

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 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

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

INTRODUCCIÓN

El observatorio Paranal, ubicado en el Norte de Chile, es uno de los más importantes a nivel mundial. Su enfoque en la investigación estelar se ve apoyado por el VLT, uno de los telescopios ópticos más avanzados del mundo, el cuál a su vez esta compuesto de 4 UTs, entre otros. Estos telescopios están equipados con el sistema de Óptica Activa: un sistema implementado en los UTs, que busca optimizar la imagen tomada por el respectivo telescopio en una observación.

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, luego generar el set de fuerzas que se deben aplicar al espejo principal del telescopio para 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 para modelos de software.

Dadas las nuevas capacidades computacionales de la actualidad, se proponen 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 registros.

Considerando la problemática descrita anteriormente, el objetivo de esta memoria será el desarrollo de un procedimiento que extraiga los datos, bajo contexto del sistema de Óptica Activa, de los registros de log, los transforme de forma acorde, y los deposite en forma de dataset con información útil y estructurada.

Este documento se dividirá 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 más 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 corrobora que la información extraída cumpla con la lógica y la calidad deseada.

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, extrayendola 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 registros de log y extraer sets de información precisas de estos, bajo ciertas condiciones específicas, con tal de generar los datasets deseados.

Este procedimiento es esencial, ya que el análisis as-is de la información dispuesta en dichos logs no es posible, debido a que el costo para reordenar y organizar los datos acorde a lo necesitado supera los beneficios del análisis mismo.

Esto es a causa de que los datos retornados por el sistema de Óptica Activa usan un formato abstracto y poco estructurado. Los registros de log generados durante las operaciones dentro del sistema también usan dicho formato. Además, estos logs registran todas las actividades y factores presentes durante la operación de los UTs en una noche, 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, uno de los telescopio ópticos más avanzados 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; todos estas formas destinadas tanto a fines científicos como para la mejora del funcionamiento del telescopio [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 esta reflejada en el Anexo 1. 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. 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. 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 UT se implementa un sistema de soporte bajo el espejo M1, con 150 pistones de funcionamiento hidráulico y neumático que aplican fuerza al espejo, deformandolo [ESO, 1998].

Durante las observaciones de los UTs, estos 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 calculos, se determina una nueva distribución de fuerzas para los actuadores, la cuál luego es aplicada por los mismos en el M1, brindando a este último de una nueva forma que permita tomar una imagen de mejor calidad [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 hasta lograr una imagen de calida óptima [ESO, 1998].

1.2. ACTORES INVOLUCRADOS

Los actores involucrados en el problema corresponden a los elementos participantes en el sistema de Óptima 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 softcomputing.

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), Interferómetro (por sus siglas en inglés VLTi) y dos telescopios de survey (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. Dicha montura permite al espejo moverse según azimut (eje horizontal) y altitud (eje vertical) [ESO, 1998], como se muestra en la Imagen X.

– INSERTAR IMAGEN DE AZIMUT Y LATITUD –

El M1 está compuesto de espejos más pequeñas, distribuidas en forma de dona. Bajo M1, repartidos en 6 anillos concéntricos, se encuentran 150 actuadores de fuerza axiales; estos son, pistones hidráulicos y neumáticos, donde cada actuador se encuentra debajo de una espejo pequeña respectiva. Estos actuadores dan a M1 una determinada forma óptica, determinada por el patrón de fuerza presentado por los actuadores [ESO, 1998].

El M2 consiste de una espejo de 0.9 metros de diámetro, montada a una distancia de aproximadamente 12.3 metros de M1 a lo largo del eje azimutal, con la espejo de M2 apuntando hacia M1. El M2 además está montado sobre un mecanismo electromecánico que sujeta y controla su inclinación [ESO, 2011b].

El M3 consiste de una espejo elíptica 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 azimutal [ESO, 2011a].

2.2. SISTEMA DE ÓPTICA ACTIVA

La Óptica Activa es un sistema integrado en el M1, encargado de corregir las aberraciones y 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, debido a que esta proporción es la que permite deformar el espejo [Wilson *et al.*, 1987].

El ciclo básico del sistema de Óptica Activa se ilustra en la imagen del Anexo 1. El mismo sigue el siguiente procedimiento:

1. El sensor de frente de onda Shack-Hartmann toma una imagen de una estrella en el cielo y la toma como guía [ESO, 1998].
2. Se analiza la imagen y se miden las perturbaciones ópticas, calculando las aberraciones [Wilson *et al.*, 1987].
3. Cuando la desviación es considerable, se genera un set de fuerzas que se deben aplica 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]. Ejemplos de las imágenes tomadas por el sensor Shack-Hartmann se muestran en las Imágenes X e Y, donde X presenta pocas aberraciones ópticas e Y preseneta más aberraciones:
 - Imagen X –
 - Imagen Y –
5. Todo este ciclo es repetido hasta que la imagen tomada durante el Paso 1 venga con una calidad óptica [Wilson *et al.*, 1987].

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, con cada UT poseyendo uno en un brazo mecánico específico para los sensores. [ESO, 1998]

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

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 registros de log son los único tipo de datos que, valga la redundancia, registran información de la operación interna de un sistema de software, por lo que su rol en la industria es importante. Debido a esto mismo, los registros de log pueden alcanzar tamaños en el orden de hasta el orden de millones de líneas [Ma y Sun, 2023].

Los registros de log de los telescopios UT registra 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, entre otros. Debido a lo anterior, estos registros de log suelen alcanzar el orden de cientos de miles de líneas. Un ejemplo de estos registros de log puede verse en el Anexo 2.

Las líneas de un registro de log estan 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 y Sun, 2023].

2.5. ANÁLISIS DE LOGS

Originalmente, el análisis de registros 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 [Ma y Sun, 2023], debido a que, por la naturaleza de la información contenida en los registros de log, 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, durante la prevalencia del enfoque original, el análisis de registros de log 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 registros de logs, por lo que ya no es posible depender solamente de los métodos anteriormente mencionados. [Ma y Sun, 2023]

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

2.6. ESTRUCTURACIÓN DE LOGS

Los registros de 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 registros de log [Ma y Sun, 2023]. Por esta condición es que se refiere a que los registros de log sean “No Estructurados” o “Semi Estructurados”.

Además, el gran tamaño de los archivos de log promedio también se vuelve un problema para el análisis manual de los registros de log [Ma y Sun, 2023].

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

Actualmente, el workflow de análisis automático de registros de log se divide en 2 etapas centrales [Ma y Sun, 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 registros de log, dividiéndose en los tokens estáticos y valores dinámicos. [Ma y Sun, 2023]
- **Feature Extraction:** Se aplican las plantillas generadas sobre los registros de log para obtener las características, esto es, las variables dinámicas, de los mismos. [Ma y Sun, 2023]

2.7. ORIGENES DE DATOS

Tres orígenes de datos son considerados para la presente memoria:

1. Registros de log: El origen de dato más importante para esta memoria, considerando la problemática a resolver. Estos datos son propios de los telescopios UT, y como tales, se obtienen directamente de los mismos una vez terminada su ejecución [ESO, 1998]
2. Imágenes en formato fits: Corresponden a las imágenes tomadas por el telescopio durante un ciclo de Óptica Activa. Estas imágenes se disponibilizan en formato fits, el cuál es un formato universal en ciencias y astronomía diseñado para almacenar y distribuir datos astronómicos. Este se compone de un Header y uno o más listas multidimensionales, donde el Header contiene la metadata que describe a la misma imagen que lo aloja, y las listas multidimensionales componen el dataset de datos científicos [NASA, 2025]. Las imágenes tomadas durante el ciclo de Óptica Activa también se obtienen directamente de los telescopios UT.
3. Observaciones en formato csv: Corresponde a un archivo csv que registra información sobre las observaciones realizadas en un periodo de tiempo dado. Para obtener este archivo, se debe enviar una query a la página de archivado Observational Raw Data de la ESO (https://archive.eso.org/wdb/wdb/eso/eso_archive_main/query), la cuál disponibiliza información sobre las observaciones de forma pública. Se considera este origen de datos bajo la teoría de que cada observación debe ser válida para ser cargada a la página de archivado.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. 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 del diseño del procedimiento ETL, es necesario hacer un estudio de los registros de log y determinar aquellos que se tengan importancia dentro del contexto 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 distribuciones de fuerzas de los 150 actuadores, las imágenes tomadas durante la operación y las instancias de corrección; todas dentro del rango de una observación.

3.1.1. Instancia de corrección

Se realizan varias correcciones de las distribuciones de fuerza en un ciclo de Óptica Activa, cada una de las cuáles se realiza en base a una imagen tomada anteriormente.

Si se toma en consideración que estas correcciones van en serie, entonces cada una de estas es precedida por la distribución de fuerza que corrigieron, y seguida por la nueva imagen capturada por el telescopio.

Con esto en consideración, se puede separar la instancia de la corrección de las distribuciones de fuerza, y de esta forma tener un dato con el cuál poder relacionar las imágenes y las distribuciones de fuerza.

3.1.2. Distribución de fuerzas

Para el contexto del sistema de Óptica Activa, se hace necesario conocer los valores de las distribuciones de fuerza de los 150 actuadores, ya que esta es la cantidad que se busca op-

timizar durante la Óptica Activa [ESO, 1998].

Durante cada iteración de la Óptica Activa, se analizan las aberraciones de la imagen tomada previamente y luego se calculan los cambios en los valores de la distribución de fuerza en los actuadores del espejo M1. Finalmente, se suman estas diferencias a la distribución de fuerza en vigencia, terminando en una distribución de fuerza final [Wilson *et al.*, 1987].

Tanto los cambios en la distribución de fuerzas como la distribución final son escritas en los registros de log. Como se busca extraer un conjunto de distribuciones de fuerzas, presentes en el registro de log, se decide extraer la distribución final y descartar los cambios en la distribución de fuerzas.

3.1.3. Imágenes tomadas

Durante el proceso de Óptica Activa, las correcciones de las distribuciones de fuerza 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 la nueva distribución de fuerzas para una nueva corrección, continuando el ciclo [ESO, 1998].

Con estas consideraciones en mente, se puede inferir para una instancia de corrección, debe existir una imagen tomada antes y una imagen tomada después, donde esta última correspondería a la imagen previa de la siguiente instancia de corrección.

3.1.4. Datos adicionales

Existen otras características ambientales que son de interés, debido a que podrían estar relacionados con la modificación de la toma de datos desde el espejo.

Debido a esto, se ha generado la siguiente lista de características ambientales que también se deben extraer:

- 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].

3.2. Almacenamiento de datos

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

Tener esta claridad previo al desarrollo del procedimiento ETL permite tener la claridad de que transformaciones realizar a los datos extraídos.

Para la solución a desarrollar, es presenta una estructura de almacenamiento de datos en la Imágen 1.

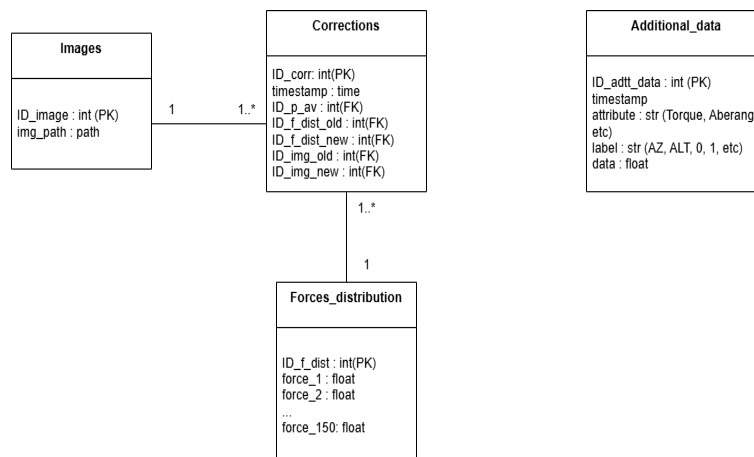


Figura 1: Malla Curricular Ingeniería Civil Informática.
Fuente: Elaboración propia.

3.2.1. Estructura de los datos

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

Se puede considerar entonces agrupar los datos deseados según la instancia de corrección asociada. Por lo señalado anteriormente, cada instancia de corrección tendría asociado un set de datos previos a la corrección y otro set de datos 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, una vez agrupados los datos, se opta por separar las correcciones y los sets de datos asociados en tablas distintas pero relacionadas entre sí con llaves foráneas. Al mantener entidades distintas con múltiples relaciones por tabla, se logra normalizar la información.

Siguiendo con el caso anterior, se tendrían dos tablas para las correcciones más tres tablas para los sets de datos: una tabla para el set de datos pre-corrección asociada a la primera corrección, una tabla para el set de datos post-corrección asociada a la segunda corrección y una tercera tabla para el set de datos entre ambas correcciones, a las cuales se relaciona usando llaves foráneas desde las tablas de las correcciones.

Finalmente, considerando que el set de datos se compone de datos de naturaleza variada (rutas a imágenes, valores numéricos de fuerzas, etc), se decide agrupar los datos en tablas propias según su naturaleza.

Recapitulando, con respecto a los datos asociados a correcciones, se opta almacenar los datos en un modelo de tablas relacionales, compuesta de una tabla central dedicada a las instancias de corrección y varias tablas para los datos correspondientes al antes y después de una corrección. Esta se puede apreciar en la Imagen 2.

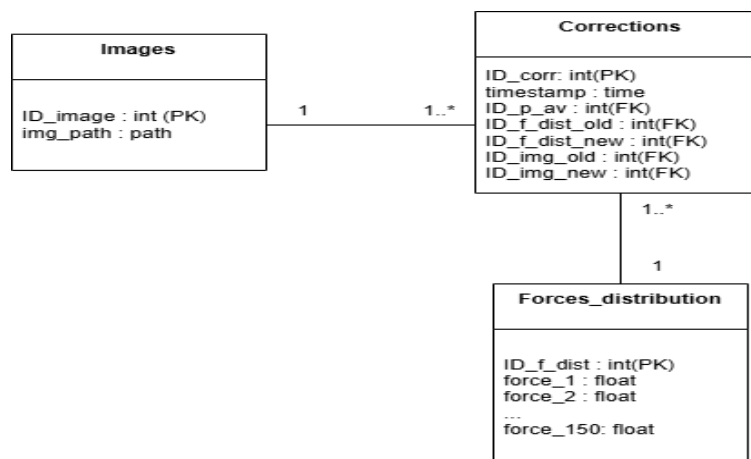


Figura 2: Malla Curricular Ingeniería Civil Informática.

Fuente: Elaboración propia.

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 Imagen 3.

Additional_data
ID_adtt_data : int (PK) timestamp attribute : str (Torque, Aberang, etc) label : str (AZ, ALT, 0, 1, etc) data : float

Figura 3: Malla Curricular Ingeniería Civil Informática.
Fuente: Elaboración propia.

3.2.2. 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 imagen X entrega la impresión que una base de datos SQL sería la mejor opción. Si bien esta puede permanecer como una opción para futuros usos de los datos extraídos, esta opción no sería óptima para el objetivo de esta memoria, debido a que el procedimiento ETL a desarrollar se acoplara a nivel de código con otros modelos de computación y/o análisis, por lo que subir los datos a un servidor SQL para luego bajarlos en el mismo sistema es redundante e innecesario.

Por ende, y considerando el lenguaje de programación usado (Python 3.10.8), la mejor opción es la librería Pandas, más específicamente la estructura Dataframe, la cuál a nivel técnico es homóloga a una tabla SQL dentro de un ambiente Python.

3.3. Descripción general del proceso

El proceso de la solución puede ser revisado en el Anexo 3. 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 registros de log de una noche 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 a las líneas de los registros de log, 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 datasets dedicados para uno de los datos a extraer (según discutido en la sección 3.1) y se ingestan 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.

3.4. Filtrado de líneas de log

Los registros de log 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 registros de log de forma previa, aplicando transformaciones que retornen un conjunto de líneas más apto para el filtrado.

3.4.1. Remoción de tokens estáticos comunes

Tras revisar los registros de log, como el ejemplo presente en el Anexo 2, se puede notar que existe un patrón recurrente en todas las líneas.

Estos tokens iniciales, como se menciona en la sección 2.4, aportan información que es redundante dentro de la misma línea y, a su vez, que es estática, ya que la misma no cambia entre las distintas líneas. De este modo, se puede concluir que dicha información no es necesaria para el objetivo de esta memoria.

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

3.4.2. Cruce con archivo de observaciones

Parte del objetivo de esta memoria es garantizar que la información extraída sea fidedigna y correcta. Por ende, es necesario asegurar que las observaciones a tratar sean exitosas.

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 registros de log no muestran de qué tipo es cada observación, por ende es necesario extraer esta información de un documento externo y cruzarla con las líneas de observaciones ya extraídas, con tal de así poder obtener las líneas que sí influyen en la Óptica Activa.

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.

Para obtener el documento de observación necesario para el procedimiento ETL, se deben añadir los campos mostrados en la Imagen 4

Output preferences: Return max: rows.

Target, Program, and Scheduling Information

Target Name Resolved by SIMBAD Night ☐ 2025 08 04 (YYYY MM(M) DD)

Search Box Input J2000

Output ☐ Sexagesimal (h, deg)

List of Targets No file selected.

Otherwise give a query range using the following start/end dates:

Start End

Program ID ☐ Program Type ☐ Any

PI Col ☐ SV ☐ Any

Title

Observing Information

MarkOptical ☐ MarkInfrared ☐

Imaging ☐ ALL ☐ NONE

Spectroscopy ☐ ALL ☐ NONE

Interferometry ☐ ALL ☐ NONE

Other ☐ ALL ☐ NONE

EFOSC2/LaSilla ☒ CES/LaSilla ☒ AMBER/VLT ☒ ALPACA/Paranal ☒

EMM/LaSilla ☒ CRRES/VLT ☒ GRAVITY/VLT ☒ APICAM/Paranal ☒

ERIS/VLT ☒ EFOSC2/LaSilla ☒ MATISSE/VLT ☒ BOL/APEX ☒

FORS1/VLT ☒ EMM/LaSilla ☒ MID/VLT ☒ HEL/APEX ☒

FORS2/VLT ☒ CRIS/VLT ☒ PIONEER/VLT ☒ FAIR/0ASIS ☒

HAWK/VLT ☒ ESPRESSO/VLT ☒ VINCI/VLT ☒ FAIR/2/OASIS ☒

GROND/LaSilla ☒ FEROS/LaSilla ☒ GRIPS/OASIS ☒

ISAC/VLT ☒ FORS1/VLT ☒ LGSF/VLT ☒

NACO/VLT ☒ FORS2/VLT ☒ MAD/VLT ☒

OMEGACAM/VST ☒ GIRAFFE/VLT ☒ MASCO/Paranal ☒

SOFI/LaSilla ☒ HARPS/LaSilla ☒ SPECULOOS/Paranal ☒

SPHERE/VLT ☒ ISAAC/VLT ☒ WFCAM/Paranal ☒

SUSI2/LaSilla ☒ KMOS/VLT ☒ SOFI/LaSilla ☒

TIMM2/LaSilla ☒ MUSE/VLT ☒ NACO/VLT ☒

VIMOS/VLT ☒ NIRPS/LaSilla ☒ SPHERE/VLT ☒

VIRCA/VISTA ☒ SINFI/VLT ☒ EFOSC2/LaSilla ☒

VISIR/VLT ☒ SOFI/LaSilla ☒ ERIS/VLT ☒

WFI/LaSilla ☒ SPHERE/VLT ☒ NACO/VLT ☒

XSHOOTER/VLT ☒ XSHOOTER/VLT ☒ SPHERE/VLT ☒

Coronagraphy ☐ ALL ☐ NONE

Sparse Aperture Mask ☐ ALL ☐ NONE

Category ☒

SCIENCE ☐ CALIB ☐ ACQUISITION ☒

Data Product Info

Type ☐ Any

User defined input:

Mode ☐ Any

User defined input:

Dataset ID ☐

Orig Name ☐

Release Date ☐

OB Name ☐

OB ID ☐

TPL START ☐

Instrumental Setup

TPL ID ☐

Exptime ☐

(imaging only) Filter bandpass ☒

(imaging only) Bandpass FWHM ☒

Grism ☐

Grating ☐

Site ☐

Figura 4: Malla Curricular Ingeniería Civil Informática.
Fuente: Elaboración propia.

Estos campos permiten obtener todas las observaciones científicas y de adquisición de todos los instrumentos en una noche determinada.

Es posible realizar una petición POST 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

Una vez cargado el documento, se procede a usar el mismo para delimitar los horarios de inicio y término de cada periodo de observación valido, para luego usar estos límites en el filtrado de las observaciones obtenidas desde el registro de logs.

Inicialmente, se pueden crear estos 'bloques 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 estos bloques, se mantenga la distinción de bajo que instrumento se produce la observación en cuestión.

Luego de este paso, es natural intuir que algunos de estos bloques para un mismo instrumento se puedan sobrelapar, lo cuál provocaría que se duplicaran algunas líneas de log tras filtrar. Por ende, tras crear los bloques de tiempo, se debe cerciorar que aquellos bloques que se sobrelapen en horario para un mismo instrumento sean unidos.

Sumado a esto, se conoce que los telescopios estarán en funcionamiento por ciertos intervalos de tiempo previo y posterior a una observación [ESO, 1998]. Por último, se realiza la suposición que entre el fin de un bloque 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 inferior: Cantidad de segundos previo a un bloque de tiempo.
- Margen superior: Cantidad de segundos posterior a un bloque de tiempo.
- Umbral intermedio: Cantidad de segundos entre bloques de tiempo para un mismo instrumento (con márgenes incluidos) donde, si la distancia horaria es menor al umbral, se procede a unir dichos bloques.

Si bien se conoce que cada observatorio es operativo antes y después de una observación formal, la cantidad de tiempo específica (X segundos antes y Z segundos después de cada observación) es desconocida, y no se asegura que sean cantidades estables para cada UT o para una noche específica.

Es por esta razón, que los márgenes y el umbral son paramétricos, con tal de buscar la cantidades bajo las cuales se puede obtener mejores resultados.

3. Clasificación según cruce

Con los bloques de tiempo formados, 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 los bloques de tiempo.

3.5. 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.

Debido a la proliferación y estandarización de este método, se procederá a usar el mismo para las líneas previamente procesadas.

3.5.1. 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.

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, según el análisis realizado en la sección 3.1, se ha determinado usar 16 plantillas para obtener los datos deseados; una plantilla para cada línea a analizar.

3.5.2. 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].

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]:

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.

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.

Como se mencionó en el Capítulo 1, el sistema de logging de Óptica Activa tiene décadas de servicio continuo, por lo que es poco probable que se realicen cambios en esta área. Por el lado de los datos relevantes para el sistema de Óptica Activa, es posible que puedan existir

cambios en los datos requeridos en un futuro, sin embargo el análisis realizado en la sección 3.1, se puede asegurar que los cambios se mantendrían dentro de ese modelo de datos. Por ende, cualquier cambio futuro no incrementa el número de plantillas en gran medida.

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 preprocesamiento, 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.

1. 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 Listing 1:

Listing 1: Plantillas Regex

```
(([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]+)*) 1t[1-4]m1m3 m1as  
m1asSetGlb[Abs|Rel]+ [0-9.-]+ [0-9.-]+ AS Forces  
#([0-9.-]+) \ (abs\): ([0-9.\-s]+))
```

```
(([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]+)*) 1t[1-4]m1m3 m1as  
m1asSetGlb[Abs|Rel]+ [0-9.-]+ [0-9.-]+ AS Executed  
cmd #([0-9.-]+): SUCCESS)
```

```
(([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]+)*) 1t[1-4]m1m3 m1as  
m1asSetGlb[Abs|Rel]+ [0-9.-]+ [0-9.-]+ AS Received  
cmd #([0-9.-]+): [A-Z]+)
```

```
(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
```

```
(.[0-9]+)*> DET EXP N0 = ([0-9]+) \ / Unique exposure
ID number \[(1t[1-4][a-z]{2}[a-c])\])

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL ACT0 (INTTIME) = ([0-9]+))

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL (ALT) (TORQUE) = ([0-9.-]+))

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL (AZ) (TORQUE) = ([0-9.-]+))

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL (ALT)( )= ([0-9.-]+))

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL (AZ)( )= ([0-9.-]+))

(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.-]+))

(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.-]+))

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL (ACT0) (ALTPOS) = ([0-9.-]+))

(wt[1-4]tcs ([0-2][0-9]:[0-5][0-9]:[0-5][0-9]
(.[0-9]+)*> TEL (ACT0) (AZPOS) = ([0-9.-]+))

(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.-]+))

(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.-]+))

(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.-]+))

(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.-]+))
```

2. Extracción de datos con Template Text Parser

Para este caso, se usa la librería externa Template Text Parser, o TTP, la cuál, de forma similar al caso anterior, permite escribir los tokens deseados en la misma 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 Listing 2:

Listing 2: Plantillas TTP

```
{{ date }} {{ hour }} lt[1-4]m1m3 m1as m1asSetGlbAbs  
[0-9.-]+ [0-9.-]+ AS Forces #([0-9.-]+) \ (abs\):  
{{ data }}  
  
{{ date }} {{ hour }} lt[1-4]m1m3 m1as m1asSetGlbAbs  
[0-9.-]+ [0-9.-]+ AS Executed cmd #([0-9.-]+): SUCCESS  
  
{{ date }} {{ hour }} lt[1-4]m1m3 m1as m1asSetGlbAbs  
[0-9.-]+ [0-9.-]+ AS Received cmd #([0-9.-]+): [A-Z]+  
  
wt[1-4]tcs {{ hour }}> DET EXP NO = ([0-9]+) \ / Unique  
exposure ID number \([lt[1-4][a-z]{2}[a-c])\]  
  
wt[1-4]tcs {{ hour }}> TEL ACTO (INTTIME) = ([0-9]+)  
  
wt[1-4]tcs {{ hour }}> TEL {{ group }} >{{ label }} =  
{{ data }}  
  
wt[1-4]tcs {{ hour }}> TEL {{ group }} {{ label }} =  
{{ data }}  
  
wt[1-4]tcs {{ hour }}> TEL {{ group }}{{ label }}=  
{{ data }}  
  
wt[1-4]tcs {{ hour }}> TEL {{ group }}{{ label }}=  
{{ data }}
```

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 distribuciones de fuerza, 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 ejemplo disponible en el Anexo 4.

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

3.6. Guardado de los datos

Una vez extraídos los datos, se deben crear y validar los dataframes donde se almacenaran estos mismos, siguiendo la filosofía descrita en la sección 3.2.

Es necesario no solo crear los dataframes de forma coherente, sino que también validar que los datos ingresados sean correctos ya sea en formato, en contexto, etc.

3.6.1. 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 distribuciones de fuerza anteriores y posteriores.
- Uno para las distribuciones de fuerza, 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 leyó la imagen y la dirección local al archivo .fits correspondiente.

- Uno para todos 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.

3.6.2. Ingesta a los dataframes

Debido a que los datos extraídos se encuentran en formato json, corresponde iterar por todos los datos, y migrar los datos desde el formato json a dataframe.

En la sección 3.4.2. 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 distribución de fuerzas posee 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.

Asi 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.

3.6.3. 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.

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 con ID nulo, y también se remueven aquellas instancias que no poseen distribución de fuerza posterior

a la corrección.

- Para las distribuciones de fuerza, se deben remover las instancias con ID nulo y también se remueven aquellas distribuciones que no son antecedidas y/o seguidas por imágenes únicas.
- Para las imágenes, solo se remueven las instancias con ID nulo.

El dataframe de additional-data no es validado, debido a ser datos de carácter miscelaneo y secundario.

3.6.4. 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 3.2.2.

Para esto, simplemente se hacen calzar los timestamp de los datos de imagen y de distribución de fuerza, y luego se entrega sus ID a la instancia de corrección para incluirlas en sus campos de llaves foráneas.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

4.1. Algoritmo

El algoritmo final sigue el flujo detallado en el Anexo 3, el cuál fue detallado en el Capítulo 3. El mismo se divide en los siguientes pasos:

4.1.1. Log Formatting

Cumple con el proceso planteado en la sección 3.3.1, 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.

4.1.2. Obs Filtering

Cumple con el proceso descrito en la sección 3.3.2, donde se obtiene un archivo csv con las observaciones de la noche seleccionada, para luego formar bloques de tiempo basados en los tiempos que tarda cada observaciones, más el efecto de parámetros de margen y unión, donde finalmente se filtran las líneas de log con tal de quedar solo con aquellas cuya fecha se encuentre dentro de alguno de esos bloques de tiempo.

4.1.3. Log Parsing

Cumple con el proceso descrito en la sección 3.4, donde a base de un archivo plano de plantillas, se itera sobre las líneas de log, aplicando cada plantilla por línea hasta encontrar alguna que calze y guardar la data extraída en un json final.

Como se mencionó en la sección 3.4.2, el método para realizar la extracción es mediante Expresiones Regulares, lo cuál se manifiesta tanto en el archivo de plantillas que se ingresa a la función, junto con las líneas a extraer, como en el método de extracción que se aplica a línea de log.

4.1.4. Generate Dataframes

Cumple con el proceso de la sección 3.5.2, donde se menciona el traslado de los datos del json generado en la función anterior a un conjunto de dataframes, usando los campos `groupby_label` como etiquetas para reconocer a qué dataframe destinar los datos.

De esta función se generan 4 dataframes, como se señalo en la sección 3.5.1: `df_corrections`, para las instancias de corrección; `df_images`, para las imágenes capturadas; `df_f_dist`, para las distribuciones de fuerza de los actuadores; y `df_additional_data`, para los datos adicionales relacionados con la Óptica Activa.

4.1.5. Validate Forces

Cumple con el segundo item de la sección 3.5.3, donde se especifica la validación de los datos dentro de `df_f_dist` generado en la función anterior.

Principalmente, se eliminan los datos con `ID_f_dist` nulo, además de remover aquellos que no sean anteceditas y/o seguidas por imágenes únicas.

4.1.6. Validate Images

Cumple con el tercer item de la sección 3.5.3, donde se especifica la validación de los datos dentro de `df_images` 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 3.5.1, 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 3.5.3, donde se especifica la validación de los datos dentro de `df_corrections` generado en la función anterior.

Principalmente se eliminan los datos con ID_corr nulo, además de remover aquellos sin distribución de fuerza posterior a la corrección.

4.1.9. Link Dataframes

Cumple con el proceso descrito en la sección 3.5.4, en la que se indica como calzar los datos de un dataframe con los datos de dataframe df_corrections 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 df_corrections.

Para esta función, como la construcción de relaciones entre df_images con df_corrections y entre df_f_dist con df_corrections es identico, por lo que se pudo generalizar y parametrizar esta construcción en una misma función, que varía con el dataframe de la cualidad deseada, el nombre de la cualidad deseada y el campo de tiempo dentro del dataframe, como parámetros de entrada.

4.2. Ejecución

Para ejecutar el código desarrollado, se debe entregar la fecha y el número del UT, en ese orden, del registro a analizar. 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 4.2.1. Estas se pueden ver en 3.

Listing 3: Flags del codigo

```
usage: poc-final-v2.py [-h] [-t TEMPLATEFILE]
[-p] [-o] [-i] [-l] [-c] [-s SAVE] [-r] date ut

positional arguments:
  date                  Date to analyzed
  ut                    Number of the UT telescope to be analyzed

options:
  -h, --help            show this help message and exit
  -t TEMPLATEFILE, --templatefile TEMPLATEFILE
                        Path to the text file where the templates
                        are stored (if no name is written,
                        ./poc_templates.txt
                        will be set by default)
  -p, --preprocess      Skips logs pre-processing stage: removal
```

```
of log headers and grouping of log
lines in sections, based on when
AG GUIDE has stopped activity
-o, --obsprocess  Skips observation files processing stage:
filtering of log lines based on time
intervals where a valid observation has
been performed
-i, --linkimages  Skips linking of image log data with
fits files: write of the fits files'
path respective to each image row
in the images' dataframe
-l, --linkdataframes  Skips linking of correction
dataframe with related dataframes:
write of the ids of the force
distribution and image respective
to each row in the correction
dataframe
-c, --console      Prints resulting dataframes on console
-s SAVE, --save SAVE Stores resulting dataframes in
csv files
-r, --report        Prints a report detailing the processing
of log lines
```

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 imagenes presenten 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.

4.3.2. Resultados generales

Los resultados encontrados de forma manual para cada archivo de log se refleja en la Tabla 1. Luego, las cantidades encontradas en los mismos archivos por el algoritmo se encuentran en la Tabla 2.

Tabla 1: Cantidades encontradas de forma manual

Log	Líneas	Correcciones	Fuerzas	Imágenes
L1	427195	626	586	851
L2	384729	598	562	802
L3	409830	537	277	908
L4	371450	546	262	741
L5	444126	491	287	786
L6	422312	495	358	713

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: se ha visto que en los registros de log, existen varias fotos que se pueden tomar consecutivamente en lapsos de unos pocos segundos;

Tabla 2: Cantidades encontradas por el algoritmo

Log	Líneas	Correcciones	Fuerzas	Imagenes	Tiempo (en segundos)
L1	427195	4	5	11	61.92
L2	384729	31	33	54	89.33
L3	409830	4	5	8	69.49
L4	371450	19	19	46	79.14
L5	444126	3	3	4	59.18
L6	422312	18	18	31	103.25

distribuciones de fuerza que no son anticipados ni seguidos directamente por una toma de imagen; entre otros. Estos casos, son detectados por el algoritmo, evitando la extracción de estos datos extra.

El otro motivo de porque la diferencia en el número de cantidades es la filtración por observación: la cantidad de observaciones totales es baja, lo cuál inmediatamente deja fuera a una buena parte de las cantidades presentes en el archivo de log.

Ahora bien, como se menciona en la sección 3.3.2, el filtro de observaciones se realiza con tres parámetros: el margen inferior y el margen superior para cada bloque de tiempo, y el umbral de unión entre bloques 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.

Es posible aumentar estos parámetros, sin embargo tampoco se pueden llegar a extremos muy altos, ya que puede comenzar a presentar ruido por el solapamiento de bloques de tiempo de las observaciones.

4.4. 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, con tal de filtrar los primeros, tal como se explicó en la sección 3.4.2. 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 3

Se puede notar de inmediato el peso de los filtros en esta etapa: en todos los casos probados se redujo el número de líneas de log en más de un 90 %.

Tabla 3: Cantidad de líneas de log filtradas

Log	Líneas entrada	Líneas salida	Porcentaje reducción
L1	427195	3829	99.10 %
L2	384729	23475	93.90 %
L3	409830	4496	98.90 %
L4	371450	22048	94.06 %
L5	444126	3593	99.19 %
L6	422312	27365	93.52 %

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.

4.5. Resultados para Extracción de los datos

Se prueban los dos métodos de extracción mencionados en la sección 3.4.2, usando la serie de parámetros iniciales señalados en el inicio, y se registran los tiempos y cantidad de datos extraídos para cada método:

Tabla 4: Tiempos y cantidad de datos para modos de extracción

Log	Datos TTP	Tiempo TTP (en segundos)	Datos Regex	Tiempo Regex (en segundos)
L1	64	156.56	68	61.92
L2	533	636.70	546	89.33
L3	59	138.05	62	69.49
L4	441	569.63	450	79.14
L5	31	144.02	34	59.18
L6	462	710.65	473	103.25

Con esta prueba, se puede notar que ambos métodos de extracción retornan cantidades similares 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 más del doble que el método con Regex, mientras que en los casos con cientos de datos, el método de TTP tarda alrededor de 7 veces más que el método de Regex.

Con esta prueba, se decide escoger el método de extracción con Regex por sobre el método de extracción usando TTP, basándose exclusivamente en la velocidad del método de extracción predilecto.

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 5

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.

Tabla 5: Cantidad de líneas de log extraídas

Log	Líneas entrada	Líneas salida	Porcentaje reducción
L1	3829	135	96.47 %
L2	23475	940	95.99 %
L3	4496	142	96.84 %
L4	22048	819	96.29 %
L5	3593	73	97.97 %
L6	27365	806	97.05 %

Según lo reflejado en la Tabla 5, los porcentajes de reducción son incluso mayores que en la etapa anterior, donde no bajan del 95 %.

Sin embargo, a diferencia de la etapa anterior, estos porcentajes altos no comprometen la cantidad de datos extraídos; por el contrario, son un sintoma esperable del proceso de aplicación de plantillas.

Para ilustrar esta defensa, se analizan los logs de operación del algoritmo, donde se destacan campos desechados durante la etapa de extracción. Los resultados para cada caso se presenta en la Tabla 6.

Tabla 6: Perdidas de datos durante extracción

Log	Corrección perdidas	Fuerzas perdidas	Imágenes perdidas
L1	2	0	0
L2	2	0	0
L3	3	0	0
L4	20	0	0
L5	0	0	0
L6	17	0	0

Incluso para el caso de L4 y L6, donde se pierden más datos en la tabla, son cantidades menores en comparación a los datos que extraen exitosamente (referidos en la Tabla 4).

4.6. Resultados para Guardado de los datos

Similar a las etapas anteriores, se va a comparar la cantidad de datos que entran a la etapa con la cantidad de datos que salen de la misma.

Solo se contarán datos de distribución de fuerzas, instancias de corrección e imágenes tomadas. Esto porque en esta etapa se realizan validaciones que pueden eliminar datos dependiendo del contexto, y dichas validaciones solo se aplican a esos tipos de datos.

Los resultados de este análisis se presentan en la Tabla 7

Tabla 7: Cantidad de datos filtrados

Log	Datos entrada	Datos salida	Porcentaje reducción
L1	26	20	30 %
L2	146	118	19.18 %
L3	24	18	25 %
L4	116	84	27.59 %
L5	14	10	40 %
L6	100	67	33 %

Si bien presenta bajas en los números de datos, son porcentajes menores a los presentes en las etapas anteriores. Es posible que estas bajas sean causados por las validaciones, como se menciono anteriormente, considerando que en dichas validaciones se remueven valores con nulidad o que no cumplan con el orden distribución de fuerza - imagen establecido.

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, intentando implementar métodos de la forma más fiel a la fuente original, hasta que se decantó por tomar las bases de estos métodos para desarrollar uno propio.

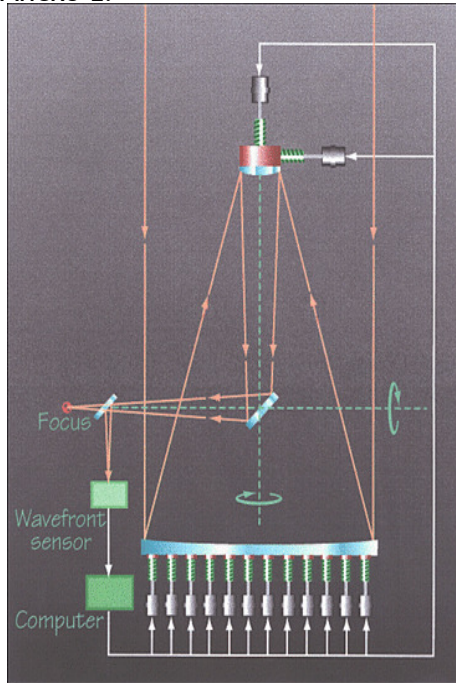
Igualmente, durante una parte considerable del desarrollo, se consideraron dos posibles métodos de extracción: una usando Expresiones Regulares, y otra usando una librería especializada de Python. Si bien ambas opciones son válidas, para el caso de esta memoria se demostró que la opción de Expresiones Regulares era la más apta.

El algoritmo actual genera datasets con una cantidad considerable de datos, extraídos con la seguridad total de que se generan durante las observaciones del telescopio. Sin embargo, el algoritmo deja 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 su 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. Se debe evaluar 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

- Anexo 1:



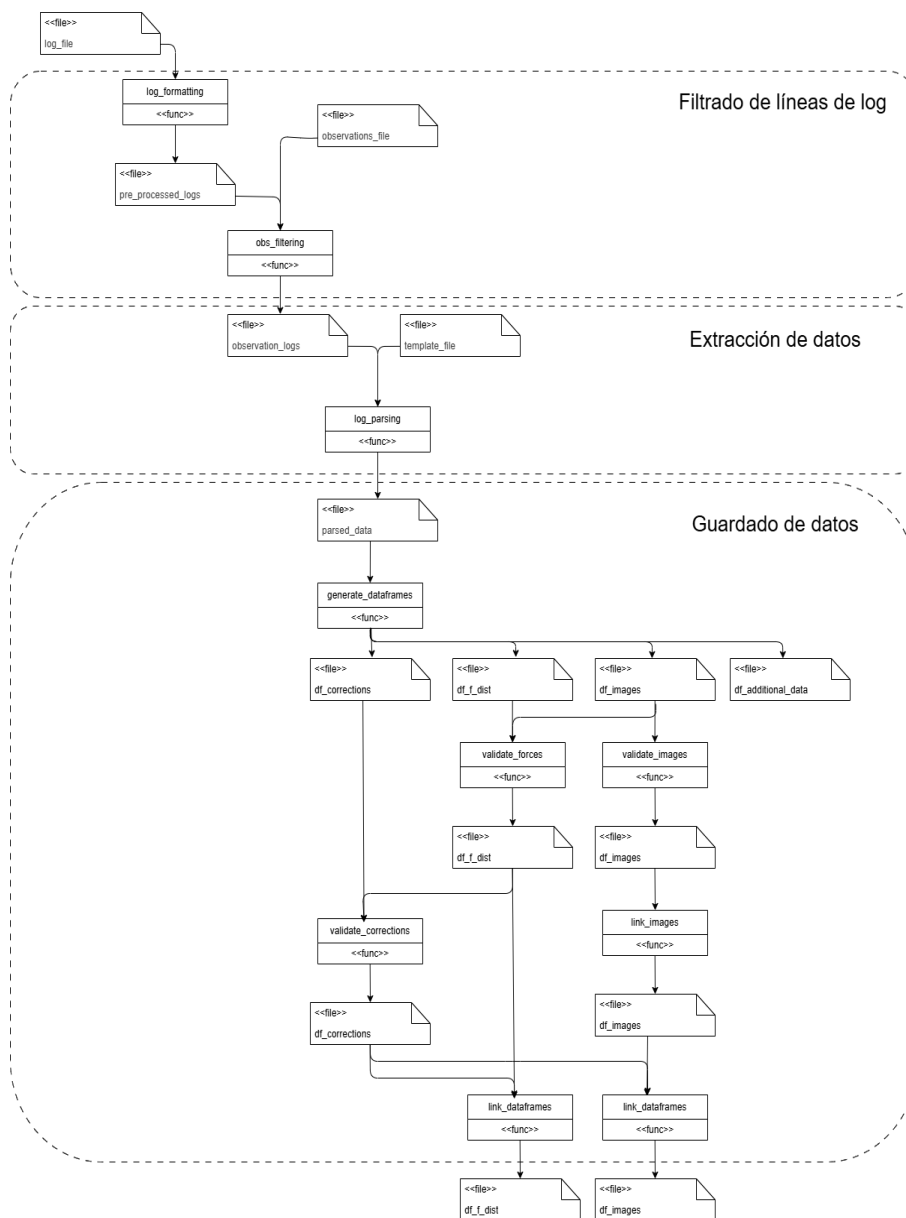
- Anexo 2:

```

Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:11> DET WIN1 UTT1 = 45.000000 / user defined subintegration time [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:11> START DET EXP / Start exposure [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:11> START DET CHIP WIPE / CDD start wipe action [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET EXP NO = 25771 / Unique exposure ID number [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> START DET CHIP WIPE / CDD start wipe action [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 UTT1 = 45.000000 / user defined subintegration time [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET EXP WUPETIM = 0 / wipe before starting exposure in a loop [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 STRX = 1 / Lower left pixel in X [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 STRY = 1 / Lower left pixel in Y [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 BINX = 2 / Binning factor along X [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 BINY = 2 / Binning factor along Y [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 NX = 1074 / # of pixels along X [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 NY = 1032 / # of pixels along Y [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 STRX = 1 / Lower left pixel in X [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 STRY = 1 / Lower left pixel in Y [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 REFX = 210.000 / X position of reference [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 REFY = 310.000 / Y position of reference [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 CENTROID = 'none' / type of centroiding calculation [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 BACKGROUND = 0.000000 / sky background [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 THRMIN = 0.000000 / Lower threshold [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> DET WIN1 FILTER = 'none' / Filter type [ltiaa]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> TEL AG AGRA = 0.1289 / [arcsec] RA correction offloaded by AG [ltiaga]
Aug 5 01:13:12 wtltcs logManager[47772]: wtltcs 01:13:12> TEL AG ADDEC = -0.0562 / [arcsec] DEC correction offloaded by AG to axis [ltiaga]
Aug 5 01:13:13 wtltcs tifControl[48141]: 2025-08-05 01:13:13.760633 wtltcs tif tifCO 99 0 TIF : Received command: OFFSAD, Buffer: -offsetAlpha -0.009 -offsetDelta
Aug 5 01:13:13 wtltcs trkwsControl[48147]: 2025-08-05 01:13:13.764541 wtltcs msw trkwsControl 103 0 Sent Command OFFSAD to ltiaa:alttrkServer with -0.009,-0.030
Aug 5 01:13:13 wtltcs trkwsControl[48147]: 2025-08-05 01:13:13.766928 wtltcs msw trkwsControl 103 0 Sent Command OFFSAD to ltiaa:aztrkServer with -0.009,-0.030
Aug 5 01:13:13 wtltcs trkwsControl[48147]: 2025-08-05 01:13:13.771507 wtltcs msw trkwsControl 103 0 Sent Command OFFSAD to ltiaa:rottrkServer with -0.009,-0.030
Aug 5 01:13:13 wtltcs logManager[47772]: wtltcs 01:13:13> START TEL OFFSET / Initiate new telescope offset [wtltcs]
Aug 5 01:13:13 wtltcs logManager[47772]: wtltcs 01:13:13> TEL OFFSET RA = -0.009 / [arcsec] Offset applied in RA [wtltcs]
Aug 5 01:13:13 wtltcs logManager[47772]: wtltcs 01:13:13> TEL OFFSET DEC = -0.030 / [arcsec] Offset applied in DEC [wtltcs]
Aug 5 01:13:13 wtltcs logManager[47772]: wtltcs 01:13:13> TEL RA = 275.641316 / [deg] 18:22:33 actual telescope pos RA 12000 [wtltcs]
Aug 5 01:13:13 wtltcs logManager[47772]: wtltcs 01:13:13> TEL DEC = -16.596145 / [deg] -16:35:46 actual telescope pos DEC 12000 [wtltcs]

```

■ Anexo 3:



■ Anexo 4:

```

AS Executed cmd #218073: SUCCESS
AS Forces #1 (abs): 23 21 17 9 -2 -16 -29 8 12 21 32 42 49 -23 -22 -9 8 20 -21 62 -35 -11 7 -38 -29
AS Forces #2 (abs): -22 -18 -11 1 12 21 24 14 19 21 21 21 16 11 -10 -4 -5 -14 -36 -24 104 96 17 11 -23 -17
AS Forces #3 (abs): -23 -24 -27 -29 -25 12 10 13 21 31 40 -16 -18 -9 6 15 -25 89 -28 -17 -6 -33 -34
AS Forces #4 (abs): 7 8 -0. -11 -28 -23 -24 50 43 38 38 41 43 9 2 5 13 -19 -26 22 65 -46 -66 -43 -41
AS Forces #5 (abs): -74 -72 -66 -56 -41 -22 -4 75 81 81 76 69 60 43 55 56 44 26 -30 -8 -75 -82 -92 -6 -11
AS Forces #6 (abs): -72 -67 -59 -51 -46 -42 -37 66 59 56 54 54 52 29 21 21 23 -26 -35 -14 13 -67 -90 -19 -33

```


- **Diccionario 1: Frente de Onda**

El frente de onda es una superficie imaginaria que representa los puntos correspondientes de una onda que vibra al unísono. 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].

- **Diccionario 2: 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, 2001]

- **Diccionario 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>.
- [Jayathilake, 2011] Jayathilake, P. W. D. C. (2011). A novel mind map based approach for log data extraction. (6).
- [Ma y Sun, 2023] Ma, J., L. Y. W. 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, 2001] Platt, B.; Shack, R. (2001). History and principles of shack-hartmann wavefront sensing. (573-577).
- [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.