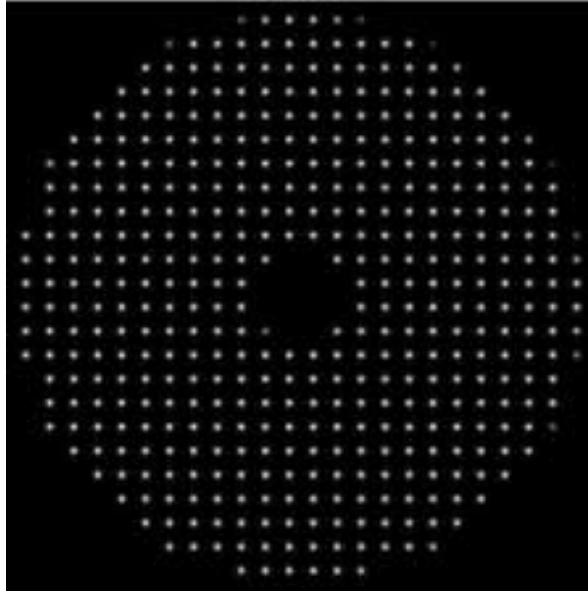


Figura 10: Imagen capturada por un sensor de frente de onda Shack-Hartmann  
Fuente: How to use Active Optics Shack-Hartmann data to measure the seeing at the focal plane of a telescope [Kolb et al., 2010]

[Ma et al., 2023], debido a que, por la naturaleza de la información contenida en los logs, su análisis permite a los operadores detectar, diagnosticar e incluso predecir errores que puedan afectar la disponibilidad y el rendimiento del sistema de software [Jayathilake, 2011].

En el pasado el análisis de logs era realizado aplicando revisiones visuales y reglas construidas manualmente. Sin embargo, la complejidad de los sistemas de software actuales ha llevado a la complejización de sus respectivos logs, por lo que ya no es posible depender solamente de los métodos anteriormente mencionados. [Ma et al., 2023]

Por esto, en los últimos años se ha desarrollado ampliamente el área del análisis automatizado de logs, mejorando su eficiencia y exactitud mediante la aplicación de tecnologías distribuidas y técnicas de machine learning. [Ma et al., 2023]

## 2.6. ESTRUCTURACIÓN DE LOGS

Los logs son considerados como "Semi Estructurados", lo cuál alude al hecho de que los logs, si bien son almacenados con un formato, este no es compartido por todas las líneas. Sumado a esto, los logs generalmente poseen una composición demasiado compleja como para ser interpretada de forma directa y manual: sin el acceso de conocimiento profesional, es difícil seleccionar de forma manual las reglas apropiadas para la comprensión de los logs. Por último, el gran tamaño del archivo de log promedio también se vuelve un problema para el análisis manual de los logs [Ma et al., 2023].

Debido a esto, durante los últimos años, se han desarrollado herramientas, procedimientos y frameworks para el análisis automático de logs, y una parte considerable de los esfuerzos realizados se enfocan en la “Estructuración” de los logs.

Actualmente, el workflow de análisis automático de logs se divide en 2 etapas centrales [Ma *et al.*, 2023]:

- Log Parsing: Se toman los registros semi-estructurados de log y se generan plantillas a partir de estos. Una plantilla es una sentencia estructurada que se repite entre varios logs, dividiéndose en los tokens estáticos y valores dinámicos. [Ma *et al.*, 2023]
- Feature Extraction: Se aplican las plantillas generadas sobre los logs para obtener las características, esto es, las variables dinámicas, de los mismos. [Ma *et al.*, 2023]

## 2.7. FUENTES DE DATOS

Se ha determinado la siguiente lista de fuentes de datos que son relevantes para la presente memoria:

1. Archivos de logs: El origen de dato más importante para esta memoria, considerando la problemática a resolver. Los logs son propios de cada telescopio UT que los produzca, siendo disponibilizados en archivos donde son agrupados en ciclos de 24 horas UTC [ESO, 1998].
2. Imágenes en formato fits: Las imágenes tomadas por el telescopio usan el formato fits, el cuál es el estándar en ciencias y astronomía. Este formato se divide en dos secciones: el Header y la imagen, donde el Header contiene la metadata que describe a la misma imagen que lo aloja, como por ejemplo su número de ID, su tiempo de exposición, las fechas de inicio y término de grabado, entre otros [NASA, 2025]. Para el caso de esta memoria, es relevante ser capaz de identificar las imágenes que fueron tomadas desde el sensor de frente de onda como parte del ciclo de Óptica Activa.
3. Observaciones en formato csv: La ESO dispone de una página web cuyo fin es archivar la información sobre todas las adquisiciones validas de los diferentes telescopios. La página web puede ser vista en la Figura 11. Desde esa página se pueden obtener los datos de todas las observaciones de un día, además permite la aplicación de diferentes filtros de búsqueda. La información disponible en la página típicamente incluye el número de ID de la observación, la fecha de inicio, el tiempo de exposición, el instrumento de captura de la imagen, entre otros. Cabe destacar que esta información no incluye el UT al que pertenece cada instrumento.

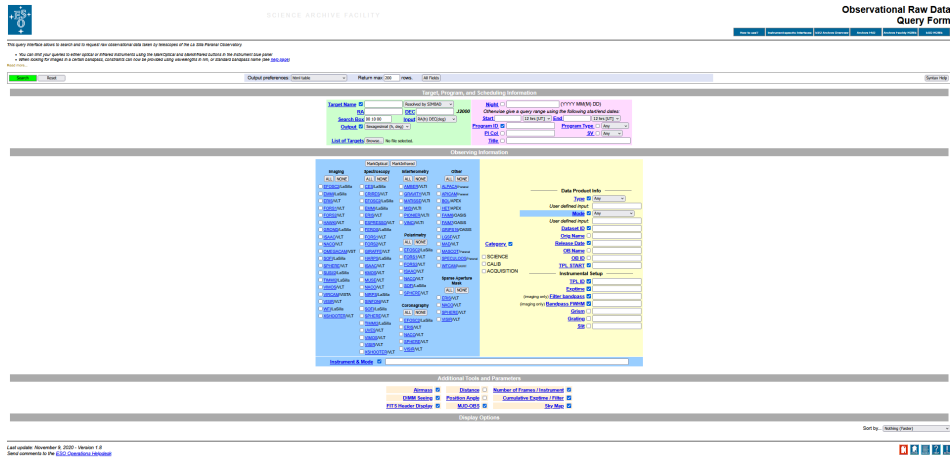


Figura 11: Página web de archivo de observación de la ESO  
 Fuente: [https://archive.eso.org/eso/eso\\_archive\\_main.html](https://archive.eso.org/eso/eso_archive_main.html), visitada el 20/10/2025.

## 2.8. PROBLEMÁTICAS RELEVANTES

### 2.8.1. Identificación de datos deseados

Los registros de log proporcionan una gran cantidad de información sobre las operaciones del VLT en una noche. Sin embargo, para efectos prácticos no es necesario ingerir toda esta información; acorde al objetivo general, solo se buscan datos relevantes para el sistema de Óptica Activa.

Debido a esto, antes de la propuesta de una solución, es necesario hacer un estudio de los registros de log y determinar aquellos que se tengan importancia para el sistema de la Óptica Activa.

Cada telescopio durante su operación nocturna, genera un log sobre su comportamiento. De esta información se deben extraer dos características centrales: datos sobre la Óptica Activa y datos sobre las características ambientales.

Los datos principales serán aquellos relacionados con el sistema de Óptica Activa: las fuerzas aplicadas de los 150 actuadores, las imágenes tomadas durante la operación y las instancias de corrección; todas dentro del rango de una observación.

#### 1. Iteración de corrección:

Se realizan múltiples iteraciones del ciclo de Óptica Activa, donde cada iteración comienza con la adquisición de una imagen y termina con una adquisición posterior de otra imagen, donde ambas imágenes son capturadas por el sensor de frente de onda.

En cada iteración, se corrigen las fuerzas aplicadas por los actuadores al espejo M1, retornando un nuevo set de fuerzas a aplicar.

Con esto en consideración, es posible tomar las iteraciones como un tipo de dato central, con el cuál se pueden relacionar las fuerzas aplicadas y las imágenes tomadas.

Cabe destacar que solo son de interés las iteraciones que concluyeron sin fallos, lo cuál se extiende tanto a las imágenes adquiridas como a las fuerzas corregidas.

## 2. Imágenes del frente de onda

Durante el proceso de Óptica Activa, las correcciones de las fuerzas aplicadas en los actuadores son realizadas en base a imágenes tomadas previamente por los sensores. Se debe considerar, además, que el proceso es iterativo, por lo que después de una corrección, se toma una nueva imagen con las nuevas fuerzas aplicadas para una nueva corrección, continuando el ciclo [ESO, 1998].

Con estas consideraciones en mente, se tiene que para cada iteración de corrección habrá una imagen que determine su inicio y otra que determine su final, y que esta última será a su vez dará inicio a la iteración siguiente.

## 3. Fuerzas aplicadas:

Durante el sistema de Óptica Activa, se ajustan exitosamente los valores de las fuerzas aplicadas de los 150 actuadores, por lo que se hace necesario conocer estos valores [ESO, 1998].

Durante cada iteración de la Óptica Activa, se determinan las diferencias de valores entre las fuerzas que se ejercen en ese momento y las fuerzas ajustadas; estas diferencias se conocen como fuerzas relativas. Luego, las fuerzas ajustadas, conocidas como fuerzas absolutas, son las que ejercen los actuadores al espejo M1 luego de la iteración [Wilson et al., 1987].

Tanto las fuerzas relativas como las fuerzas absolutas son escritas en los registros de log. Como se busca extraer un conjunto de fuerzas aplicadas, presentes en el registro de log, se decide extraer únicamente las fuerzas absolutas.

## 4. Datos adicionales

Existen otras características ambientales que son de interés, debido a que podrían estar relacionados con las deformaciones que afectan al espejo.

Debido a esto, se ha determinado la siguiente lista de valores que también se deben extraer de los logs:

- Torque aplicado al motor, tanto en altitud como en acimut, durante el tracking del telescopio [ESO, 1998].
- Posición y ángulo del UT en el momento de la observación, representados ambos con altitud y acimut [ESO, 1998].
- Velocidad y dirección del viento detectado dentro de la estructura del telescopio, medidos en metros por segundo y grados, respectivamente [ESO, 1998].
- Temperatura ambiental detectada dentro de la estructura del telescopio, medida en grados Celsius [ESO, 1998].

- Porcentaje de humedad ambiental relativa detectada dentro de la estructura del telescopio [ESO, 1998].
- Presión del aire detectada dentro de la estructura del telescopio [ESO, 1998].

## 2.8.2. Almacenamiento de los datos

Una vez identificados los datos a extraer desde las líneas de log, se hace necesario plantear como se almacenarán los mismos despues de obtenerlos.

Para la solución a desarrollar, se ha determinado la estructura de tablas vista en la Figura 12.

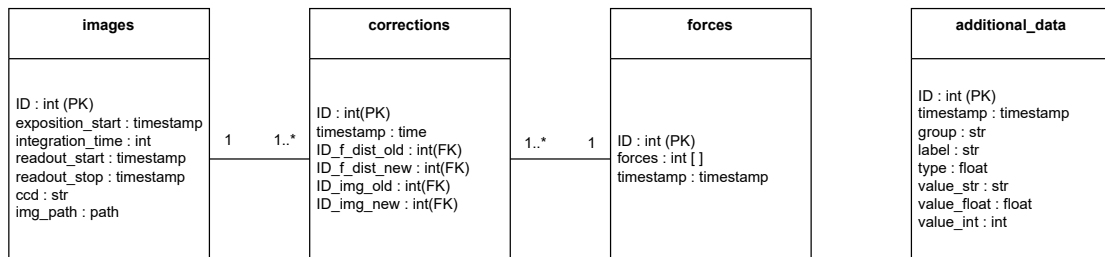


Figura 12: Diagrama general de las tablas de datos  
Fuente: Elaboración propia.

Según lo visto anteriormente, los datos principales consisten en imágenes, fuerzas aplicadas, etc. asociados a una iteración de corrección de la forma del espejo M1. Más específicamente, se tiene que generalmente cada iteración de corrección cambia los valores de estos datos.

Se puede considerar entonces agrupar los datos deseados según la iteración de corrección asociada. Por lo señalado anteriormente, cada iteración de corrección tendrá asociado un número de fuerzas aplicadas y una imagen previa a la corrección y otro número de fuerzas aplicadas y otra imagen similar luego de la misma. Entonces, cada corrección correspondería a una tabla con ambos sets de datos como atributos de la misma.

Esto puede llevar a un problema, ya que en el escenario de dos correcciones consecutivas, el set de datos post-corrección de la primera instancia y el set de datos pre-corrección de la segunda instancia serían los mismos, por lo que estos datos aparecen repetidos en ambas tablas.

Para solucionar este problema, las fuerzas aplicadas y las imágenes se almacenarán de forma separada a las iteraciones de corrección, mediante una llave foránea. Al mantener entidades distintas con múltiples relaciones por tabla, se logra normalizar la información.

De este modo, se tiene una tabla para las iteraciones de corrección, la cuál se relaciona vía llaves foráneas a otras dos tablas: una para las fuerzas aplicadas y otra para la imágenes.

Este modelo de tablas y sus respectivos campos se puede apreciar en la Figura 13.

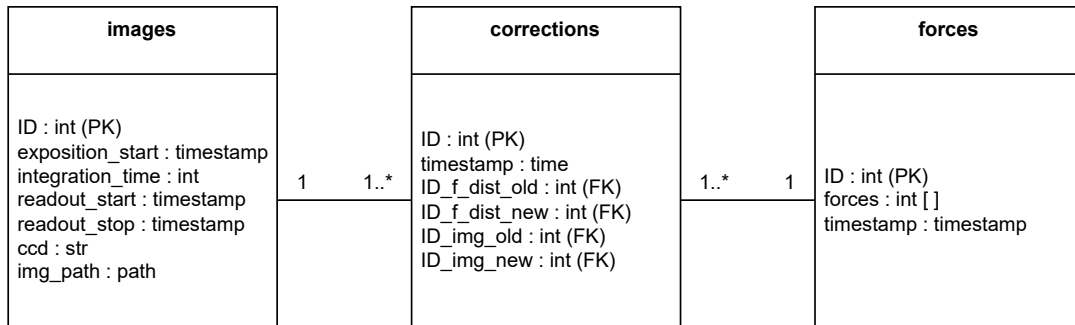


Figura 13: Diagrama del almacenamiento de datos directamente relacionados con la corrección de fuerzas.

Fuente: Elaboración propia.

Para cada tabla, se definen sus campos:

- images:
  - ID: integer, llave primaria, id del dato de imagen.
  - exposition\_start: timestamp, hora de inicio de exposición.
  - integration\_time: integer, tiempo en segundo que tardó en tomarse la imagen.
  - readout\_start: timestamp, hora de inicio de captura de la imagen.
  - readout\_stop: timestamp, hora de fin de captura de la imagen.
  - ccd: string, punto focal de captura de la imagen.
  - img\_path: string, dirección del archivo fits que almacena la imagen.
- forces:
  - ID: integer, llave primaria, id del dato de las fuerzas.
  - forces: array de int, valores numéricos de las fuerzas aplicadas.
  - timestamp: timestamp, hora de la corrección de dichas fuerzas.
- corrections:
  - ID: int, llave primaria, id del dato de la iteración de corrección.
  - timestamp: timestamp, hora del momento de la corrección.
  - ID\_f\_dist\_old: int, id de las fuerzas aplicadas pre-corrección.
  - ID\_f\_dist\_new: int, id de las fuerzas aplicadas post-corrección.
  - ID\_img\_old: int, id de la imagen pre-corrección.
  - ID\_img\_new: int, id de la imagen post-corrección.

Con respecto a los datos adicionales, debido a que estos no cambian con las correcciones, se decide almacenar estos en una tabla propia, conteniendo los atributos clave y sin necesidad de relaciones con otras tablas, como se muestran en la Figura 14.

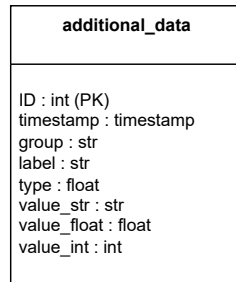


Figura 14: Diagrama de almacenamiento de datos sobre valores adicionales  
Fuente: Elaboración propia.

Para esta tabla, se definen los campos:

- additional\_data:
  - ID: int, llave primaria, id del dato de valor adicional.
  - timestamp: timestamp, hora del registro del valor en el log.
  - group: string, procedimiento que agrupa a la característica descrita por el valor (ejemplo, Óptica Activa, Análisis de Imágen, etc.)
  - label: string, característica descrita por el valor (ejemplo, torque, temperatura, presión, etc.)
  - type: string, tipo real del valor.
  - value\_str: string, valor en tipo string.
  - value\_float: float, valor en tipo float.
  - value\_int: int, valor en tipo int.

### 2.8.3. Formato de almacenamiento

Después de diseñar la estructura de los datos, se hace necesario saber qué tecnologías usar para llevarla a cabo.

El modelo relacional de la Figura 12 parece particularmente adecuado para implementar tecnologías como base de datos relacionales. Sin embargo, esta opción no sería óptima para el objetivo de esta memoria, debido a que es más factible que los datasets a generar sean consumidos a nivel de código por los modelos de soft-computing, por lo que subir los datos a un servidor de tablas relacionales para luego bajarlos en el mismo sistema es redundante e innecesario.

Un formato de datos que es apropiado para las observaciones previas es dataframes, una estructura de la librería Pandas para Python. Considerando lo anterior, sumado

a que el procedimiento va a manejar tamaños que no son tan extensos, y el uso estandarizado de Python para el manejo de datos, es que se prefiere usar Python en la última versión para la fecha de esta memoria (3.13).

#### 2.8.4. Extracción de información de las líneas de log

La extracción de datos de líneas de log se refiere al análisis de las líneas de texto presentes en los registros de log con el fin de identificar los tokens presentes y detectar los datos relevantes con el fin de extraerlos [Jayathilake, 2011].

Las arquitecturas de software grandes y complejas generan grandes cantidades de información de log mientras se ejecutan, por lo que la forma tradicional de construir manualmente expresiones regulares es demasiado costosa. Las funciones de sistemas de software complejos y actualizaciones de negocio de alta frecuencia llevan a actualizaciones más frecuentes de plantillas de línea, incrementado así en gran medida la diversidad de plantillas de línea.

Existen varias formas de realizar la extracción de datos en registros de log; escoger la adecuada no es trivial, ya que se presentan dos desafíos principales a la hora de aplicar la misma en logs [Jayathilake, 2011]:

- La problemática 1 se orienta principalmente a la extracción de datos general de registros de log, el cuál no es el caso para este trabajo; ya que se tiene claridad de los datos requeridos y de las líneas específicas que los poseen, el uso de expresiones regulares y plantillas creadas manualmente no es descartado.
- La problemática 2, si bien también se orienta hacia la extracción de datos general de registros de log y a la creación automática de plantillas de log, se puede extender su cuestionamiento a la creación de plantillas manuales; a medida que se actualicen las reglas de negocio de la Óptica Activa, las plantillas manuales creadas pueden aumentar.

El sistema de logging de Óptica Activa tiene décadas de servicio continuo, por lo que se espera que no hayan cambios de ese registro. Por el lado de los datos relevantes para el sistema de Óptica Activa, la posibilidad de que los datos requeridos cambien en el futuro es baja, debido a que los datos principales de la Óptica Activa reflejan el funcionamiento del procedimiento, como se puede concluir del análisis de la sección [2.8.1](#).

Debido a todo lo anterior, se decide optar por un modelo sencillo de extracción de datos en líneas de log, donde se itera sobre las secciones de log obtenidas tras el filtrado, y dentro de cada iteración se aplica de forma greedy cada plantilla sobre la línea seleccionada hasta que una calce. En caso de calzar, se extraen los datos acordes a la plantilla y se almacenan estos en una lista.

Para aplicar la extracción de datos en las líneas de texto, existen varios métodos y librerías diseñadas para la extracción de datos desde strings. Para este trabajo, se analizaron dos opciones posibles y se compararon según eficiencia y exactitud.

a) Extracción de datos con Expresiones Regulares

En este caso, se tienen las plantillas en forma de expresiones regulares, donde los tokens deseados están marcados como grupos dentro de las mismas plantillas.

Para este caso, se iteran en todas las líneas de log filtradas por observación, y se detectan las líneas que contienen los datos deseados usando ciertos tokens claves detectados al estudiar el formato del archivo de log.

Luego de detectar las líneas, se aplican a estas todas las plantillas hasta que se pueda extraer correctamente con una.

Para este método de extracción, las plantillas tienen el formato presentado en el Código 1:

b) Extracción de datos con Template Text Parser

Para este caso, se usa la librería externa Template Text Parser, o TTP (ver Diccionario 2 del Anexo), la cuál, de forma similar al caso anterior, permite escribir los tokens deseados en una plantilla, y luego durante la extracción solo retorna los datos llamados anteriormente.

Como es una librería diseñada para estos fines exclusivos, se procede solo a ejecutar un método de la misma para extraer datos.

Para este método de extracción, las plantillas tienen el formato presentadas en el Código 2:

Para ambos casos, los datos extraídos quedan en formato json, donde la llave corresponde al nombre de la característica (definida desde la plantilla con TTP y definida desde el código con Regex) y el valor corresponde al dato extraído.

Independiente del origen del dato, todos poseen campos de nombre 'group' y la mayoría posee también un campo 'label'. Se usará el campo 'group' para definir el tipo del dato (distribución de fuerza, imagen, etc.) y el campo 'label' para identificar subdivisiones dentro de cada tipo de dato cuando sea conveniente.

Estos campos se tornan particularmente útiles, cuando se considera que la información para un dato puede estar repartida en varias líneas, como es por ejemplo el caso de las fuerzas aplicadas, donde las fuerzas de los 150 actuadores están siempre repartidos en 6 líneas de 25 actuadores, y que a su vez todas esas líneas están separadas de la ID de la distribución, que se encuentra en su línea propia. Este formato se puede revisar en el Código 3.

De esta forma, mediante el uso de estos campos, se puede dejar registro de la información de un mismo dato para su posterior unión.

Para el algoritmo de la solución, se usará el método de extracción con Regex, ya que es más rápido en la extracción de los datos que el método con TTP, como se puede apreciar en el análisis y comparación presente en la sección 4.4.2.

```
1
2 (([0-2][0-9]{3}-[0-1][0-9]-[0-3][0-9]) ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)
↳ lt[1-4]m1m3 mlas mlasSetGlb[Abs|Rel]+ [0-9.-]+ [0-9.-]+ AS Forces #[0-9.-]+
↳ \ (abs\): ([0-9.\-\\s]+)
3
4 (([0-2][0-9]{3}-[0-1][0-9]-[0-3][0-9]) ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)
↳ lt[1-4]m1m3 mlas mlasSetGlb[Abs|Rel]+ [0-9.-]+ [0-9.-]+ AS Executed cmd
↳ #[0-9.-]+): SUCCESS)
5
6 (([0-2][0-9]{3}-[0-1][0-9]-[0-3][0-9]) ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)
↳ lt[1-4]m1m3 mlas mlasSetGlb[Abs|Rel]+ [0-9.-]+ [0-9.-]+ AS Received cmd
↳ #[0-9.-]+): [A-Z]+)
7
8 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> DET EXP NO = ([0-9]+) \ /
↳ Unique exposure ID number \([lt[1-4][a-z]{2}[a-c])\})
9
10 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL ACTO (INTTIME) =
↳ ([0-9]+))
11
12 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (ALT) (TORQUE) =
↳ ([0-9.-]+))
13
14 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (AZ) (TORQUE) =
↳ ([0-9.-]+))
15
16 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (ALT) ( )= ([0-9.-]+))
17
18 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (AZ) ( )= ([0-9.-]+))
19
20 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (IA) (ABER) ([0-9.-]+) =
↳ ([0-9.-]+))
21
22 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (IA) (ABERANG) ([0-9.-]+)
↳ = ([0-9.-]+))
23
24 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (ACTO) (ALTPOS) =
↳ ([0-9.-]+))
25
26 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (ACTO) (AZPOS) =
↳ ([0-9.-]+))
27
28 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (IA) (ABERAVE) ([0-9.-]+)
↳ = ([0-9.-]+))
29
30 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (IA)
↳ (ABERAVEANG) ([0-9.-]+) = ([0-9.-]+))
31
32 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (IA)
↳ (ABERFILMOD) ([0-9.-]+) = ([0-9.-]+))
33
34 (wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9](.[0-9]+)*)> TEL (IA)
↳ (ABERFILANG) ([0-9.-]+) = ([0-9.-]+))
35
```

### Código 1: Plantillas Regex

```

1
2 {{ date }} {{ hour }} lt[1-4]mlm3 mlas mlasSetGlbAbs [0-9.-]+ [0-9.-]+ AS Forces
↳ #([0-9.-]+) \ (abs\): {{ data }}
3
4 {{ date }} {{ hour }} lt[1-4]mlm3 mlas mlasSetGlbAbs [0-9.-]+ [0-9.-]+ AS Executed cmd
↳ #([0-9.-]+): SUCCESS
5
6 {{ date }} {{ hour }} lt[1-4]mlm3 mlas mlasSetGlbAbs [0-9.-]+ [0-9.-]+ AS Received cmd
↳ #([0-9.-]+): [A-Z]+
7
8 wt[1-4]tcs {{ hour }}> DET EXP NO = ([0-9]+) \ / Unique exposure ID number
↳ \[(lt[1-4][a-z]{2}[a-c])\]
9
10 wt[1-4]tcs {{ hour }}> TEL ACTO (INTTIME) = ([0-9]+)
11
12 wt[1-4]tcs {{ hour }}> TEL {{ group }} >{{ label }} = {{ data }}
13
14 wt[1-4]tcs {{ hour }}> TEL {{ group }} {{ label }} = {{ data }}
15
16 wt[1-4]tcs {{ hour }}> TEL {{ group }}{{ label }}= {{ data }}
17
18 wt[1-4]tcs {{ hour }}> TEL {{ group }}{{ label }}= {{ data }}
19
20 wt[1-4]tcs {{ hour }}> TEL {{ group }} {{ label }} = {{ data }}
21
    
```

### Código 2: Plantillas TTP

```

1
2 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Executed cmd #807: SUCCESS
3
4 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Forces #1 (abs): 42 45 49 50 39 14 -20
↳ -33 -10 21 47 57 53 -38 -13 25 47 43 9 18 -20 36 46
↳ 11 25
5
6 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Forces #2 (abs): -58 -53 -47 -31 -6 19 35
↳ 40 26 1 -22 -38 -42 31 16 -14 -37 -8 -6 9 52 29 -14 9
↳ 9
7
8 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Forces #3 (abs): -66 -75 -74 -55 -26 -2 3
↳ 38 28 23 31 46 57 21 2 0. 21 45 -32 -37 -27 -71 -64 -18
↳ -43
9
10 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Forces #4 (abs): 39 53 43 14 -11 -19 -8
↳ -24 3 24 38 49 57 -23 15 43 53 9 38 37 6 23 22
↳ -7 29
11
12 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Forces #5 (abs): -83 -58 -38 -28 -24 -14 10
↳ 80 67 36 -4 -35 -41 73 64 24 -25 -45 -28 -22 -5 -44 -83 14
↳ -12
13
14 Sep 21 23:30:55 wt1tcs logManager[318044]: 2024-09-21 23:30:51.970000 lt1mlm3 mlas
↳ mlasSetGlbRel 33 110 AS Forces #6 (abs): -99 -99 -89 -78 -72 -66 -49
↳ 80 73 62 51 44 45 58 36 22 26 -22 -33 -80 -62 -88 -114 14
↳ -13
15
    
```

### Código 3: Ejemplos de fuerzas aplicadas

## CAPÍTULO 3

### PROPUESTA DE SOLUCIÓN

Se implementará como solución un procedimiento ETL, escrito en Python 3.13, para extraer y transformar los logs, y luego cargarlos en datasets estructurados.

#### 3.1. Descripción general del proceso

El proceso de la solución puede ser revisado en la Figura 15. Igualmente, el mismo puede resumirse de tres etapas principales, cada una dividida en distintas sub-etapas:

1. Filtrado de líneas de logs: En esta etapa, a los logs de 24 horas de operación se les remueven elementos repetidos dentro de las líneas, para luego rescatar las líneas que se encuentren dentro de los periodos de observación ocurridos durante la noche.
2. Extracción de datos: En esta etapa, se extrae la información deseada de las líneas de los logs, usando una plantilla de líneas prefabricada, que especifica la ubicación de los datos a extraer dentro de las líneas a leer.
3. Guardado de datos: En esta etapa, se construyen los datasets para cada una de las tablas descritas en la sección 2.8.2, y se cargan con la información previamente extraída. Luego de esto, se valida que la información ingestada sea consistente y se construyen las relaciones correspondientes entre los datasets.

Los siguientes capítulos se encargan de describir a mayor detalle las etapas del proceso.

#### 3.2. Filtrado de líneas de log

Los logs poseen cientos de miles de líneas, por lo que un proceso de filtrado aplicado directamente sobre estos sería costoso en tiempo y poder de cómputo.

Por ende, es prudente procesar los logs de forma previa, aplicando transformaciones que retornen un conjunto de líneas más apto para el filtrado.

##### 3.2.1. Remoción de valores redundantes

Tras revisar los logs, como el ejemplo presente en el Anexo 1, se puede notar que existe un patrón recurrente en todas las líneas.

Estos valores iniciales, que incluyen la primera mención a la hora, nombre del UT y del log-Manager, aportan información que es redundante dentro de la misma línea, como la hora y el nombre del UT, o dentro del archivo, como la fecha. De este modo, se puede concluir que dichos valores iniciales no son necesarios para el objetivo de esta memoria, debido a que la información que aportan puede ser extraída desde otras fuentes.

Por lo mismo, con tal de facilitar la extracción de datos, se procede a eliminar estos valores.

### 3.2.2. Cruce con archivo de observaciones

Parte del objetivo de esta memoria es garantizar que la información extraída sea fidedigna y correcta. Durante la operación de un UT, pueden ocurrir problemas técnicos o de observación, por lo que es necesario corroborar que los logs pertenezcan a rangos de tiempo donde el telescopio funcionó de forma correcta.

Además, no todas las observaciones son usadas en el proceso de Óptica Activa; las observaciones de calibración no son consideradas para este proceso, solo las de adquisición y científicas.

Los logs no muestran de qué tipo es cada observación ni si en el rango de tiempo de la observación el UT funcionó de forma correcta, por ende es necesario extraer esta información de un documento externo y cruzarla con las líneas de observaciones ya extraídas, con tal de obtener las líneas de log correspondientes a un rango de tiempo de correcto funcionamiento del UT.

#### 1. ESO Observational Raw Data Query

Como se mencionó en la sección [2.7](#), la ESO posee una página donde se puede acceder de forma libre a información sobre las observaciones realizadas por el VLT. Dentro de esta información, se puede extraer las observaciones realizadas, la hora en que se realizaron y el tipo de observación, entre otros atributos.

Es posible realizar una búsqueda con los campos deseados a la página, y esta luego retorna un archivo csv con las observaciones de la noche descrita para todos los instrumentos disponibles durante la misma.

#### 2. Segmentación de los periodos de observación

Se asume que durante una observación válida el UT funcionó de forma correcta y, por consiguiente, se asume que todas las líneas de log dentro de ese tiempo de observación corresponden a una operación correcta del UT.

Bajo esta hipótesis, se procede a cargar el documento de observaciones y usar el mismo para delimitar los horarios de inicio y término de cada periodo de observación válido, para luego usar estos límites en el filtrado de las observaciones obtenidas desde los logs.

Inicialmente, se pueden crear 'ventanas de tiempo' usando la hora de inicio de cada observación junto con el tiempo de exposición de la misma. Es importante que, al momento de crear estas ventanas, se mantenga la distinción de bajo que instrumento se produce la observación en cuestión.

Luego de este paso, es natural intuir que algunas de estas ventanas para un mismo instrumento se puedan sobrelapar, como con observaciones que puedan iniciar antes que termine una anterior, lo cuál incrementa las ventanas de tiempo a analizar. Por ende, tras crear las ventanas de tiempo, se debe cerciorar que aquellas ventanas que se sobrelapen en horario para un mismo instrumento sean unidas.

Se puede asumir que el UT habrá funcionado correctamente por un intervalo de tiempo anterior y posterior a la ventana de tiempo usada por referencia [ESO, 1998]. Además, se realiza la suposición que entre el fin de una ventana de tiempo y el comienzo del siguiente, la operación del telescopio continua por un lapso de tiempo.

Debido al párrafo anterior, se considera añadir tres márgenes paramétricos de tiempo:

- Margen anterior: Cantidad de segundos previo a una ventana de tiempo.
- Margen posterior: Cantidad de segundos posterior a una ventana de tiempo.
- Umbral intermedio: Cantidad de segundos entre ventanas de tiempo para un mismo instrumento (con márgenes incluidos) donde, si la distancia horaria es menor al umbral, se procede a unir dichos bloques.

Debido a lo anterior, se ha determinado que estos márgenes sean paramétricos, con tal de ajustar de forma apropiada las ventanas de tiempo.

### 3. Clasificación según cruce

Con las ventanas de tiempo formadas, solo resta filtrar las líneas de log en base de las mismas.

Para este proceso, solo basta con incluir todas las líneas de log cuyo horario se encuentre dentro de las ventanas de tiempo.

## 3.3. Extracción de datos

El método tradicional para extraer datos de líneas de log es usar plantillas en las mismas. El objetivo es extraer los datos deseados de las líneas de texto, y disponerlos en los formatos adecuados. Este método es el estándar, por lo mismo se decide usarlo para las líneas previamente procesadas.

### 3.3.1. Plantillas de líneas

Las plantillas corresponden a una secuencia de caracteres que marcan la posición de los valores deseados dentro de una línea de texto. Debido a que se conoce la cantidad específica de datos requeridos, se decide crear las plantillas manualmente.

Estas plantillas se disponen finalmente al procedimiento ELT, el cuál extraerá los datos de las líneas de log con las plantillas de forma automática.

Este procedimiento implica iterar por cada línea del log, aplicando cada plantilla hasta que una de estas logre extraer datos de la línea de log exitosamente. Este proceso es de complejidad polinomial, al tratarse de un ciclo dentro de otro ciclo. El número del ciclo interno correspondiente al número de plantillas, y por lo mismo no puede reducirse. El número del ciclo externo corresponde al número de líneas de log, el cuál si puede reducirse aplicando el filtrado de la sección [3.2](#). Con esto dicho, considerando que el número de líneas de log sin filtrar es del orden de cientos de miles, este número no es de un tamaño tan grande en el contexto de la big data, por lo que la complejidad polinomial es aceptable.

El formato final de las plantillas varía según el método de extracción de datos usado, por lo que se muestran imágenes de las mismas en las subsecciones correspondientes.

Sin embargo, el análisis realizado en la sección [2.8.2](#) lleva a la conclusión de usar 16 plantillas para obtener los datos deseados; una plantilla para cada línea a analizar.

### 3.3.2. Construcción de los dataframes

En la sección [3.2.1](#), se definieron las tablas a desarrollar con sus relaciones y atributos correspondientes, por ende, los dataframes a crear deben seguir la misma estructura.

Así mismo, se crearan 4 dataframes:

- Uno para las instancias de corrección, donde para cada instancia se guardan las referencias a las imágenes y fuerzas aplicadas anteriores y posteriores.
- Uno para las fuerzas aplicadas, donde para cada distribución se guardan los pesos de los 150 actuadores y el timestamp del registro en el log.
- Uno para las imágenes tomadas, donde para cada imagen se guardan los tiempos de inicio y término de lectura, el tiempo de inicio de exposición, el tiempo de integración, el sensor que leyo la imagen y la dirección local al archivo .fits correspondiente.
- Uno para los datos adicionales, donde para cada dato se guarda el grupo (relacionado con la cantidad con la que trata, ya sea altitud, acimut, imagen, etc.), la etiqueta (relacionado con la cualidad específica que maneja esa cantidad, sean aberraciones,

torque, etc.), el dato numérico o en texto. Para lo último, se decide guardar el dato en formato entero, flotante y texto, con fines de respaldo, sumado a un campo de tipo de dato que describe cuál es el formato correcto.

De esta forma, los dataframes finales poseen los mismos esquemas que los presentados en el diagrama de la sección [3.2.1](#).

### 3.3.3. Carga a los dataframes

Debido a que los datos extraídos se encuentran como diccionarios de Python, corresponde iterar por todos los datos, y migrar los datos desde el formato diccionario a dataframe.

En la sección [2.8.4](#), se mencionaron las llaves 'group' y 'label' para diferenciar los tipos de datos. Estas llaves se usan en este proceso, con tal de poder identificar el tipo de dato, y según aquello, migrar los datos de forma acorde.

Como ejemplo, se conoce que los datos de fuerzas aplicadas poseen un campo con las fuerzas de los actuadores, y que ningún otro tipo de dato posee tal campo. Entonces, al tener registro del tipo de dato en la llave 'group', se puede validar que datos poseen ese campo y extraerlos sin temor a falla.

Así mismo, se puede concatenar la información de ciertos datos que se encontraban en múltiples líneas. Tras revisar el campo 'group', se revisa que no existan datos en el dataframe con información faltante, y se revisa el campo 'label' para verificar que tipo de información específica es la que falta.

### 3.3.4. Validación de los dataframes

Eventualmente durante el proceso de ingesta a los dataframes se producirán errores, principalmente por errores en los datos desde el origen o por las convenciones tomadas para el desarrollo de los filtros.

Una fuente de errores común es el método de filtrado usando las ventanas de tiempo de observación, el cual puede excluir líneas de log que definan fuerzas aplicadas o una imagen relacionada a una iteración de corrección ya está incluida en las ventanas de tiempo.

Debido a ello, luego de la ingesta, se deben validar los datos dentro de los dataframes con tal de asegurar que dicha información es correcta y de utilidad.

Para cada tipo de dato, el proceso de validación es distinto:

- Para las instancias de corrección, se deben remover las instancias inválidas, por ejemplo por ID nulo, y también se remueven aquellas instancias que no poseen fuerzas aplicadas posterior a la corrección.
- Para las fuerzas aplicadas, se deben remover las instancias inválidas, por ejemplo por ID nulo, y también se remueven aquellas fuerzas que son anteceditas por imágenes que ya anteceden a otras fuerzas, y lo mismo aplica para fuerzas seguidas por imágenes que ya suceden a otras fuerzas.
- Para las imágenes, solo se remueven las instancias inválidas, por ejemplo por ID nulo.

El dataframe de `additional_data` no es validado, debido a ser datos de carácter miscelaneo y secundario.

### 3.3.5. Relación de los dataframes

Finalmente, una vez se valida la información presente en los dataframes, se procede a crear las relaciones mencionadas en la sección [2.8.2](#).

Para esto, simplemente se hacen calzar los timestamp de los datos de imagen y de fuerzas aplicadas, y luego se entregan sus IDs a la instancia de corrección correspondiente para incluirlas en sus campos de llaves foráneas.

## 3.4. Guardado de los datos

Una vez extraídos los datos, se deben crear y validar los dataframes donde se almacenarán estos mismos.

Este almacenamiento sigue el modelo de la Figura [12](#) y los lineamientos discutidos en la sección [2.8.2](#).

## CAPÍTULO 4

### VALIDACIÓN DE LA SOLUCIÓN

#### 4.1. Algoritmo

El algoritmo final sigue el flujo del Capítulo 3, cuya representación visual se puede encontrar en el Figura 15, cuyas partes se describen en esta sección:

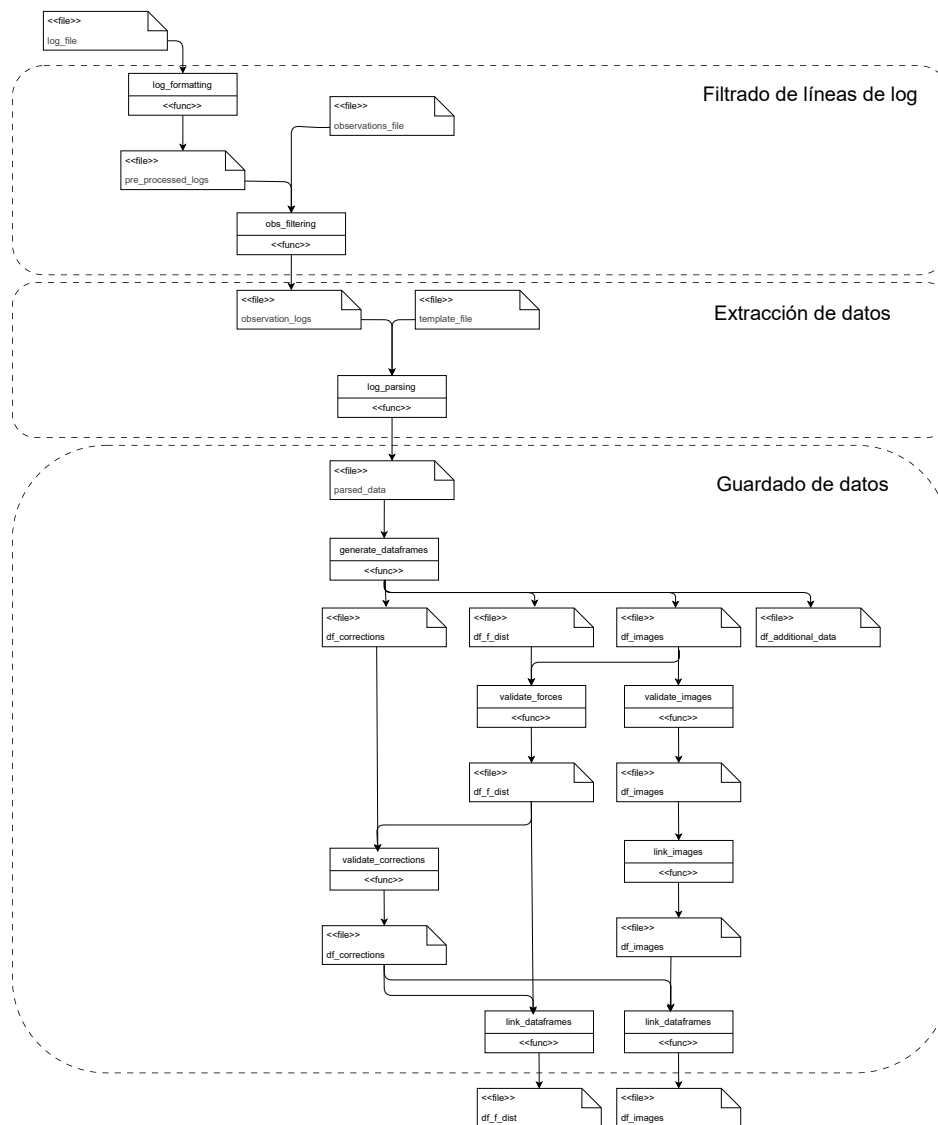


Figura 15: Diagrama de flujo del proceso de solución  
Fuente: Elaboración propia

#### 4.1.1. `log_formatting()`

Cumple con el proceso planteado en la sección [3.2.1](#), donde se remueven los tokens de fecha y hora iniciales, debido a su redundancia a nivel de línea, además de corregir otros detalles de formato con tal de facilitar el posterior análisis.

Se puede ver un ejemplo de los resultados de `log_formatting()` con los Códigos [4](#) y [5](#).

```
1 Aug 3 21:57:27 wtltcs logManager[47272]: wtltcs 21:57:27> TEL M1 X = -247.50940 / M1 X
  ↳ passive positions [wtltcs]
2 Aug 3 21:57:27 wtltcs logManager[47272]: wtltcs 21:57:27> TEL M1 Y = 4.30516 / M1 Y
  ↳ passive positions [wtltcs]
3 Aug 3 21:57:27 wtltcs logManager[47272]: wtltcs 21:57:27> TEL M1 Z = -0.38670 / M1 Z
  ↳ passive positions [wtltcs]
4 Aug 3 21:57:27 wtltcs logManager[47272]: wtltcs 21:57:27> TEL M1 A = 0.13895 / M1 A
  ↳ passive positions [wtltcs]
5 Aug 3 21:57:27 wtltcs logManager[47272]: wtltcs 21:57:27> TEL M1 B = 0.23212 / M1 B
  ↳ passive positions [wtltcs]
6 Aug 3 21:57:27 wtltcs logManager[47272]: wtltcs 21:57:27> TEL M1 C = 0.55083 / M1 C
  ↳ passive positions [wtltcs]
```

#### Código 4: Ejemplos de líneas previa a `log_formatting()`

```
1 Aug 3 wtltcs 21:57:27> TEL M1 X = -247.50940 / M1 X passive positions [wtltcs]
2 Aug 3 wtltcs 21:57:27> TEL M1 Y = 4.30516 / M1 Y passive positions [wtltcs]
3 Aug 3 wtltcs 21:57:27> TEL M1 Z = -0.38670 / M1 Z passive positions [wtltcs]
4 Aug 3 wtltcs 21:57:27> TEL M1 A = 0.13895 / M1 A passive positions [wtltcs]
5 Aug 3 wtltcs 21:57:27> TEL M1 B = 0.23212 / M1 B passive positions [wtltcs]
6 Aug 3 wtltcs 21:57:27> TEL M1 C = 0.55083 / M1 C passive positions [wtltcs]
```

#### Código 5: Ejemplos de líneas posteriores a `log_formatting()`

#### 4.1.2. `obs_filtering()`

Cumple con el proceso descrito en la sección [3.2.2](#), donde se obtiene un archivo csv con las observaciones de la noche seleccionada, para luego formar ventanas de tiempo basadas en los tiempos de las observaciones, estas ventanas luego son modificadas según los parámetros de margen anterior, margen posterior y umbral intermedio, para finalmente filtrar las líneas de log con tal de quedar solo con aquellas líneas cuya fecha se encuentre dentro de alguna de esas ventanas de tiempo.

#### 4.1.3. `log_parsing()`

Cumple con el proceso descrito en la sección [3.3.1](#), donde en base a un archivo plano de plantillas, se itera sobre las líneas de log, aplicando cada plantilla línea por línea hasta encontrar alguna que calze y guardar la data extraída en un json final. Este proceso tiene una complejidad polinomial, equivalente a  $O(n^2)$ .

En la sección [2.8.4](#) se describen los dos métodos posibles para realizar la extracción: mediante Expresiones Regulares y mediante la librería TTP. En la sección [4.4.2](#) se analizan y comparan los resultados de ambos métodos, concluyendo que el método con Regex es el más apropiado para este algoritmo.

#### 4.1.4. `generate_dataframes()`

Cumple con el proceso de la sección [3.3.3](#), donde se establece el traslado de los datos de los diccionarios generados en la función anterior a un conjunto de dataframes, usando los campos 'group' y 'label' como etiquetas para reconocer a qué dataframe destinar los datos.

De esta función se generan 4 dataframes, como se señalo en la sección [3.3.2](#):

- `df_corrections`: Para las instancias de corrección.
- `df_images`: Para las imágenes capturadas.
- `df_f_dist`: Para las distribuciones de fuerza de los actuadores.
- `df_additional_data`: Para los datos adicionales relacionados con la Óptica Activa.

#### 4.1.5. `validate_forces()`

Cumple con el segundo ítem de la sección [3.3.4](#), donde se especifica la validación de los datos dentro de dataframe de fuerzas aplicadas generado en la función anterior. Esto se logra con las siguientes actividades:

- Eliminar los datos con ID nulo.
- Remover los datos que sean antecidos por imágenes que ya anteceden a otros datos.
- Remover los datos que sean seguidos por imágenes que ya seguidos a otros datos.

#### 4.1.6. `validate_images()`

Cumple con el tercer ítem de la sección [3.3.4](#), donde se especifica la validación de los datos dentro de dataframe de imágenes generados en la función anterior. Solamente se eliminan aquellos datos con `ID_image` nulo.

#### 4.1.7. `link_images()`

Cumple con el requisito de la sección [3.3.5](#), sobre `df_images` manteniendo la dirección local del archivo `.fits` correspondiente como un atributo de los datos del dataframe.

En esta función, se ingresa a la carpeta donde estan almacenados los archivos `.fits`, se extrae el número de ID y el tiempo de exposición del header de cada `.fits` y se usan estos para buscar el dato de `df_images` que le sea correspondiente. En caso de encontrarlo, se guarda su dirección local en el atributo correspondiente del dato.

#### 4.1.8. `validate_corrections()`

Cumple con el primer item de la sección [3.3.4](#), donde se especifica la validación de los datos dentro del dataframe de iteraciones de corrección generado en la función anterior. Esto se logra con las siguientes actividades:

- Eliminar los datos con ID nulo.
- Remover los datos sin distribución de fuerza posterior a la corrección.

#### 4.1.9. `link_dataframes()`

Cumple con el proceso descrito en la sección [3.3.5](#), en la que se indica como relacionar los datos del dataframe de fuerzas aplicadas y del dataframe de imágenes con los datos de dataframe de iteraciones de corrección mediante los timestamp de estos datos, comparando estos para luego copiar la ID del dato de la cantidad deseada en un campo designado dentro del dato correspondient en el dataframe de iteraciones de corrección.

Para esta función, como la construcción de relaciones entre el dataframe de imágenes y el dataframe de fuerzas aplicadas con el dataframe de iteraciones de corrección es identico para ambos dataframes, se pudo generalizar y parametrizar este proceso en una misma función, que tiene como parámetros al dataframe de la cualidad deseada (fuerzas aplicadas o imágenes), el nombre de la cualidad deseada y el campo de tiempo dentro del dataframe elegido.

## 4.2. Ejecución

El comando y los argumentos para la ejecución del código desarrollado se describen en el Código [6](#). Para el apropiado funcionamiento del algoritmo, el archivo de log debe estar previamente cargado en el sistema local.

Además se desarrolló un sistema de flags opcionales que otorga la posibilidad de habilitar o deshabilitar las funciones descritas en la sección 4.1. Estas se pueden ver en el Código 6.

```
1
2 usage: nmle.py [-h] [-t TEMPLATEFILE]
3 [-p] [-o] [-i] [-l] [-c] [-s SAVE] [-r] date ut
4
5 positional arguments:
6   date                Date to analyzed
7   ut                  Number of the UT telescope to be analyzed
8
9 options:
10  -h, --help          show this help message and exit
11  -t TEMPLATEFILE,
12  --templatefile TEMPLATEFILE
13                      Path to the text file where the templates
14                      are stored (if no name is written,
15                      ./poc_templates.txt
16                      will be set by default)
17  -p, --preprocess    Skips logs pre-processing stage: removal
18                      of log headers and grouping of log
19                      lines in sections, based on when
20                      AG GUIDE has stopped activity
21  -o, --obsprocess    Skips observation files processing stage:
22                      filtering of log lines based on time
23                      intervals where a valid observation has
24                      been performed
25  -i, --linkimages    Skips linking of image log data with
26                      fits files: write of the fits files'
27                      path respective to each image row
28                      in the images' dataframe
29  -l,
30  --linkdataframes    Skips linking of correction
31                      dataframe with related dataframes:
32                      write of the ids of the force
33                      distribution and image respective
34                      to each row in the correction
35                      dataframe
36  -c, --console       Prints resulting dataframes on console
37  -s SAVE,
38  --save SAVE         Stores resulting dataframes in
39                      csv files
40  -r, --report        Prints a report detailing the processing
41                      of log lines
```

### Código 6: Ejemplos de fuerzas aplicadas

Existen 2 posibles modos de retorno de los dataframes generados: por consola o por csv. Estos se habilitan con las flags -c y -s, respectivamente.

## 4.3. Pruebas

Para corroborar el correcto funcionamiento del algoritmo, es necesario analizar los resultados del mismo, junto con revisar el rendimiento y exactitud de la ejecución.

### 4.3.1. Metodología

Para poder analizar el desempeño y resultados del algoritmo, primero se realiza un conteo manual de las instancias de corrección, las distribuciones de fuerza y las imágenes que se

presentan en una muestra de archivos de log. Luego, se ejecuta el algoritmo para procesar los mismos archivos de log, y se comparan los resultados.

La muestra se compone de los siguientes archivos de log:

- L1 : Archivo de log para la observación del UT 1 durante la noche del 04/08/2025.
- L2 : Archivo de log para la observación del UT 1 durante la noche del 05/08/2025.
- L3 : Archivo de log para la observación del UT 3 durante la noche del 04/08/2025.
- L4 : Archivo de log para la observación del UT 3 durante la noche del 05/08/2025.
- L5 : Archivo de log para la observación del UT 4 durante la noche del 04/08/2025.
- L6 : Archivo de log para la observación del UT 4 durante la noche del 05/08/2025.

Una vez realizadas las ejecuciones, se procede a analizar las estadísticas de los resultados, más específicamente el número de instancias de corrección, distribución de fuerzas e imágenes obtenidas, además del tiempo de ejecución.

Este proceso de revisión se realizará primero en ámbitos generales del algoritmo, y luego se revisará para cada etapa de forma más específica.

Además del desempeño, es fundamental evaluar la eficiencia del algoritmo.

Para esto, se tomarán extractos de una hora de L2, L4 y L6, abarcando desde las 04:00:00 hasta las 05:00:00. Luego se analizarán estos extractos de forma manual y se extraerán valores de iteraciones de corrección, fuerzas aplicadas e imágenes tomadas. Esta extracción manual se realizará aplicando las validaciones expresadas en la Sección 3.3.4, pero no comparará las líneas con las ventanas de observación. Luego, se ejecutará el algoritmo sobre estos tres extractos. Finalmente, se compararán los valores extraídos de forma manual con los valores extraídos mediante el algoritmo, obteniendo de este modo una métrica porcentual de la eficiencia del algoritmo.

## 4.4. Resultados

Las líneas encontradas de forma manual, que contienen valores de iteraciones de corrección, fuerzas aplicadas e imágenes, en cada archivo de log, se reflejan en la Tabla 1.

Cabe destacar que las líneas con valores de fuerza van en grupos de 6, donde cada línea contiene el registro de 25 fuerzas, cómo se aprecia en el Código 3.

Log	Líneas	Correcciones	Fuerzas	Imágenes
L1	427195	626	3516	851
L2	384729	598	3372	802
L3	409830	537	1662	908
L4	371450	546	1572	741
L5	444126	491	1722	786
L6	422312	495	2148	713

Tabla 1: Líneas encontradas de forma manual

Luego, las líneas que contienen los valores de las iteraciones de corrección, fuerzas aplicadas e imágenes, encontradas por el algoritmo en los mismos archivos se encuentran en la Tabla 2.

Log	Líneas	Correcciones	Fuerzas	Imágenes	Tiempo total [s]	Tiempo por línea [ms]
L1	427195	4	5	11	61.92	0.15
L2	384729	31	33	54	89.33	0.23
L3	409830	4	5	8	69.49	0.17
L4	371450	19	19	46	79.14	0.21
L5	444126	3	3	4	59.18	0.13
L6	422312	18	18	31	103.25	0.24

Tabla 2: Líneas encontradas por el algoritmo

Log	Ventanas	Tiempo total [s]
L1	18	512
L2	43	3028
L3	18	512
L4	43	3028
L5	18	512
L6	43	3028

Tabla 3: Ventanas de observación y tiempo total de observación

Es evidente la amplia diferencia entre ambas cantidades, sin embargo, existen motivos para explicarlo.

Primero, se debe mencionar que la extracción manual se realizó mediante un simple conteo de líneas de log según ciertos tokens clave, mientras que para la extracción con el algoritmo se aplicó lógica durante las validaciones y filtros que permiten un conteo con información más certera, aunque reducida en tamaño.

Ejemplos de reglas incumplidas en los logs:

- Existen varias imágenes que se pueden tomar consecutivamente en lapsos de unos pocos segundos.
- Existen distribuciones de fuerza que resultan fallidas.

Estos casos son detectados por el algoritmo, evitando la extracción de estos datos inválidos.

El otro motivo de porque la diferencia en el número de cantidades es la filtración por observación: según se puede observar en la Tabla 3, la suma de los tiempos de las ventanas de tiempo de las observaciones es baja, lo cuál deja fuera a una buena parte de las líneas presentes en el archivo de log. Además, se puede notar que la cantidad de ventanas y el tiempo total no dependen del número del UT, sino que depende exclusivamente de la fecha de observación, lo cuál puede causar inconsistencias en la captura de datos, ya que cada UT dispone de instrumentos específicos, como se menciona en la sección 2.1. A su vez, en la sección 2.7 se menciona que los archivos csv de las observaciones no contienen información sobre el UT al cuál corresponde el instrumento de observación.

Ahora bien, como se menciona en la sección 3.3.2 el filtro de observaciones se realiza con tres parámetros: el margen anterior y el margen posterior para cada ventana de tiempo, y el umbral intermedio entre las ventanas de tiempo. Estos tienen sus valores asignados de forma rígida dentro del código. Por ende, la modificación de estos parámetros ampliaría el rango de búsqueda de las cantidades en el archivo de log. Actualmente, se definieron los márgenes anterior y posterior como 10 segundos cada uno, y el umbral de unidad de los bloques de observación se encuentra en 30 segundos.

Adecuar apropiadamente estos márgenes pudiera ser relevante para mejorar la captura de valores, sin embargo tampoco se pueden llegar a extremos muy altos, ya que puede comenzar a presentar ruido por el sobrelape de bloque de tiempo de las observaciones.

Para la revisión de la eficiencia, se esquematizan los datos extraídos manualmente con los datos extraídos por el algoritmo en la Tabla 4.

Log	Datos Manual	Datos Algoritmo	% eficiencia
L2	194	31	15.98 %
L4	132	19	14.39 %
L6	48	7	14.58 %
Prom			14.98 %

Tabla 4: Comparación de datos extraídos manualmente con datos extraídos mediante el algoritmo desde muestras de líneas de log entre las 04:00 hrs y las 05:00 hrs

Se puede notar que la eficiencia es baja, manteniendo a un porcentaje similar para las tres muestras analizadas. Esto se puede deber principalmente al uso de las ventanas de observación por parte del algoritmo.

#### 4.4.1. Resultados para Filtrado de líneas de log

Parte crucial de esta etapa es el cruce de los registros de log con los datos de observaciones, tal como se explicó en la sección 3.2.2. Con esto en mente, se puede examinar el rendimiento de esta etapa comparando la cantidad de líneas de log que ingresa con la cantidad que sale, para cada uno de los casos descritos al inicio de esta sección.

Los resultados de estas pruebas se exponen en la Tabla 5.

Log	Líneas entrada	Líneas salida	% extracción
L1	427195	3829	0.9%
L2	384729	23475	6.1%
L3	409830	4496	1.1%
L4	371450	22048	5.94%
L5	444126	3593	0.81%
L6	422312	27365	6.48%
Prom			3.56%

Tabla 5: Cantidad de líneas de log filtradas

Se puede notar de inmediato la influencia de los filtros en esta etapa: en todos los casos probados se extrajeron menos del 10 % de las líneas de log.

Esto trae ventajas y desventajas: por un lado, se asegura que las líneas de log que pasen a la etapa de Extracción de Datos correspondan a observaciones válidas del telescopio; por el otro lado, es posible teorizar que se están desechando líneas de log que puedan poseer información relacionada con la ejecución del sistema de Óptica Activa.

#### 4.4.2. Resultados para Extracción de los datos

Se prueban los dos métodos de extracción mencionados en la sección 2.8.4, usando la serie de parámetros iniciales señalados en el inicio, y se registran en la Tabla 6 los tiempos y cantidad de datos extraídos para cada método, además de un porcentaje de comparación calculado dividiendo el tiempo del método de Regex en el tiempo del método de TTP.

Log	Líneas del Log	Datos TTP	Tiempo TTP [s]	Datos Regex	Tiempo Regex [s]	%comparación ttp/regex
L1	427195	68	323.12	68	61.92	521.83%
L2	384729	546	1442.49	546	89.33	1614.79%
L3	409830	62	352.58	62	69.49	507.38%
L4	371450	450	1416.33	450	79.14	1789.65%
L5	444126	34	339.54	34	59.18	573.74%
L6	422312	473	1644.52	473	103.25	1592.76%

Tabla 6: Tiempos y cantidad de datos para modos de extracción

Con esta prueba, se puede notar que ambos métodos de extracción retornan la misma cantidad de datos, sin embargo TTP toma una mayor cantidad de tiempo en comparación con Regex. Más particularmente, se puede notar que para los casos con decenas de datos, el método con TTP tardaba 5 veces más que el método con Regex, mientras que en los casos con cientos de datos, el tiempo de análisis del método de TTP supera por más de 10 veces al tiempo del método de Regex.

Este es el fundamento principal detrás de la elección del método de extracción con Regex por sobre el método de extracción con TTP mostrada en la sección 2.8.4.

Una vez determinado el método de extracción, se procede a determinar la cantidad de líneas

cuyos datos fueron extraídos sobre la cantidad de líneas que ingresaron a la etapa, similar a la prueba realizada para Filtrado de Líneas de Log. Los resultados se muestran en la Tabla 7.

Esto se revisa en esta etapa porque, si bien aquí no tiene el fin de filtrar directamente, esto igualmente ocurre frecuentemente debido a la aplicación de plantillas, donde aquellas líneas de log que no calzen con ninguna plantilla serán descartadas.

Log	Líneas entrada	Líneas salida	% extracción
L1	3829	135	3.53 %
L2	23475	940	4.01 %
L3	4496	142	3.16 %
L4	22048	819	3.17 %
L5	3593	73	2.03 %
L6	27365	806	2.95 %
Prom			3.14 %

Tabla 7: Cantidad de líneas de log extraídas

Según lo reflejado en la Tabla 7, los porcentajes de extracción son incluso menores que en la etapa anterior. Sin embargo, son porcentajes esperables para esta etapa, ya que el filtrado extensivo es un sintoma del proceso de aplicación de plantillas.

Por último, se analizará la aplicación de las validaciones, una vez construidos los datasets.

Para este caso, se contarán datos de distribución de fuerzas, instancias de corrección e imágenes tomadas. Esto porque para este punto del algoritmo los datos ya han sido extraídos de las líneas, sumado a que las validaciones solo se aplican a esos tipos de datos y no a las características adicionales.

Los resultados de este análisis se presentan en la Tabla 8.

Log	Datos entrada	Datos salida	% extracción
L1	26	20	70 %
L2	146	118	80.82 %
L3	24	18	75 %
L4	116	84	72.41 %
L5	14	10	60 %
L6	100	67	67 %
Prom			70.87 %

Tabla 8: Cantidad de datos filtrados

Se puede notar que los porcentajes de extracción son más altos que los vistos en pasos anteriores. Sin embargo, el filtrado de datos realizado por las validaciones tampoco es despreciable, removiendo un promedio del 29.13 % de los datos validados.

#### 4.4.3. Resultados para Guardado de los datos

Para esta sección no es necesario realizar comparaciones, ya que esta parte del logaritmo solo se encarga de cargar los datos a los dataframes finales.

## CAPÍTULO 5

### CONCLUSIONES

En esta memoria se logró abordar la problemática planteada, con el desarrollo de un procedimiento que extrae información de los registros de log de los telescopios UTs, en relación con el sistema de Óptica Activa, y los disponga en datasets de datos.

Para lograr esto, fue necesario estudiar varios métodos de extracción de datos y estructuración de logs, cuyas bases fueron tomadas para desarrollar un procedimiento propio.

Durante una parte considerable del desarrollo, se consideraron dos posibles métodos de extracción: una usando Expresiones Regulares, y otra usando la librería Text Template Parser de Python. Si bien ambas opciones son válidas, para el caso de esta memoria se demostró que la opción de Expresiones Regulares era la más eficiente, tardando menos de 5 veces lo que tarda el método de TTP.

El algoritmo actual genera datasets con absoluta fiabilidad de que la información contenida este completa y corresponda a un momento en la que el UT respectivo funcionó con normalidad. Sin embargo, el algoritmo posee una eficiencia de alrededor del 14.98 % con respecto al número original de líneas de log, dejando una alta cantidad de líneas de log sin procesar, y el algoritmo no corrobora que todas las líneas de log descartadas no posean datos de distribución de fuerzas o de condiciones ambientales del telescopio. Por lo mismo, un punto crucial para estudio futuro es la búsqueda de nuevos métodos, o mejorar los ya existentes, con tal de ampliar el rango de captura de líneas de log para el análisis y extracción de datos, y que el algoritmo pueda confirmar que las líneas descartadas no posean datos de interés.

Otro enfoque de posible interés para estudio futuro es la generalización de este algoritmo, ya que actualmente está enfocado al sistema de Óptica Activa. Es digno de análisis si la misma arquitectura puede servir para la extracción de datos orientados a otras áreas de la operación de los UTs, como por ejemplo la detección de errores en tiempo real.

## ANEXOS

### 1. Extracto de un registro de log del UT 1.

```
1 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23>-START DET EXP / Start exposure
2   ↪ [ltliab]
3 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23> TEL M2 DECOFF = 0.0079 /
4   ↪ [arcsec] DEC correction offloaded by M2 to axis by AG [ltlm2]
5 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23> TEL M2 RAOFF = 0.0013 /
6   ↪ [arcsec] RA correction offloaded by M2 to axis by AG [ltlm2]
7 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23>-START DET CHIP WIPE / CCD start
8   ↪ wipe action [ltliab]
9 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23> DET EXP NO = 821 / Unique
10  ↪ exposure ID number [ltliab]
11 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23>-START DET CHIP WIPE / CCD start
    ↪ wipe action [ltliab]
12 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23> DET WIN1 UIT1 = 0.300000 / user
    ↪ defined subintegration time [ltliab]
13 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23> DET EXP WIPETIM = 0 / wipe
    ↪ before starting exposure in a loop [ltliab]
14 Sep 22 00:25:24 wtlts logManager[318044]: wtlts 00:25:23> DET OUT1 GAININD = 4 / Gain
    ↪ index for output [ltliab]
```

### Código 7: Ejemplos de fuerzas aplicadas

## Diccionario

### 1. Sensor de frente de onda

Es un sensor diseñado para calcular las aberraciones de una imagen mientras se toma la misma. Generalmente las aberraciones son producidas por ladeos en los frentes de onda, presentes en imágenes que capturan objetos al otro lado de la atmósfera [Platt y Shack, 2001]

### 2. Frente de Onda

El frente de onda es una superficie imaginaria que representa los puntos correspondientes de una onda que vibra al unisono. Cuando ondas idénticas con un origen en común viajan a través de un medio homogéneo, los senos y picos correspondientes a cada uno se mantienen en fase [Britannica, 2022].

### 3. Librería Template Text Parser

Template Text Parser, o TTP, es una librería de Python que permite la extracción de datos de texto semi estructurado usando plantillas, manteniendo un rendimiento relativamente rápido. Inicialmente fue desarrollado para permitir el acceso procedural a datos producidos por las consolas de aparatos de red, sin embargo, actualmente puede ser usada para extraer cualquier texto semi estructurado que contenga patrones de repetición distintivos [dmulyalin, 2021].

## REFERENCIAS BIBLIOGRÁFICAS

- [Britannica, 2022] Britannica (2022). Wave front. <https://www.britannica.com/science/wave-front>.
- [dmulyalin, 2021] dmulyalin (2021). Overview - ttp. <https://ttp.readthedocs.io/en/latest/Overview.html>.
- [ESO, 1998] ESO (1998). *The VLT White Book*, volumen 1. ESO.
- [ESO, 2011a] ESO (2011a). The m1 cell and m3 tower. <https://www.eso.org/sci/facilities/paranal/telescopes/ut/m1cellm3.html>.
- [ESO, 2011b] ESO (2011b). The secondary mirror, m2 unit. <https://www.eso.org/sci/facilities/paranal/telescopes/ut/m2unit.html>.
- [ESO, 2011c] ESO (2011c). The vlt active optics system. <https://www.eso.org/sci/facilities/paranal/telescopes/ut/actopt.html>.
- [ESO, 2011d] ESO (2011d). The vlt primary mirrors. <https://www.eso.org/sci/facilities/paranal/telescopes/ut/m1unit.html>.
- [Jayathilake, 2011] Jayathilake, P. W. D. C. (2011). A novel mind map based approach for log data extraction. (6).
- [Kolb et al., 2010] Kolb, J., Martinez, P., Sarazin, M., y Navarrete, J. (2010). How to use active optics shack-hartmann data to measure the seeing at the focal plane of a telescope. En *Astronomical site testing data conference*.
- [Ma et al., 2023] Ma, J., Liu, Y., Wan, H., y Sun, G. (2023). Automatic parsing and utilization of system log features inlog analysis: A survey. (21).
- [NASA, 2025] NASA (2025). Overview of the fits data format. <https://heasarc.gsfc.nasa.gov/docs/heasarc/fits/overview.html>.
- [Platt y Shack, 2001] Platt, B. y Shack, R. (2001). History and principles of shack-hartmann wavefront sensing. (4).
- [Wilson et al., 1987] Wilson, R. N., Franza, F., y Noethe, L. (1987). Active optics i. a system for optimizing the optical quality and reducing the costs of large telescopes. 34(4):485-509.