



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**GENERACIÓN DE IMÁGENES MULTI RESOLUCIÓN PARA
IDENTIFICACIÓN DE GALAXIAS EN TIEMPO REAL**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

DIEGO JESÚS LEÓN GARAY

PROFESORES GUÍA:
BENJAMÍN BUSTOS CÁRDENAS
FRANCISCO FORSTER BURON

MIEMBROS DE LA COMISIÓN:
CLAUDIO GUTIÉRREZ GALLARDO
JOSÉ SAAVEDRA RONDO

Este trabajo ha sido parcialmente financiado por:
FONDECYT

SANTIAGO DE CHILE
2023

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL
EN COMPUTACIÓN
POR: DIEGO JESÚS LEÓN GARAY
FECHA: 2023
PROF. GUÍA: BENJAMÍN BUSTOS

GENERACIÓN DE IMÁGENES MULTI RESOLUCIÓN PARA IDENTIFICACIÓN DE GALAXIAS EN TIEMPO REAL

En el presente trabajo de memoria se presenta el desarrollo de una infraestructura digital la cual sea capaz de almacenar y entregar una elevada cantidad de imágenes multi resolución astronómicas en un corto lapso de tiempo.

El proyecto chileno astronómico ALerCE tiene la misión de procesar alertas de supernova provenientes del próximo observatorio Vera Rubin para lograr predecir la galaxia anfitriona asociada a cada alerta. Si bien ALerCE creó el algoritmo DELIGHT el cual es capaz de identificar la galaxia anfitriona descargando imágenes astronómicas de resolución simple, el rendimiento de este mismo es ineficiente. En primer lugar, el algoritmo requiere de imágenes de resolución múltiple para realizar sus predicciones, las cuales son generadas a partir de la imagen de resolución simple. Además de lo anterior el algoritmo descarga tal imagen de resolución simple en un tiempo considerable. Frente tal problema nace la oportunidad de entregarle al algoritmo DELIGHT un servidor dedicado especialmente a la rápida entrega de imágenes multi resolución astronómicas. El proyecto ALerCE desea lograr obtener 10 mil conjuntos de 5 imágenes de resolución múltiple en aproximadamente 40 segundos.

Así, en primer lugar se analizaron los métodos utilizados por el algoritmo DELIGHT para descargar las imágenes astronómicas de resolución simple. Se demostró la necesidad de un servidor propio para la consulta eficiente de imágenes multi resolución, al evidenciar la incapacidad de los métodos presentes para obtener imágenes de resolución múltiple de forma rápida. Luego, se definió la estructura de datos a utilizar para el almacenamiento de imágenes astronómicas multi resolución. Se presentó el proceso de descarga de las imágenes que corresponden a las partes del cielo que se desea tener acceso junto con detallar la forma en la que estas fueron almacenadas utilizando el esquema de datos definido.

Posteriormente, se mostraron los algoritmos necesarios para el procesamiento que requieren las imágenes multi resolución almacenadas para que estas mismas puedan ser entregadas al algoritmo DELIGHT. Una vez teniendo el servidor local creado con las imágenes almacenadas en formato multi resolución, se evaluaron los rendimientos otorgados por el algoritmo ideado en el trabajo de memoria. Se obtienen tiempos muy cercanos a los exigidos por el proyecto ALerCE para la consulta de 10 mil conjuntos de 5 imágenes de resolución múltiple con las características que DELIGHT exige. En específico, ejecuciones en paralelo del algoritmo escrito en el trabajo de memoria si logran entregar la cantidad de conjuntos pedidos en el tiempo solicitado. Finalmente, se concluye sobre todo el trabajo realizado en la memoria. Se entrega el recuento de objetivos tanto generales como específicos, se comenta sobre las ventajas y desventajas de la solución obtenida, y se recomienda el trabajo futuro que debiese realizar el proyecto ALerCE sobre los resultados de esta memoria para continuar con el trabajo expuesto.

*“Siempre intentaste. Siempre fallaste.
No importa. Intenta otra vez.
Falla de nuevo. Falla mejor”.*

Stanislas Wawrinka

Agradecimientos

Antes que nada quisiera aprovechar esta instancia para agradecer a toda persona que de una u otra forma me ha apoyado durante todo este proceso universitario.

Agradecer en primer lugar a mi familia. A mis padres Humberto y Marcela los cuales han mostrado un apoyo sincero e incondicional durante toda mi vida y no solo en mi proceso universitario. Sin el ánimo y respaldo que me entregan día a día nada de esto habría sido posible. Se han esforzado mucho por entregarme enseñanzas y valores importantes y siempre estaré muy agradecido por ello. A mis hermanos Marcelo y Marcela por ser personas ejemplares para mí. Todo lo vivido con ellos dejaron enseñanzas muy profundas en mí y siempre estaré muy agradecido por toda la preocupación y dedicación que mostraron sobre el desarrollo de mi persona. La lealtad, el amor, y el compañerismo que me han dado desde que tengo consciencia es algo que no tiene comparación. Los amo mucho. A cada uno de ustedes.

También hacer mención especial a toda mi familia restante tanto paterna como materna. A mis abuelos, a mis tíos, a mis primos y a mis sobrinos. A los que están y a los que no. Gracias a cada uno de ustedes por formar una familia fuerte y unida, y por enseñarme que el amor y el cariño son ingredientes fundamentales en las relaciones interpersonales.

Agradecer a todo mi círculo tenístico. Desde que agarré mi primera raqueta a los 7 años, muchas personas se han vuelto muy importantes para mí. A mis grandes amigos tenistas. Gracias a cada uno de ustedes por enseñarme lo que es ser amigos y rivales al mismo tiempo. Gracias por enseñarme de respeto, de admiración, de tolerancia, de esfuerzo y de persistencia. Gracias a mis entrenadores más importantes. A Jaime, a José Miguel, a Pablo, a Jorge y a Oswaldo. Cada uno de ustedes me ayudó a formar la mentalidad que tengo ahora. Me enorgullece haber tenido la posibilidad de aprender tan profundamente de sus personas y de su trabajo. Son unos excelentes trabajadores y formadores. Mención especial a Jaime Montero por todo lo que tuvo que aguantar de mí. Por acompañarme siempre a mis viajes más importantes. Por estar conmigo en mis victorias más dulces y en mis derrotas más amargas. Por celebrar o llorar conmigo. Eres increíble Jaime.

A mis amigos que he formado en mi vida fuera del ámbito deportivo. Gracias por siempre mostrar confianza en mí y por estar a mi lado en los momentos más difíciles. Por generar espacios y momentos de relajación los cuales sin lugar a duda fueron siempre muy importantes para mantenerme enfocado y tranquilo en mis deberes universitarios.

Por último, a mis profesores guía y co-guía, Benjamín y Francisco. Gracias profesor Benjamín por la preocupación y dedicación mostrada hacia mi trabajo de memoria. Gracias por darme tranquilidad cuando más la necesite, por todos los consejos que me dió durante todo el proceso, y por el constante análisis respecto a los avances semanales. Fue un proceso muy pulcro y estoy muy agradecido de sus esfuerzos por mantener un trabajo constante y ameno. Gracias profesor Francisco por todo el aprendizaje que me ha brindado hacia mi formación

buscada de astrónomo e ingeniero civil en computación. Gracias por la excelente oportunidad que me presentó para poner en práctica las herramientas que pretendo utilizar en mi futuro, y gracias por la constante ayuda que recibí en los momentos donde entendía bastante poco como podía avanzar en el trabajo. He tenido la oportunidad de trabajar con excelentes personas las cuales a su vez muestran un conocimiento importante y una experiencia bastante alta. Fue un orgullo realizar mi trabajo de memoria bajo la supervisión de dos grandes académicos de la escuela.

Tabla de Contenido

1. Introducción	1
1.1. Contexto	1
1.2. Problema y Relevancia	1
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
2. Estado del Arte	4
2.1. Algoritmo DELIGHT	4
2.2. PanSTARRS PS1	4
2.3. Servidor Panstamps	5
2.4. Limitaciones	6
2.5. Esquema HiPS	6
2.6. Esquema HTM	8
3. Análisis del Servidor Panstamps	10
3.1. Evaluando Panstamps	10
3.2. Servidores Alternativos	10
4. Creación de Servidor HiPS	13
4.1. Servidor HiPS de Prueba	13
4.2. Descarga de imágenes PANSTARRS	14
4.3. Creación de servidor HiPS	15
5. Generación de Imágenes Multi Resolución	18
5.1. Generación de Imagen de Resolución Simple	19
5.2. Generación de Imágenes de Resolución Múltiple	21
6. Evaluación	25
6.1. Imágenes de Resolución Simple	25
6.2. Imágenes de Resolución Múltiple	26
6.3. Mejoras Realizadas	27
6.3.1. Imágenes de Resolución Simple	28
6.3.2. Imágenes de Resolución Múltiple	28
7. Conclusiones	31
7.1. Recuento de Objetivos	32
7.2. Impacto del Trabajo Realizado	32

7.3. Trabajo Futuro	33
7.4. Comentarios Finales	33
Bibliografía	34
Anexos	36
A. Ejemplos de Generación de Imágenes	36
A.1. Ejemplos de Generación de Imagen de Resolución Simple	36
A.2. Ejemplos de Generación de Imágenes de Resolución Múltiple	38

Índice de Tablas

3.1.	Tiempo de ejecución para consulta y descarga de imágenes, utilizando servidores <i>Panstamps</i> . y <i>fitscut.cgi</i>	11
4.1.	Órdenes de profundidad (Desde orden 4 hasta orden 11) con la respectiva resolución por pixel de sus archivos HiPS para un tamaño de 512×512 pixeles por archivo.	16
4.2.	Tiempo y tamaño en memoria utilizado en cada paso importante de la creación del servidor HiPS.	17
5.1.	Órdenes de profundidad de los archivos HiPS a utilizar para el cálculo de las imágenes a distintas resoluciones por pixel, junto con la resolución por pixel que identifican a tales órdenes.	22
6.1.	Reducción de tiempo para la obtención de imágenes <i>PanSTARRS</i>	28
6.2.	Tiempo promedio máximo entre N ejecuciones en paralelo para la generación de 1,000 conjuntos de 5 imágenes multi resolución.	30

Índice de Ilustraciones

2.1.	Representación visual de la porción de cielo cubierta por la misión <i>PanSTARRS PS1</i>	5
2.2.	Teselación HEALPix de la esfera celeste a distintos órdenes de profundidad. . .	7
2.3.	Teselación HTM a nivel 0.	9
4.1.	Representación visual de la porción de cielo que será cubierta por el servidor HiPS objetivo.	15
4.2.	Algunas de las propiedades importantes del servidor local HiPS creado para el proyecto ALeRCE.	17
5.1.	Comparación de imagen HiPS v/s imagen Panstamps.	18
5.2.	Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características.	21
5.3.	Ejemplo de imágenes multi resolución. En la parte superior se muestran las 5 imágenes multi resolución de tamaño 30×30 pixeles centralizadas en las coordenadas celestes (36.549250, 27.606694). En la parte inferior, se muestra la imagen de resolución simple que debiese tener como input la implementación actual de <i>DELIGHT</i> para generar las imágenes de la parte superior.	24
6.1.	Tiempo en segundos en función del número de imágenes de resolución simple generadas. Cada imagen generada cuenta con una resolución de $0.25''/\text{pixel}$ y un tamaño de 480×480 pixeles.	26
A.1.	Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características. Coordenadas centrales (240.346160, 26.291187). .	36
A.2.	Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características. Coordenadas centrales (344.920080, 24.071530) .	37
A.3.	Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características. Coordenadas centrales (60.866428, 19.181077). .	37
A.4.	Ejemplo de imágenes multi resolución. Cada imagen centralizada en las coordenadas celestes (246.259991, 11.997976).	38
A.5.	Ejemplo de imágenes multi resolución. Cada imagen centralizada en las coordenadas celestes (16.541830, 31.406560).	39
A.6.	Ejemplo de imágenes multi resolución. Cada imagen centralizada en las coordenadas celestes (179.613540, 10.017920).	39

Capítulo 1

Introducción

1.1. Contexto

Uno de los grandes proyectos astronómicos que presenciara la comunidad científica en los próximos años será el observatorio Vera Rubin. Tal proyecto se encargará de procesar y almacenar productos de datos de aproximadamente 20 terabytes cada noche, los cuales buscan abordar preguntas sobre la estructura y evolución del Universo.

Tal cantidad de datos debe ser procesada a través de infraestructuras digitales sin precedentes que logren atender la excesiva información que provendrá del observatorio. El proyecto astronómico chileno ALERCE es uno de los proyectos elegidos para procesar datos del nuevo observatorio Vera Rubin [1].

Un tipo de alertas astronómicas que deberá procesar ALERCE para el proyecto Vera Rubin será la de alertas asociadas a explosiones de supernova. Cada una de estas alertas, a través de un procesamiento interno, deberá ser asociada a su respectiva galaxia anfitriona. Parte de tal problema es el que se aborda en esta memoria.

En el contexto de esta memoria, se entenderá al concepto de “imagen de resolución simple” como el resultado de coañadir imágenes hechas a partir de las múltiples exposiciones tomadas durante las observaciones, lo que da como resultado una imagen del cielo con una cobertura más uniforme que una sola exposición. Tales imágenes, en los catálogos astronómicos que se nombrarán más adelante, reciben el nombre de *Stack Image*. Por su parte, las imágenes multi resolución de una determinada *Stack Image* corresponden a versiones redimensionadas de la imagen principal. Una mejor resolución significará un campo de visión menor de la imagen. Por su parte, una peor resolución será consecuencia de un campo de visión mayor de la imagen de resolución simple. Es importante aclarar que una resolución mejor es menor en valor que la resolución original, y una resolución peor es mayor en valor que la resolución original.

1.2. Problema y Relevancia

Recientemente, ALERCE desarrolló un algoritmo que dada una imagen de resolución múltiple, es capaz de identificar la galaxia anfitriona [2]. La implementación actual descarga imágenes de resolución simple que se convierte en un conjunto de 5 imágenes multi resolución, lo que es altamente ineficiente. Además el algoritmo utiliza un servidor externo, llamado *Panstamps* [3], para descargar tales imágenes de resolución simple, lo que se realiza en un

tiempo considerable.

Por lo tanto, se vuelve necesario e importante levantar infraestructuras digitales, e implementar métodos para el almacenamiento y consulta de grandes volúmenes de imágenes multi resolución para cumplir con el propósito ya dicho, los cuales logren tiempos de ejecución que satisfagan las necesidades de ALeRCE. Se espera un orden de tiempo de 40 segundos para aproximadamente 10,000 consultas de imágenes multi resolución. Es decir, se desea un tiempo cercano a 4[ms] para consultar un conjunto de 5 imágenes de resolución múltiple.

Esta infraestructura digital deberá entregar las imágenes multi resolución que podrían ser eventualmente consultadas por el algoritmo DELIGHT. Además de lo anterior, el servicio a generar debe responder a la consulta de estas imágenes de una manera suficiente.

Tal infraestructura objetivo será la encargada de almacenar y entregar todas las imágenes multi resolución necesarias para procesar las alertas de un determinado stream, correspondientes a explosiones de supernova que serán detectadas por el observatorio Vera Rubin.

1.3. Objetivos

1.3.1. Objetivo General

Con lo dicho hasta este punto se puede definir el objetivo general del trabajo de memoria, el cual consiste en la obtención de un servicio de almacenamiento y consulta de una elevada cantidad de imágenes multi resolución astronómicas. Se desea que tal servicio sea capaz de entregar un conjunto de 5 imágenes multi resolución en un tiempo aproximado de 4[ms].

1.3.2. Objetivos Específicos

El servicio mencionado en el objetivo general, será en primera instancia programado mediante técnicas de consulta al servidor *Panstamp*s. Una vez logrado lo anterior, y ya teniendo claridad sobre los métodos eficientes para realizar una conexión suficiente, se deberá levantar un servidor propio el cual cumpla con los propósitos buscados. Así, se pueden listar los siguientes objetivos específicos:

1. Estudiar el método de conexión al servidor *Panstamp*s utilizado por el algoritmo DELIGHT.
2. Evaluar los métodos utilizados por el servidor *Panstamp*s para la entrega de imágenes astronómicas.
3. Diseñar algoritmos los cuales logren tiempos de conexión y de respuesta óptimos hacia el servidor *Panstamp*s. En caso de que *Panstamp*s utilice otros servidores, se deberá lograr la conexión hacia esos servidores.
4. Estudiar estructuras de datos eficientes para el almacenamiento y consulta de imágenes. Se espera que tales estructuras permitan descargar imágenes multi resolución de manera directa.
5. Complementar las estructuras de datos escogidas para el almacenamiento, con métodos eficientes de indexación.

6. Levantar un servidor, el cual almacene eficientemente las imágenes astronómicas de interés en formato multi resolución, y logre los tiempos de conexión y de respuesta buscados para la descarga de tales imágenes.

Capítulo 2

Estado del Arte

2.1. Algoritmo DELIGHT

DELIGHT es un algoritmo escrito por el proyecto ALerCE el cual, descargando una imagen de resolución simple centralizada en las coordenadas celestes deseadas, es capaz de identificar la galaxia anfitriona [2]. Hoy, el algoritmo funciona de forma ineficiente en términos de tiempo de descarga de imágenes, lo que no permite procesar la cantidad de imágenes deseadas por unidad de tiempo. Además de lo anterior, a pesar de que el algoritmo necesita tales imágenes descargadas en múltiples resoluciones, DELIGHT descarga sólo una única resolución de la imagen, para luego encargarse de obtener las versiones multi resolución de la misma.

El algoritmo utiliza el servidor *Panstamps* para descargar todas las *Stack Images* necesarias. Mediante una lista de identificadores y de coordenadas (ascensión recta y declinación), el algoritmo descarga imágenes *PanSTARRS*, cada una centrada en las coordenadas indicadas, con un tamaño de $2[\text{arcmin}] \times 2[\text{arcmin}]$. *PanSTARRS* es un sistema de imágenes astronómicas desarrollado y operado por el Instituto de Astronomía de la Universidad de Hawaii.

Algo sumamente importante de notar hasta este punto, es que el tiempo de descarga de imágenes de resolución simple por parte del algoritmo DELIGHT, es totalmente dependiente del tiempo que utiliza el servidor *Panstamps*. La acción de descarga contemplará los tiempos de conexión y respuesta del servidor, así como también el tiempo de búsqueda de las imágenes en el servidor. La implementación actual del algoritmo muestra una velocidad de descarga promedio de 4 segundos por imagen [2].

2.2. PanSTARRS PS1

Para obtener las imágenes *PanSTARRS*, el Instituto de Astronomía de la Universidad de Hawaii llevó a cabo la misión de sondeo de cielo llamada *PanSTARRS PS1*. El sondeo *PS1* cubre una fracción de 3/4 del cielo (Figura 2.1) en longitudes de onda óptica, utilizando 5 filtros ópticos: *g*, *r*, *i*, *z*, e *y* [4]. Para realizar las observaciones se utilizó el *telescopio PS1* el cual se encuentra ubicado en la cumbre del volcán Haleakala, en la isla Maui que forma parte del archipiélago hawaiano. Como resultado, el sistema de imágenes se compone por aproximadamente 200,000 imágenes astronómicas por cada filtro óptico, teniendo un tamaño de datos de aproximadamente 15 TB por filtro [5]. Tal sistema es consecuencia del

procesamiento de datos realizado por la misión *PS1* y sus imágenes se caracterizan por una resolución de $0.25''/\text{pixel}$ [6].

Se pueden acceder a los productos de datos resultantes de *PanSTARRS PS1* mediante consultas SQL a la base de datos relacional del *Archivo Mikulski para Telescopios Espaciales* [7]. MAST (debido a sus siglas en inglés) es un archivo de datos astronómicos centrado en las longitudes de onda ultravioleta, óptica, e infrarroja. El archivo alberga datos de distintas misiones de escaneo de cielo como lo son Webb, Hubble, TESS, y PS1.

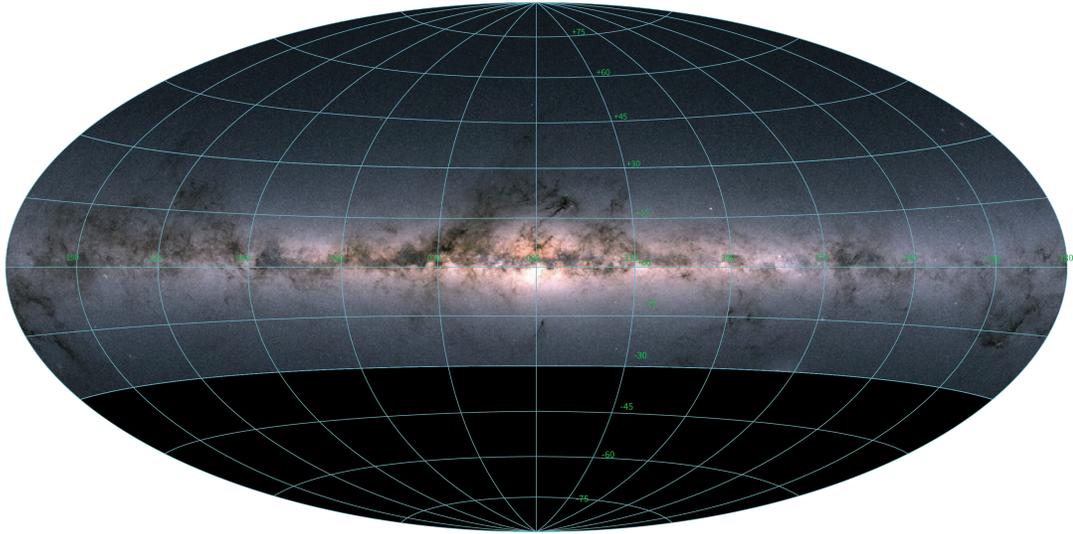


Figura 2.1: Representación visual de la porción de cielo cubierta por la misión *PanSTARRS PS1*.

2.3. Servidor Panstamps

El servidor *Panstamps* consiste básicamente de códigos escritos en lenguaje *Python*, los cuales permiten conectarse mediante peticiones HTTPs al sistema de imágenes astronómicas *PanSTARRS*. *PanSTARRS* cuenta con una [página de libre acceso](#), la cual recibiendo las coordenadas astronómicas, el tipo de archivo de imagen (JPEG o FITS), y el tamaño de la imagen en pixeles, entrega las imágenes correspondientes a las partes de cielo solicitadas, en la resolución indicada.

Es decir, el servidor *Panstamps* no almacena ningún tipo de imagen astronómica, y sólo ayuda al algoritmo DELIGHT a realizar la petición HTTPs correspondiente para obtener la imagen que requiere el algoritmo. Por lo tanto, *Panstamps* se encuentra limitado a descargar imágenes de la forma en que la página de *PanSTARRS* lo permita. En específico, sólo se puede consultar de a una imagen a la vez, con tiempos de espera significativos (del orden de 0.5 segundos) por cada imagen.

Así, se descubre que el servidor *Panstamps* no realiza consultas directamente al servidor de almacenamiento de las imágenes astronómicas *PanSTARRS*, sino más bien realiza peticiones a una interfaz simple que consiste en una página web, sostenida por el mismo servidor

PanSTARRS, pero que entrega la descarga de imágenes en órdenes de tiempo insuficientes para los futuros propósitos del algoritmo DELIGHT.

2.4. Limitaciones

En base a las dos subsecciones anteriores, se puede mencionar que el funcionamiento del algoritmo DELIGHT se encuentra limitado, al no contar con un servidor con métodos para almacenar y consultar grandes volúmenes de conjuntos de imágenes multi resolución, del orden de 10 mil conjuntos de 5 imágenes cada 40 segundos. Tales conjuntos de imágenes son las que el algoritmo DELIGHT utiliza de input, y son necesarias para que el algoritmo pueda asociar galaxias anfitrionas a un gran número de alertas asociadas a explosiones de supernova en tiempo real.

Implementar un sistema que descargue directamente las imágenes multi resolución es importante, ya que esto se traducirá en una mayor eficiencia en la descarga de imágenes, junto con una mayor velocidad de ejecución del algoritmo.

2.5. Esquema HiPS

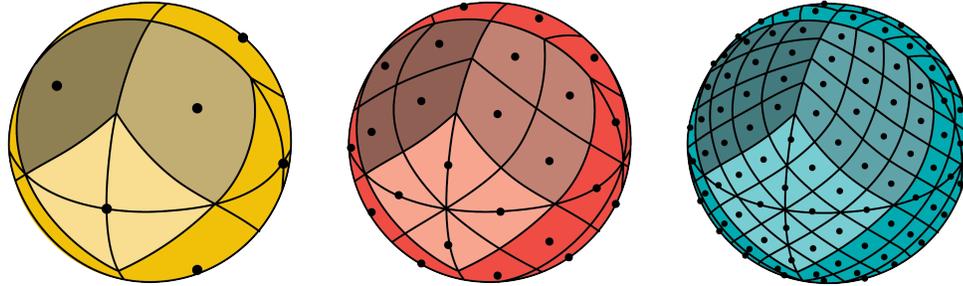
Ya que se requiere un servidor local el cual sea capaz de almacenar y consultar grandes volúmenes de imágenes multi resolución, se necesita un método genérico para el empaquetamiento, almacenamiento y consulta de tales imágenes. Todo esto es cubierto por la estructura HiPS (Hierarchical Progressive Surveys), la cual consiste principalmente de un esquema jerárquico para la descripción, almacenamiento y acceso a datos de levantamientos del cielo. El sistema se basa en mosaicos jerárquicos de las regiones del cielo con una resolución espacial cada vez más fina, lo que facilita una vista progresiva de un estudio y admite la panorámica de múltiples resoluciones [5]. HiPS fue desarrollado en el marco del proyecto *Aladin*, el cual es un atlas celeste interactivo que permite al usuario principalmente visualizar y manipular imágenes astronómicas de distintas misiones de encuesta del cielo [8]

El esquema HIPS es implementado como una estructura de archivos simple, con un sistema de indexación directa que conduce a implementaciones prácticas. En un principio, HIPS puede manejar cualquier tipo de data la cual puede localizarse y desplegarse en la esfera celeste [9], tal como lo son las imágenes *PanSTARRS* que utiliza la implementación actual del algoritmo DELIGHT. Es decir, HiPS logra almacenar una determinada data astronómica utilizando su propio esquema jerárquico.

La estructura HiPS utiliza la indexación HEALPix (Hierarchical Equal Area isoLatitude Pixelization tessellation technique) para generar los mosaicos jerárquicos del cielo. HEALPix es básicamente una partición de igual área de la esfera que admite una estructura de árbol jerárquica para aplicaciones de resolución múltiple [10]. La técnica de teselación HEALPix se refleja en particiones recursivas del cielo, donde cada partición define un nivel (llamado orden de profundidad HiPS) de resolución menor. En particular, se comienza con 12 divisiones de la esfera celeste de igual área (orden 0), para luego comenzar con subdivisiones recursivas de igual área en cada una de las celdas generadas [10]. Tales subdivisiones son ejemplificadas en Figura 2.2. Por lo tanto, en el primer orden de profundidad se almacenan 48 archivos, en el segundo 192, deduciendo inmediatamente la relación:

$$N_n = 12 \times 4^n, \tag{2.1}$$

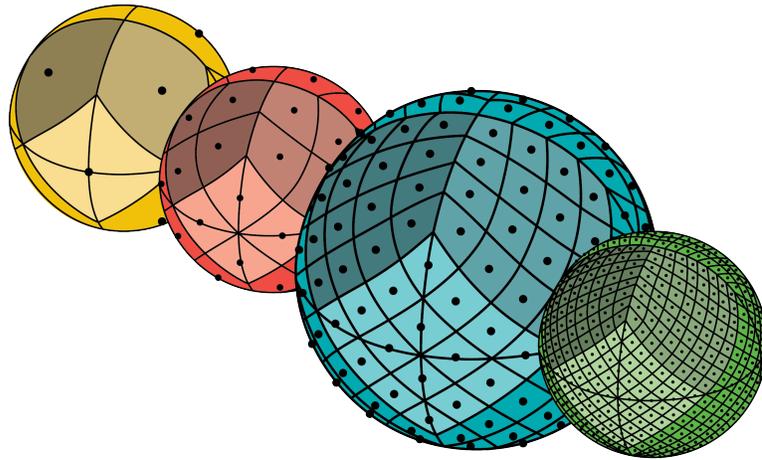
donde N_n denota la cantidad de archivos almacenados en el orden n . Estos archivos definen la unidad básica de almacenamiento en un servidor HiPS, y contienen en ellos la data astronómica organizada espacialmente en las celdas HEALPix. En particular, para los propósitos del proyecto tales unidades contendrán los archivos .fits que corresponden a las imágenes de las distintas partes del cielo, a distintas resoluciones. Es importante mencionar que las partes de cielo a almacenar corresponden a fracciones específicas de la esfera celeste, las cuales serán definidas por el proyecto ALeRCE [1].



(a) Partición original. Orden 0 con 12 divisiones de la esfera celeste de igual área.

(b) Primera partición recursiva. Orden de profundidad 1 con 48 divisiones de la esfera celeste de igual área.

(c) Segunda partición recursiva. Orden 2 con 192 divisiones de la esfera celeste de igual área.



(d) De izquierda a derecha, órdenes de profundidad 0, 1, 2 y 3 del esquema HiPS.

Figura 2.2: Teselación HEALPix de la esfera celeste a distintos órdenes de profundidad.

Así, el esquema HiPS se organiza en forma de archivos simples, organizados en un árbol de directorios. La estructura de estos directorios sigue la jerarquía *orden – archivo*, según el orden de resolución deseado a acceder, y según la parte del cielo requerida. En específico, la ubicación (índice) del archivo en un determinado orden de profundidad es calculado directamente desde la ubicación espacial ingresada. Es decir, según la ascensión recta y declinación (coordenadas de la esfera celeste) recibidas, y según el orden HEALPix entregado, la inde-

xación HEALPix logra calcular el índice del archivo en el árbol en tiempo constante, usando la “fórmula HEALPix” [10].

El orden más profundo a almacenar en un servidor HiPS se calcula considerando la resolución de las imágenes entrantes, junto con el tamaño en pixeles escogido para los archivos HiPS [9]. Tal tamaño debe ser una potencia de 2 por exigencia del esquema. En el orden más profundo los archivos en teselación HEALPix son obtenidos directamente desde la data original, mientras que los pixeles de órdenes superiores se calculan a través de los 4 pixeles que cubren su misma región en un nivel más de profundidad [9].

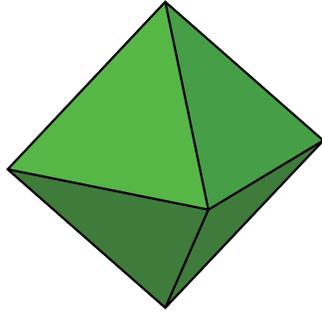
El modelo fue generado con el propósito principal de dar facilidad en el uso y en el acceso de data astronómica sin la necesidad de servidores especiales o sistemas de bases de datos [9]. Una característica importante del esquema, es que su mayor uso en la comunidad recae en el acceso de imágenes de distintas partes del cielo en distintas resoluciones. Es inferible así que tal uso es exactamente el que DELIGHT le daría a su propio servidor HiPS, lo que permite plantear el argumento de que el esquema es el modelo de almacenamiento de datos que el algoritmo requiere para la obtención de sus inputs.

Usualmente, los servidores HiPS son montados en línea para un acceso remoto por el cual distintos clientes pueden acceder a el mediante peticiones HTTP. Sin embargo, en el caso de que sea factible, un usuario con fines particulares puede levantar un servidor local HiPS para un acceso dedicado y eficiente. En tal caso, ningún mecanismo de distribución ni registro del servidor es necesario. Al ser el servidor local un simple acceso al directorio raíz por el cual se pueden acceder a los distintos niveles y archivos de forma jerárquica, los tiempos de acceso obtenidos son rápidos y de orden constante [9].

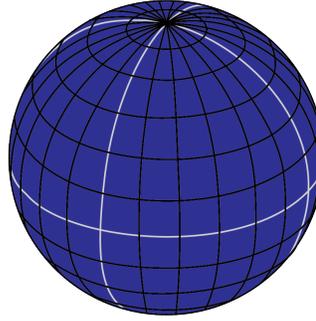
2.6. Esquema HTM

Tal como se mencionó en los objetivos específicos definidos en Sección 1.3, la meta es contar con más de una opción de estructura de datos orientada al almacenamiento y consulta de imágenes multi resolución astronómicas. La segunda alternativa contemplada consiste en el esquema conocido como Hierarchical Triangular Mesh, o más bien HTM, el cual es un esquema de subdivisión de la superficie de la esfera celeste en triángulos esféricos de forma y tamaño similar [11]. De una forma parecida al esquema HiPS, la descomposición recursiva de la estructura HTM comienza en un caso base llamado nivel 0. Tal partición inicial se obtiene proyectando los bordes de un octaedro regular sobre una esfera, creando así 8 triángulos esféricos los cuales 4 corresponden al hemisferio norte y los 4 restantes al hemisferio sur. Este hecho es visible en Figura 2.3.

Los 8 triángulos esféricos denotados en (b) de Figura 2.3 son denominados “trixeles” de nivel 0. Cada trixel del nivel 0 es dividido en 4 trixeles de igual tamaño y forma para generar el nivel 1 HTM. Así, recursivamente se pueden ir generando trixeles de niveles superiores. Los trixeles son la unidad básica del esquema jerárquico HTM y ellos representarían por ejemplo las imágenes astronómicas que contendrán las distintas partes de cielo a almacenar, en distintas resoluciones. La estructura HTM utiliza *Quadtrees* para almacenar los trixeles y admitir así la búsqueda de imágenes en multi resolución [11]. En forma breve, un *Quadtree* es un tipo de árbol utilizado para representación de imágenes que consiste de una partición recursiva del espacio en dos dimensiones, descomponiendo una región en cuatro partes iguales en tamaño y forma.



(a) Octaedro regular.



(b) Nivel 0 HTM

Figura 2.3: Teselación HTM a nivel 0.

Dado que la estructura de datos HTM utiliza *Quadrees* para almacenar sus archivos básicos se espera que el tiempo de consulta de estos mismos sean mayores en comparación al esquema HiPS. Lo anterior se debe netamente a que el servidor HiPS es implementado como un árbol básico de directorios en donde mediante la fórmula HEALPix se puede calcular en tiempo constante la ubicación de un archivo HiPS a cualquier resolución. Por su parte, la búsqueda de un trixel a un determinado nivel en el esquema HTM se efectúa mediante un método recursivo.

Distintas pruebas realizadas sobre los dos esquemas muestran que el tiempo de búsqueda de trixeles a nivel 10 en la estructura HTM es superior en cerca de un orden de magnitud en comparación al tiempo de búsqueda de archivos de orden 10 en el esquema HiPS [12]. Además de lo anterior se suma el hecho de que se demuestra que el tiempo de consulta de archivos HiPS a distintas resoluciones es de orden constante mientras que el tiempo en la consulta de archivos HTM aumenta con el nivel de profundidad [12]. Por estas razones y teniendo en cuenta de que los propósitos del proyecto ALerCE recaen en una rápida consulta de imágenes multi resolución astronómicas, se decide optar por la alternativa del esquema HiPS para cumplir con los objetivos del trabajo de memoria.

Capítulo 3

Análisis del Servidor Panstamps

3.1. Evaluando Panstamps

Como ya se mencionó anteriormente en el Capítulo 1, los primeros 3 objetivos específicos del trabajo de memoria consisten netamente en entender, ejecutar, y evaluar el servidor *Panstamps*. Así, en primer lugar se estudió en profundidad la conexión que realiza el algoritmo DELIGHT con el servidor *Panstamps*. Se ejecutó DELIGHT, estudiando los tiempos de ejecución que este mismo arroja al descargar las imágenes correspondientes a 32 coordenadas astronómicas distintas, obteniendo un tiempo cercano a 169 segundos.

Luego, con ánimos de confirmar qué tanto se demora *Panstamps* únicamente en consultar imágenes al servidor *PanSTARRS*, se ejecutó la API escrita en *Python* la cual permite descargar imágenes a través de la página de *PanSTARRS* mediante peticiones HTTPs. El repositorio de *Panstamps* es de libre acceso, por lo que las interfaces que permiten conectarse al servidor *PanSTARRS* se pueden ejecutar en cualquier computadora. Los tiempos de ejecución obtenidos son bastante lejanos a lo buscado por el proyecto ALERCE, siendo estos del orden de 6 segundos en promedio para la descarga de cada imagen, en concordancia con los resultados expuestos en la ejecución del algoritmo DELIGHT. Estas mediciones corresponden a la columna $t_{panstamps}$ en la Tabla 3.1.

Luego, sobre la investigación acerca de métodos más eficientes para la descarga de archivos entregados por el servidor *Panstamps*, al estudiar en profundidad el código fuente, y al entender que *Panstamps* más bien consulta una página web en vez de a un servidor de almacenamiento directo, se concluye que los métodos utilizados por *Panstamps* para la descarga de imágenes del servidor *PanSTARRS* son ineficientes e inmejorables, ya que peticiones HTTPs para la descarga secuencial de objetos es lejano a lo requerido por ALERCE para lograr los tiempos de descarga de imágenes deseado. Todo el trabajo que realiza el código para hacer efectiva la petición HTTPs con los parámetros ingresados, junto con poder rescatar la información deseada del contenido HTML en respuesta a la petición, deja fuera de alcance el orden de tiempo buscado por DELIGHT.

3.2. Servidores Alternativos

Dado que los códigos de *Panstamps* se conectan a una interfaz simple de usuario para proveer acceso y descarga a imágenes *PanSTARRS*, es que nace la pregunta de si el servidor *PanSTARRS* cuenta con servidores web de acceso más complejo los cuales puedan ser

utilizados para acceder y descargar imágenes mediante scripts. Lo anterior, se busca con el propósito de medir los tiempos de ejecución de tales scripts para compararlos con los tiempos de *Panstamps*, intentando demostrar así que un servidor local con acceso directo es lo que necesita el proyecto.

Se encuentra así el servicio *fitscut.cgi*, el cual es un programa diseñado para extraer recortes de una imagen en particular, según las coordenadas ingresadas como parámetros. Es un script provisto por *PanSTARRS* para obtener la descarga directa de imágenes en la resolución solicitada, a través de una petición GET en el protocolo HTTP. Es decir, el servicio responde a las consultas mediante el ingreso de parámetros en un determinado link en el cual se encuentra *fitscut.cgi*. En particular, se pueden realizar las mismas consultas que al servidor *Panstamps* ingresando como parámetros la ascensión recta, la declinación, el tamaño de la imagen, el tipo de archivo de imagen, y el filtro utilizado. Ingresando correctamente los parámetros en la URL donde se encuentra el servicio *fitscut.cgi*, se logran descargar las imágenes solicitadas.

Se midió la consulta y descarga de 6 imágenes distintas utilizando los servicios *Panstamps* y *fitscut.cgi*. En general, se encuentra que ambos servicios son ineficientes para los propósitos de DELIGHT, demostrando así que se requiere montar un servidor local el cual entregue las imágenes eficientemente. Además, se nota en los resultados de la Tabla 3.1 que el servicio *Panstamps* funciona en general peor en términos de tiempo de descarga que el servicio *fitscut.cgi*. Esto da indicios de que los métodos estudiados en los códigos de *Panstamps* son claramente mejorables.

Tabla 3.1: Tiempo de ejecución para consulta y descarga de imágenes, utilizando servidores *Panstamps*. y *fitscut.cgi*.

ra	dec	size	filter	$t_{panstamps}[s]$	$t_{fitscut.cgi}[s]$
67.58268	10.099121	2 [arcmin]	r	4.562	3.686
83.63321	22.01446	2 [arcmin]	r	5.534	3.396
73.63321	22.01446	2 [arcmin]	g	4.653	3.887
332.60045	2.199809	2 [arcmin]	i	5.401	2.622
70	-10	2 [arcmin]	y	6.070	2.698
80	-5	2 [arcmin]	z	9.350	2.678

Luego, con ánimos de demostrar lo ineficiente que es para el algoritmo DELIGHT el utilizar servicios generales sin un propósito específico para la entrega de imágenes multi resolución de distintas partes del cielo, se generó un script en *Python* el cual, utilizando el servicio *fitscut.cgi*, descarga secuencialmente 3 imágenes del servidor *PanSTARRS*. Se midió el tiempo de ejecución de este archivo, siendo este de aproximadamente 9 segundos. En base a los resultados de la Tabla 3.1, junto con el tiempo de 9 segundos para la descarga de 3 imágenes de forma secuencial, se concluye que el tiempo promedio de descarga para imágenes multi resolución como las que el algoritmo DELIGHT necesita, es de aproximadamente 3 segundos utilizando el servidor *fitscut.cgi* proporcionado por *PanSTARRS*. Por su parte, según la tabla ya mencionada, el servicio *Panstamps* cuenta con un tiempo promedio aproximado de 6 segundos. A pesar de que DELIGHT no se estaba conectando al servicio más eficiente, se infiere que las interfaces para usuarios disponible para una descarga progresiva de imágenes

de una respectiva parte del cielo, no son suficientes para el algoritmo DELIGHT.

El trabajo explicado hasta este punto, permite inferir una característica muy importante la cual debe cumplir el algoritmo DELIGHT para efectivamente poder descargar 10 mil conjuntos de 5 imágenes multi resolución en aproximadamente 30 segundos, y es que DELIGHT necesariamente debe tener acceso a un servidor orientado al empaquetamiento, almacenamiento y consulta de imágenes de resolución múltiple correspondientes a distintas regiones del cielo. A su vez, tal servidor debe estar esquematizado de tal forma que permita acceder a tales imágenes con un tiempo de ejecución de orden constante, y que idealmente tal tiempo no se vea afectado por el orden de resolución accedido.

Es así como, según los resultados obtenidos hasta el momento, se justifica en gran medida la necesidad de continuar con la segunda parte de la solución propuesta, la cual consta del estudio completo del esquema HiPS para así levantar un servidor local HiPS que atienda las necesidades del algoritmo DELIGHT.

Capítulo 4

Creación de Servidor HiPS

Una vez estudiado y entendido el esquema HiPS, se comenzó con el diseño del servidor HiPS objetivo del proyecto. Como se indicó anteriormente, se debe tener claridad sobre la resolución angular con la cual se quieren almacenar las imágenes primordiales de las partes del cielo deseadas. Dado que para efectos del proyecto se desea contar con una resolución angular de $0.25[arcsec]$, junto con escoger un tamaño por defecto de las imágenes de 512×512 pixeles, se obtiene que nuestro servidor HiPS se caracterizará por 11 niveles de profundidad.

Como se mencionó en Sección 2.5, el esquema HiPS fue desarrollado en el marco del proyecto *Aladin*. En la página web oficial del proyecto, se encuentra una documentación importante sobre como generar de manera correcta un servidor HiPS. En particular, la documentación describe un software para generar automáticamente un servidor HiPS mediante una data entrante, la cual debe contener como mínimo los archivos a almacenar (por ejemplo en formato .fits o .png), junto con las propiedades de tal input, en donde debe ser accesible la resolución angular de las imágenes, las coordenadas celestes de cada imagen, etc.

4.1. Servidor HiPS de Prueba

Antes de comenzar con la creación del servidor HiPS objetivo del proyecto, se creó un pequeño servidor de prueba para analizar y entender el proceso de generación de este mismo. Además, se desea medir el tiempo de acceso hacia los distintos archivos almacenados en el servidor. De esta forma se estará poniendo a prueba tanto el esquema de datos escogido para el proyecto, como también los métodos de indexación que este esquema ofrece para la eficiente consulta de archivos. Para utilizar las funcionalidades necesarias que ofrece la indexación HEALPix se propone el uso de *healpy*, el cual es un framework escrito en lenguaje *Python* que atiende las herramientas específicas de HEALPix [13]. Así, se podrán utilizar las funciones que ofrece el módulo *healpy* para el cálculo de los índices de los archivos HiPS según las coordenadas celestes y nivel de profundidad ingresados.

Para la creación del servidor de prueba se utilizó como input a las imágenes consultadas por el código de prueba del algoritmo DELIGHT, para así hacer una comparación con el tiempo de ejecución de este mismo en lo que a la descarga de tales imágenes se refiere. Estas 32 imágenes resultan en un servidor HiPS bastante sencillo, pero suficiente para realizar los experimentos mencionados.

Se escribe así un script en lenguaje *Python*, el cual, utilizando el archivo .csv utilizado en el código de prueba de DELIGHT, junto con ocupar algunas de las funciones ofrecidas por el

módulo *healpy*, entrega el tiempo de demora en la consulta y acceso de 32 distintos archivos correspondientes al nivel más alto de profundidad de las partes del cielo que contienen a las coordenadas celestes recibidas. El tiempo arrojado por el script número 1 el cual fue recién detallado, fue de:

$$t_1 = 0.0382[s]. \quad (4.1)$$

Como se puede notar, el tiempo obtenido es del orden de $10^{-2}[s]$, mientras que el tiempo arrojado por el código de prueba de DELIGHT es del orden de $10^3[s]$. Por su parte, el tiempo promedio obtenido para la descarga de una imagen es de:

$$\bar{t}_1 = 1.194 \times 10^{-3}[s]. \quad (4.2)$$

En este punto, es importante recordar que el tiempo promedio de consulta de imágenes deseado por el proyecto ALERCE para la ejecución de DELIGHT es de $4 \times 10^{-3}[s]$. Claramente \bar{t}_1 es inferior a este valor.

Dado que el algoritmo DELIGHT idealmente desea acceder a 5 imágenes de resolución distinta por cada coordenada, se ideó la escritura de un script en lenguaje *Python* muy similar al anterior, pero que por cada coordenada recibida calcule la ubicación del archivo en el nivel más profundo del árbol de directorios HiPS, y la ubicación en los 4 niveles superiores al más profundo. Así, se desea estudiar si el ingreso a distintos niveles de profundidad afecta o beneficia al tiempo promedio de acceso en las imágenes.

Así, el script número 2 escrito obtiene un tiempo total de ejecución de:

$$t_2 = 0.1162[s]. \quad (4.3)$$

Por lo tanto, dado que se acceden a $32 \times 5 = 160$ archivos .fits distintos, el tiempo promedio del script número 2 viene dado por:

$$\bar{t}_3 = 7.263 * 10^{-4}[s]. \quad (4.4)$$

Es inferible así, que acceder a disintos niveles de las mismas coordenadas, es más rápido que la consulta de distintas coordenadas en un mismo nivel de profundidad. Por su parte, el tiempo promedio de acceso por coordenada celeste distinta viene dado por:

$$\bar{t}_2 = 3.31 * 10^{-3}[s]. \quad (4.5)$$

Así, se puede notar que el tiempo promedio para la consulta de un conjunto de 5 imágenes multi resolución sigue siendo inferior a la cota de tiempo establecida por el proyecto ALERCE.

4.2. Descarga de imágenes PANSTARRS

Los experimentos señalados en Sección 4.1 dieron los resultados esperados para proceder a la generación del servidor HiPS objetivo del proyecto. Ya teniendo la certeza de que el acceso a los archivos HiPS es suficiente para los propósitos de DELIGHT, se debía poner énfasis en la descarga de las imágenes que serían la entrada del algoritmo que genera el servidor HiPS.

Se propuso como objetivo del proyecto generar un servidor HiPS utilizando todas las imágenes *PanSTARRS* que corresponden al hemisferio norte de la esfera celeste las cuales

son pertenecientes a un filtro óptico de los 5 mencionados en Sección 2.2. Por consecuencia, se propuso descargar 132,190 imágenes del sistema de imágenes *PanSTARRS*. En específico, se escogió el filtro *r* para generar el servidor HiPS del proyecto. Tal filtro, se compone de 199,703 imágenes en el sistema *PanSTARRS* de las cuales 132,190 corresponden a la cobertura del cielo norte de la misión *PS1*. Tales imágenes se descargan en extensión *.fits* y por tanto, cada imagen cuenta con sus propiedades disponibles a través de su *fits header*.

Para llevar a cabo la descarga de imágenes se utilizó la librería de Python *astroquery*, la cual es un conjunto de herramientas para consultar datos de distintos archivos astronómicos con interfaces expuestas en Internet [14]. En particular, dado que el archivo de datos astronómicos *MAST* cuenta con su propia interfaz para consultar sus datos, la librería *astroquery* soporta consultas hacia tal archivo de datos, y por consecuencia soporta a su vez la consulta de imágenes *PanSTARRS*.

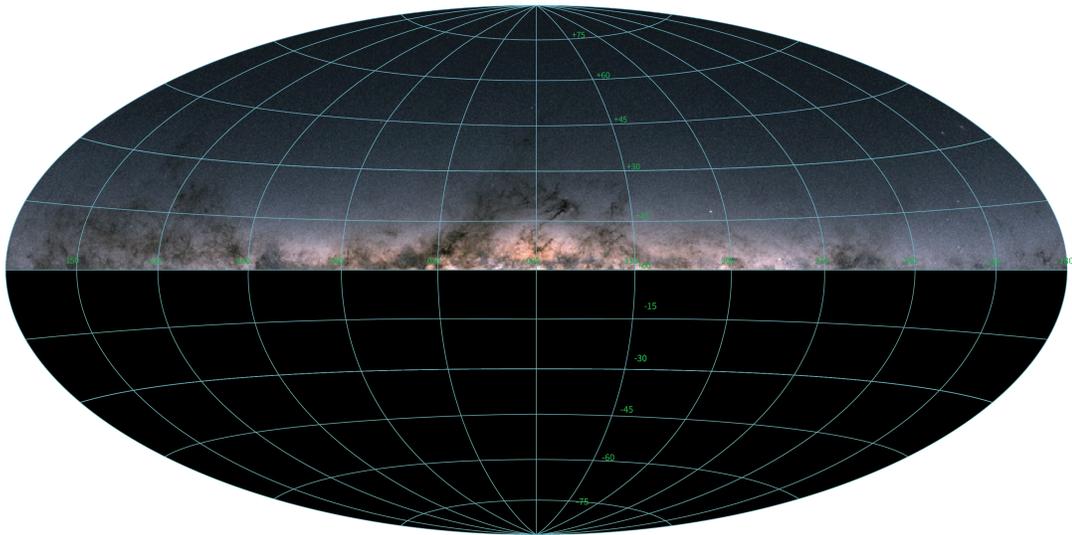


Figura 4.1: Representación visual de la porción de cielo que será cubierta por el servidor HiPS objetivo.

Se genera así el script llamado “MastPS1_Download” en lenguaje *Python* el cual de forma secuencial descarga las 132,190 imágenes *PanSTARRS* pertenecientes al filtro óptico *r*. La descarga de cada imagen demora aproximadamente 4 segundos a una velocidad de transferencia cercana a $16[MB/s]$, teniendo un tiempo total de descarga de imágenes de aproximadamente 9 días. Es importante destacar que si bien este proceso es paralelizable, para efectos del trabajo de memoria no fue necesario. Finalmente se cuenta con el almacenamiento de las 132,190 imágenes descargadas utilizando un espacio aproximado de 8.2 TB.

4.3. Creación de servidor HiPS

Tal como se indicó en el principio de este Capítulo, en la página oficial del proyecto se describe un algoritmo el cual, mediante una data entrante de imágenes astronómicas junto con sus propiedades, logra generar el servidor HiPS que cubre las partes del cielo que se

encuentran presentes en las imágenes entrantes. El código del algoritmo (llamado *Hipsgen*), es de libre acceso y se encuentra escrito en lenguaje *Java*.

Es directo inferir que las imágenes descargadas utilizando *astroquery* vienen preparadas para ser ingresadas como input en el algoritmo *Hipsgen*, al estar cada imagen en un formato aceptable para el algoritmo junto con tener disponible sus respectivas propiedades. Es importante a su vez tener presente que las imágenes del sistema *PanSTARRS* a utilizar como data entrante cuentan con una resolución de $0.25''/\text{pixel}$. Como se mencionó en Sección 2.5, el orden más profundo que caracterizará al servidor HiPS a generar se calcula teniendo en cuenta tanto la resolución por pixel de la data entrante como el tamaño en pixeles de los archivos HiPS a crear. Tal tamaño en pixeles es sugerido como 512×512 pixeles y la resolución por pixel de cada orden estará determinada únicamente por tal parámetro. Así, el último orden de profundidad a almacenar debe ser tal que su resolución sea la primera en ser menor a la resolución de la data entrante [9].

Tabla 4.1: Órdenes de profundidad (Desde orden 4 hasta orden 11) con la respectiva resolución por pixel de sus archivos HiPS para un tamaño de 512×512 pixeles por archivo.

orden	r_H
4	$25.7766''/\text{pixel}$
5	$12.8832''/\text{pixel}$
6	$6.4416''/\text{pixel}$
7	$3.2208''/\text{pixel}$
8	$1.6104''/\text{pixel}$
9	$0.8052''/\text{pixel}$
10	$0.4026''/\text{pixel}$
11	$0.2013''/\text{pixel}$

Observando la Tabla 4.1 es directo que el servidor HiPS de ALERCE contará con 12 niveles distintos de profundidad, desde el orden HiPS 0 al orden HiPS 11. Como se mencionó en Sección 2.5, en el orden más profundo los archivos HiPS son calculados directamente desde la data original. Por lo tanto, utilizando la herramienta *Hipsgen* con las 132,190 imágenes *PanSTARRS* con resolución de $0.25''/\text{pixel}$ como input, se generan a partir de ellas imágenes astronómicas con resolución de $0.2013''/\text{pixel}$ (Orden HiPS 11). Estas imágenes del orden más profundo son las utilizadas para generar los archivos HiPS de orden 10, lo que a su vez comienza el proceso recursivo que calcula los archivos HiPS de todos los niveles de profundidad restantes.

Como resultado final de la creación del servidor, se obtiene un servidor HiPS de orden 11 con alrededor de 33.6 millones de archivos HiPS que conforman los distintos 12 niveles de profundidad, y que utilizan un almacenamiento aproximado de 34.6 TB. El algoritmo *Hipsgen* demoró cerca de 8 días para generar tal servidor HiPS. Estas y otras propiedades importantes del servidor HiPS resultante son visibles en la Figura 4.2.

Tabla 4.2: Tiempo y tamaño en memoria utilizado en cada paso importante de la creación del servidor HiPS.

Paso	Tiempo	Tamaño en memoria
Descarga <i>PanSTARRS</i>	9 [días]	8.2 [TB]
<i>Hipsgen</i>	8 [días]	33.6 [TB]
Total	17 [días]	41.8 [TB]

```

hips_builder      = Aladin/HipsGen v12.044
hips_version      = 1.4
hips_frame        = equatorial
hips_order        = 11
hips_order_min    = 0
hips_tile_width   = 512
#hips_service_url = ex: http://yourHipsServer/DL
hips_status       = public master clonableOnce
hips_initial_ra   = 307.84997
hips_initial_dec  = 14.19831
hips_initial_fov  = 0.43431
hips_pixel_scale  = 5.591E-5
s_pixel_scale     = 6.944E-5
dataprodct_type   = image
hipsgen_date      = 2023-06-30T22:03Z
hipsgen_params    = maxThread=25 in=mastDownload out=HiPS id=HiPSID/DL INDEX TILES CHECKCODE DETAILS
hips_pixel_bitpix = -32
data_pixel_bitpix = -32
hips_sampling     = bilinear
hips_overlay      = overlayMean mergeOverwriteTile treeMean
hips_release_date = 2023-07-09T07:57Z
hips_tile_format  = fits
hips_pixel_cut    = -294 113488
hips_data_range   = -458994 1374641
hips_creation_date = 2023-07-09T07:50Z
hips_estsize      = 34557261536
hips_nb_tiles     = 33571099
hips_check_code   = fits:3098902299

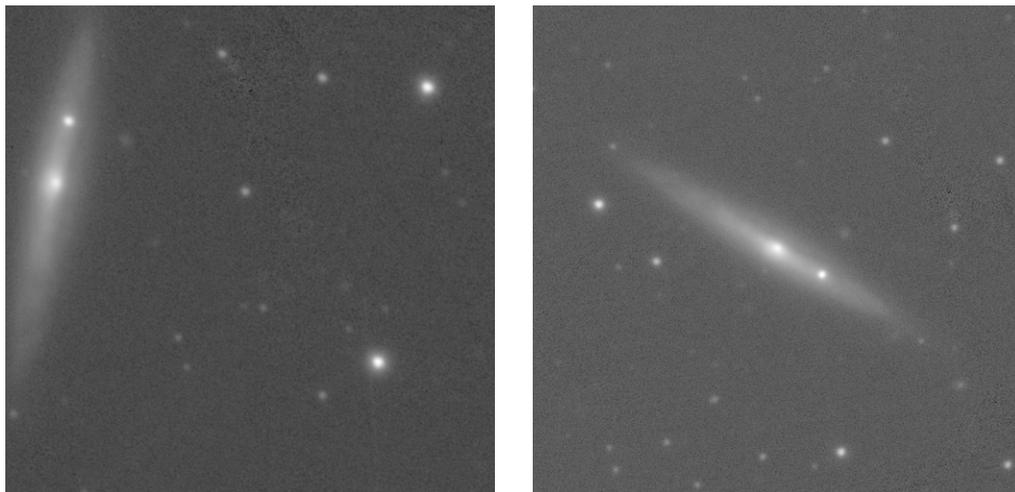
```

Figura 4.2: Algunas de las propiedades importantes del servidor local HiPS creado para el proyecto ALerCE.

Capítulo 5

Generación de Imágenes Multi Resolución

Dado que las imágenes HiPS generadas en el servidor local se encuentran orientadas en determinadas direcciones y centralizadas en ciertas coordenadas celestes acorde a la estructura de datos, la probabilidad de que tales coordenadas y tal orientación sean las deseadas por DELIGHT es bastante baja. Es decir, dado que la implementación actual del algoritmo DELIGHT requiere que la imagen entrante esté centralizada en las coordenadas a estudiar, los archivos HiPS en general no serán suficientes para ser ingresados como entrada del algoritmo. Tal hecho se ejemplifica en la Figura 5.1. Además, dado que las imágenes multi resolución que genera DELIGHT también están centralizadas en las coordenadas de interés, los archivos HiPS a distintos órdenes tampoco podrán ser utilizados directamente para una eventual entrega de las imágenes multi resolución. Debido a lo anterior, es que se requiere un procesamiento de las imágenes HiPS para poder crear en base a ellas una imagen centralizada en las coordenadas celestes de interés.



(a) Ejemplo de imagen HiPS a mayor resolución que contiene la coordenada (322.740027, 13.986060). Obtenida directamente desde servidor local HiPS.

(b) Ejemplo de imagen entrante para algoritmo DELIGHT, la cual es obtenida con coordenadas centralizadas (322.740027, 13.986060). Descargada desde servidor *Panstamps*.

Figura 5.1: Comparación de imagen HiPS v/s imagen Panstamps.

Es importante recordar en este punto que el algoritmo DELIGHT requiere de dos mejoras: La primera, es obtener la imagen de resolución simple que utiliza hoy el algoritmo como entrada de una manera mucho más rápida, ya que en este momento DELIGHT recupera tal imagen del servidor *Panstamps* a una velocidad de descarga promedio de 4 segundos por imagen. La segunda mejora, se refiere a la generación de imágenes multi resolución desde un principio para que estas sean la entrada del algoritmo DELIGHT, pues como se mencionó en Sección 2.1, la implementación actual del algoritmo descarga imágenes de resolución simple para luego obtener de ellas las imágenes multi resolución requeridas. Si bien la segunda mejora es el objetivo del trabajo de memoria, la primera es importante de realizar tanto para entender la generación de imágenes a través del servidor HiPS, como para presentar una mejora inmediata a la implementación actual del algoritmo DELIGHT.

Para lograr el procesamiento requerido en los archivos HiPS se utilizará el lenguaje *Python*. Tal elección recae netamente en la existencia de librerías escritas en tal lenguaje las cuales serán de bastante utilidad en las distintas actividades a realizar por el algoritmo deseado. En primer lugar, se necesitará relacionar cada pixel de la imagen a retornar con las coordenadas celestes que ellos representan. Luego, se requiere el manejo de la fórmula HEALPix para lograr obtener los archivos HiPS que cubren la parte del cielo pedida y poder calcular en tiempo constante sus ubicaciones en el servidor HiPS. Por último, y de una forma similar a la primera actividad, se debe relacionar cada pixel de los archivos HiPS necesarios para el procesamiento con las coordenadas celestes que ellos representan.

La relación entre pixeles y coordenadas celestes se establece utilizando el subpaquete *astropy.wcs* de la librería *astropy*, la cual es una librería de *Python* de acceso público que permite la interoperabilidad de distintos paquetes astronómicos [15]. Por su parte, el sistema mundial de coordenadas (WCS por sus siglas en inglés) es un sistema que permite definir la relación entre los pixeles de una imagen astronómica y las coordenadas celestes que ellos representan [15]. Así el subpaquete *wcs* de *astropy* contiene las herramientas necesarias para establecer las relaciones entre pixel y ubicación del cielo. Por otro lado, los archivos HiPS los cuales contienen a las coordenadas celestes de interés junto con sus ubicaciones serán obtenidos mediante las herramientas del módulo *healpy* el cual fue descrito en Sección 4.1.

Así, se propone la escritura de dos algoritmos para proponer con ellos las mejoras buscadas en DELIGHT. El primero de los algoritmos, será programado para lograr obtener mediante el servidor HiPS local las imágenes de resolución simple que requiere la implementación actual del algoritmo DELIGHT. Luego, y utilizando metodologías similares a las empleadas en el primero, se buscará generar un algoritmo el cual, recibiendo las coordenadas celestes de interés, genere mediante el servidor HiPS las imágenes multi resolución que DELIGHT crea a partir de la imagen de resolución simple. El objetivo de lo anterior es en primer lugar reducir de manera considerable el tiempo que demora DELIGHT en obtener las imágenes de resolución simple que requiere, para luego buscar entregarle al algoritmo DELIGHT las imágenes multi resolución de manera eficiente. Así, se podrán cumplir con las mejoras mencionadas anteriormente.

5.1. Generación de Imagen de Resolución Simple

Como ya se ha mencionado en este Capítulo, el primer algoritmo a escribir debe ser capaz de generar las imágenes de resolución simple que DELIGHT recibe en su implementación actual utilizando los archivos HiPS del servidor HiPS local. Entrando más en detalle sobre

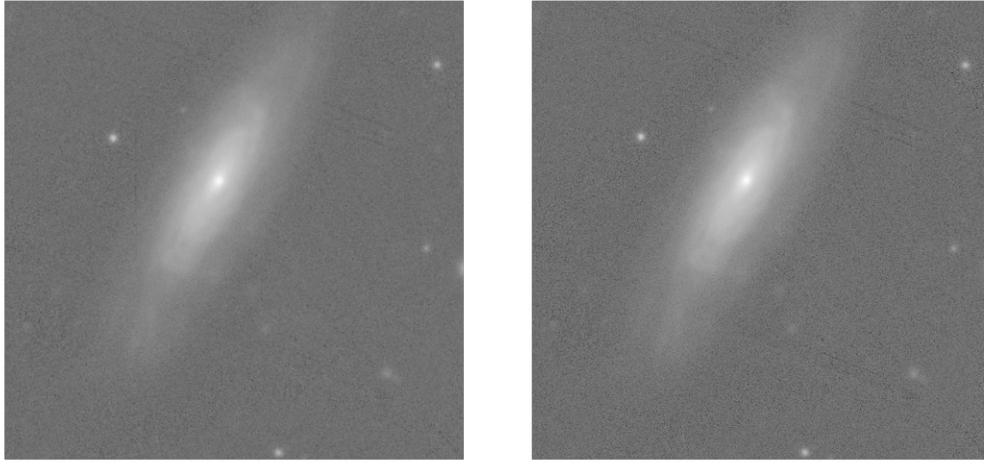
la imagen que DELIGHT descarga a través del servidor *Panstamps*, se tiene que tal imagen tiene un tamaño de 480×480 pixeles, con una resolución de $0.25''/\text{pixel}$ [2]. Por lo tanto, el algoritmo a generar debe tener como objetivo poder retornar una imagen del cielo con tales características, y lógicamente centralizada en las coordenadas celestes de interés. La idea principal es que esta imagen se genere rápidamente mediante el servidor HiPS utilizando los tiempos de consulta eficientes que entrega la estructura HiPS hacia sus archivos esenciales.

Para entender de mejor manera el proceso a realizar en el algoritmo deseado, se estudió *HiPS2FITS*, el cual es un servicio que genera recortes de imágenes en extensión *.fits* utilizando archivos de un servidor HiPS [16]. Así, se pueden identificar pasos claves que deben ser cumplidos por el algoritmo a escribir para lograr generar las imágenes de resolución simple buscadas.

Se idea así el siguiente algoritmo para generar una imagen de resolución simple. El algoritmo debe recibir las coordenadas celestes que serán el centro de la imagen, así como también el tamaño en pixeles de la imagen:

1. Definir las propiedades importantes de la imagen a retornar, así como las coordenadas celestes centrales de la imagen, su resolución por pixel, y su tamaño en pixeles. Todo esto definido en un *fits header* dado que se desea retornar la imagen en extensión *.fits*.
2. Para cada pixel (x, y) de la imagen definida, calcular las coordenadas celestes (α, δ) que le corresponden. Para realizar tal acción, se utiliza el subpaquete *astropy.wcs* de la librería *astropy*.
3. Utilizando las coordenadas celestes centrales (α_c, δ_c) , consultar y abrir el archivo HiPS de orden más profundo (orden 11) que contiene a (α_c, δ_c) . Lo anterior, mediante las herramientas del módulo *healpy*.
4. Mediante las funciones ofrecidas por el módulo *healpy* y el resultado del paso anterior, obtener y abrir los 8 archivos HiPS de orden 11 los cuales representan los vecinos N, S, E, O, NE, NO, SE y SO de la imagen HiPS obtenida en paso 3.
5. Para cada (α, δ) obtenido en paso 2, obtener la imagen HiPS de orden más profundo que contiene a tal coordenada mediante el módulo *healpy*, y su respectivo pixel el cual es más cercano a (α, δ) utilizando el subpaquete *astropy.wcs*. Abrir tal archivo HiPS en caso de que no haya sido obtenido en paso anterior.
6. Para cada pixel (x, y) de la imagen a retornar, establecer su valor según el valor del pixel más cercano a (α, δ) obtenido en paso anterior, donde (α, δ) son las coordenadas celestes de (x, y) calculadas en paso 2.

El último paso señalado ha sido optimizado en términos de tiempo utilizando decoradores *Numba*, el cual es un compilador de código abierto para *Python* el cual transforma el código escrito en este lenguaje a un código en lenguaje de máquina optimizado, consiguiendo desempeños similares a como si el código estuviera escrito en *C* o *C++* [17]. La razón por la cual sólo se utiliza la herramienta *Numba* en el último paso recae en el hecho de que *Numba* fue diseñado para trabajar sobre funciones y arreglos de la librería *Numpy* [17]. Por lo tanto, toda la información obtenida hasta antes del último paso debe ser almacenada en arreglos de la librería *Numpy* para así lograr optar a las optimizaciones de tiempo ofrecidas por *Numba*.



(a) Ejemplo de imagen de resolución simple generada por el algoritmo recién descrito, recibiendo como entrada las coordenadas (179.613540, 10.017920), una resolución de $0.25''/\text{pixel}$, y un tamaño de imagen de 480 píxeles.

(b) Ejemplo de imagen entrante para algoritmo DELIGHT, la cual es obtenida con coordenadas centralizadas (179.613540, 10.017920). Descargada desde servidor *Panstamps*.

Figura 5.2: Comparación de imagen de resolución simple generada v/s imagen *Panstamps* de las mismas características.

Es importante mencionar que el algoritmo recién descrito difiere del presentado por el servicio *HiPS2FITS*. Por ejemplo, la obtención de los archivos HiPS a utilizar por parte del algoritmo de *HiPS2FITS* no se realiza de la forma descrita en los pasos 3 y 4 del algoritmo presentado, junto con que el cálculo de los valores para los píxeles (x, y) se realiza mediante el método de interpolación bilineal [16]. Lo importante rescatado del algoritmo de *HiPS2FITS* recae netamente en los pasos 1 y 2, rutinas las cuales son necesarias para relacionar los píxeles de la imagen objetivo con las coordenadas celestes que ellos representan.

El algoritmo presentado cumple de buena manera con la generación de imágenes de resolución simple con una resolución de $0.25''/\text{pixel}$ y de tamaño 480×480 píxeles. Por ejemplo, en la Figura 5.2 se puede ver la comparación de una imagen obtenida desde el servidor *Panstamps* con una imagen generada por el algoritmo recién descrito. Tal como es visible, las imágenes mostradas no tienen diferencia alguna a simple vista, corroborando así que el algoritmo escrito por el estudiante genera una imagen que, además de cumplir con la resolución y tamaño definidos en un principio, también logra representar de buena manera las partes del cielo de interés. Más ejemplos de lo anterior son visibles en Sección A.1 del apartado de Anexos.

5.2. Generación de Imágenes de Resolución Múltiple

Teniendo definido el primer algoritmo objetivo, se puede proceder a escribir el algoritmo el cual se encargará de generar las imágenes multi resolución que DELIGHT requiere para realizar sus predicciones. La implementación actual del algoritmo DELIGHT genera un conjunto de n imágenes multi resolución a partir de la imagen de resolución simple de tamaño 480×480 píxeles con resolución de $0.25''/\text{pixel}$ descargada del servidor *Panstamps* [2]. Las imágenes multi resolución generadas son del mismo tamaño en píxeles y de distintas resoluciones por

pixel, donde tales parámetros N y r se definen en función de n como:

$$\begin{aligned} N &= \frac{480}{2^{n-1}} \\ r &= \frac{4''}{2^{i-1}} \end{aligned} \tag{5.1}$$

Donde $i \in [1...n]$, con $i = 1$ definiendo a la imagen de menor resolución por pixel, e $i = n$ haciendo referencia a la imagen de mayor resolución por pixel [2]. Las exploraciones sobre los hiper-parámetros de la red neuronal convolucional presente en el algoritmo DELIGHT, arrojan a $n = 5$ como parámetro óptimo [2].

Así, el segundo algoritmo a escribir debe tener la capacidad de generar 5 imágenes multi resolución centralizadas en las coordenadas celestes de interés, con resoluciones de 0.25"/pixel, 0.5"/pixel, 1"/pixel, 2"/pixel y 4"/pixel, y con un tamaño de 30×30 pixeles para las 5 imágenes. En este punto se aprovechará el hecho mencionado en Sección 2.5 de que el acceso a distintos niveles de profundidad de un servidor HiPS no interfiere en el tiempo de consulta de estos mismos. Para determinar los órdenes de profundidad a utilizar en la generación de las imágenes multi resolución, se utiliza un principio similar al utilizado por el algoritmo *Hipsgen* para calcular los archivos HiPS desde las imágenes entrantes, el cual asocia la resolución por pixel de las imágenes entrantes con el primer orden de profundidad el cual se identifique con una resolución por pixel inferior a la resolución de las imágenes [9].

Se pueden entonces establecer las relaciones entre resolución por pixel y orden HiPS para la generación de las imágenes multi resolución las cuales son visibles en la Tabla 5.1. Teniendo lo anterior en cuenta, se puede idear una forma similar al algoritmo ya descrito que cumpla con los nuevos requerimientos. En específico, el único detalle a tener en cuenta es la utilización de distintos órdenes HiPS de profundidad según la resolución por pixel de la imagen a generar, pues el primer algoritmo escrito utiliza sólo el orden más profundo al tener como objetivo generar imágenes de resolución simple de 0.25"/pixel.

Tabla 5.1: Órdenes de profundidad de los archivos HiPS a utilizar para el cálculo de las imágenes a distintas resoluciones por pixel, junto con la resolución por pixel que identifican a tales órdenes.

r_I	orden	r_H
0.25"/pixel	11	0.2013"/pixel
0.5"/pixel	10	0.4026"/pixel
1"/pixel	9	0.8052"/pixel
2"/pixel	8	1.6104"/pixel
4"/pixel	7	3.2208"/pixel

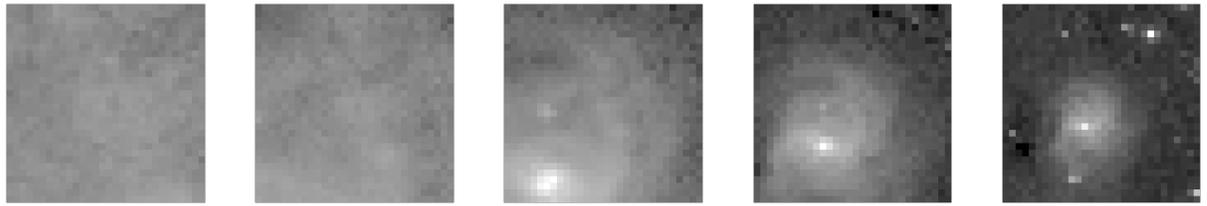
Gracias a las metodologías empleadas en la escritura del algoritmo, el código es fácilmente extensible para que se reciba como input al orden asociado a la resolución por pixel deseada además de las coordenadas celestes de interés y del tamaño en pixeles de la imagen. Por lo tanto, más que un nuevo algoritmo se debe actualizar el algoritmo anteriormente escrito para poder esta vez generar las imágenes multi resolución que el algoritmo DELIGHT requiere

para realizar sus predicciones. Así, se pueden listar los pasos de la versión actualizada del algoritmo de la siguiente manera:

1. Obtener la resolución por pixel asociada al orden ω de profundidad HiPS recibido.
2. Definir las propiedades importantes de la imagen a retornar, así como las coordenadas celestes centrales de la imagen, su resolución por pixel, y su tamaño en pixeles. Todo esto definido en un *fits header* dado que se desea retornar la imagen en extensión *.fits*.
3. Para cada pixel (x, y) de la imagen definida, calcular las coordenadas celestes (α, δ) que le corresponden. Para realizar tal acción, se utiliza el subpaquete *astropy.wcs* de la librería *astropy*.
4. Utilizando las coordenadas celestes centrales (α_c, δ_c) , consultar y abrir el archivo HiPS de orden ω que contiene a (α_c, δ_c) . Lo anterior, mediante las herramientas del módulo *healpy*.
5. Mediante las funciones ofrecidas por el módulo *healpy* y el resultado del paso anterior, obtener y abrir los 8 archivos HiPS de profundidad ω los cuales representan los vecinos N, S, E, O, NE, NO, SE y SO de la imagen HiPS obtenida en paso 3.
6. Para cada (α, δ) obtenido en paso 3, obtener la imagen HiPS de orden ω que contiene a tal coordenada mediante el módulo *healpy*, y su respectivo pixel el cual es más cercano a (α, δ) utilizando el subpaquete *astropy.wcs*. Abrir tal archivo HiPS en caso de que no haya sido obtenido en paso anterior.
7. Para cada pixel (x, y) de la imagen a retornar, establecer su valor según el valor del pixel más cercano a (α, δ) obtenido en paso anterior, donde (α, δ) son las coordenadas celestes de (x, y) calculadas en paso 2.

Nuevamente, el algoritmo generado cumple con los objetivos propuestos. En esta ocasión los cambios realizados en el algoritmo logran generar las imágenes multi resolución de tamaño 30×30 pixeles cumpliendo a su vez cada imagen con la resolución por pixel buscada desde un principio. Así, recibiendo las coordenadas celestes de interés, el algoritmo entrega el conjunto de 5 imágenes correspondientes a las imágenes que DELIGHT obtendría a partir de la imagen de resolución $0.25''/\text{pixel}$ con un tamaño de 480×480 pixeles centralizada en las mismas coordenadas ya mencionadas.

El desempeño del algoritmo recién descrito en términos de las imágenes generadas es visible en la Figura 5.3, donde se muestra una imagen de resolución $0.25''/\text{pixel}$ con un tamaño de 480×480 pixeles junto con su conjunto de 5 imágenes multi resolución correspondiente. Cada imagen de resolución múltiple cuenta con un tamaño de 30×30 pixeles con las resoluciones por pixel ya mencionadas. Más ejemplos de este resultado son mostrados en Sección A.2 del apartado de Anexos.



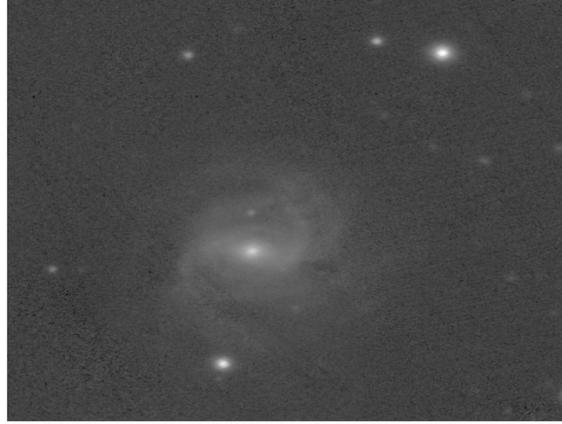
(a) $r=0.25''/\text{pixel}$.

(b) $r=0.5''/\text{pixel}$.

(c) $r=1''/\text{pixel}$.

(d) $r=2''/\text{pixel}$.

(e) $r=4''/\text{pixel}$.



(f) $r=0.25''/\text{pixel}$. $N=480$ pixeles.

Figura 5.3: Ejemplo de imágenes multi resolución. En la parte superior se muestran las 5 imágenes multi resolución de tamaño 30×30 pixeles centralizadas en las coordenadas celestes (36.549250, 27.606694). En la parte inferior, se muestra la imagen de resolución simple que debiese tener como input la implementación actual de *DELIGHT* para generar las imágenes de la parte superior.

Capítulo 6

Evaluación

Una vez que se cuenta con el algoritmo necesario para obtener las imágenes multi resolución que genera la implementación actual de DELIGHT, es importante evaluar el costo de tiempo de este mismo. Lo anterior se realiza recordando el hecho de que el proyecto ALERCE propone como objetivo obtener 10,000 conjuntos de 5 imágenes multi resolución en aproximadamente 40 segundos.

En primer lugar se evalúa el tiempo que toma el algoritmo descrito para generar las imágenes de resolución simple las cuales requiere DELIGHT para crear las imágenes de resolución múltiple. Lo anterior se realiza buscando reducir considerablemente el tiempo de 4 segundos para la obtención de imágenes *PanSTARRS* las cuales la implementación actual del algoritmo DELIGHT descarga a través del servidor *Panstamps*. Luego, se mide el tiempo del algoritmo para generar conjuntos de 5 imágenes multi resolución buscando cumplir los objetivos propuestos por el proyecto ALERCE.

6.1. Imágenes de Resolución Simple

Para evaluar el tiempo de generación de una imagen de resolución simple de $0.25''/\text{pixel}$ con un tamaño de 480×480 pixeles, se obtendrá el promedio de tiempo que demora el algoritmo descrito en retornar N imágenes de tales características, con N un número entero tal que $N \gg 1$. Así, se obtendrá una muestra aleatoria simple X_1, \dots, X_N de variables aleatorias independientes e idénticamente distribuidas, donde X_i denota el tiempo de demora de la generación de la imagen i -ésima. Cada imagen es obtenida utilizando un par de coordenadas celestes aleatorias $(\alpha, \delta)_i$, donde $i \in [1, \dots, N]$, $\alpha \in [0^\circ, 360^\circ]$, y $\delta \in [0^\circ, 90^\circ]$. Los valores de α y δ son establecidos para asegurar obtener imágenes de partes del cielo cubiertas por el servidor HiPS de ALERCE. Con esto, se buscará aproximar el valor esperado $\mu = \mathbb{E}(X_i)$ mediante el estimador:

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^N X_i. \quad (6.1)$$

Es importante señalar que el estimador de la ecuación (6.1) es consistente. En otras palabras, se cumple que:

$$\begin{aligned} \mathbb{E}(\bar{\mu}) &= \mu \\ \lim_{N \rightarrow \infty} \text{var}(\bar{\mu}) &= \lim_{N \rightarrow \infty} \text{var}(\bar{X}) = 0 \end{aligned} \tag{6.2}$$

Donde $\text{var}(\bar{\mu})$ representa la varianza del estimador. Con estos argumentos se puede pasar a medir $\bar{\mu} = \bar{X}$, para así obtener una estimación del tiempo esperado en la generación de una imagen de resolución simple.

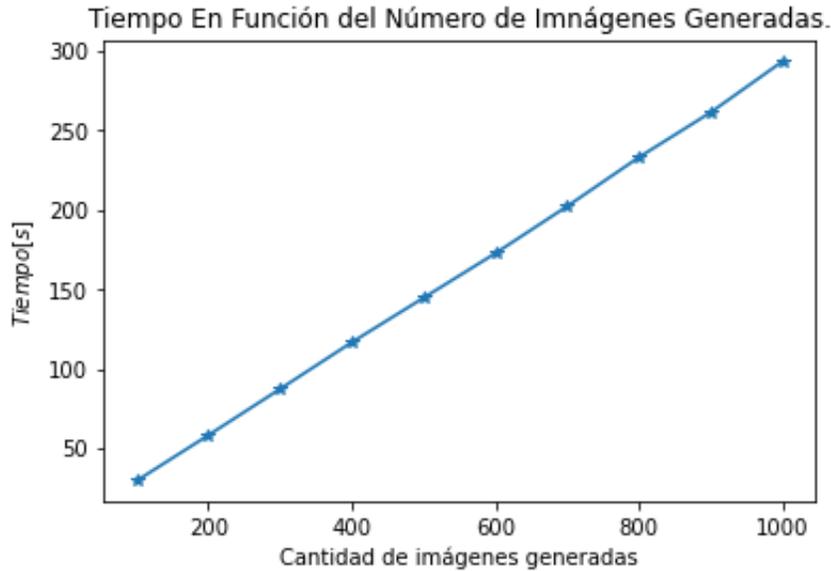


Figura 6.1: Tiempo en segundos en función del número de imágenes de resolución simple generadas. Cada imagen generada cuenta con una resolución de $0.25''/\text{pixel}$ y un tamaño de 480×480 píxeles.

Se establece así el valor de $N = 1000$ para realizar las mediciones correspondientes. Por su parte, es importante destacar que el comportamiento de \bar{X} parece no depender de la cantidad de imágenes generadas a partir de $N = 100$ tal como lo indica la Figura 6.1. Tomando así el tiempo de demora en la generación de 1,000 imágenes, se llega a que el tiempo esperado en la generación de una imagen de resolución simple de $0.25''/\text{pixel}$ con un tamaño de 480×480 píxeles se estima como:

$$\mu \approx \bar{\mu} = \frac{1}{N} \sum_{i=1}^N X_i = \frac{293.565[s]}{1000} = 293.565[ms]. \tag{6.3}$$

Dado que el tiempo que utiliza DELIGHT en descargar una imagen de este tipo mediante el servidor *Panstamp*s es de aproximadamente $4[s] = 4000[ms]$, se concluye que el algoritmo creado por el estudiante entregaría las imágenes de una forma la cual es aproximadamente 13 veces más rápida.

6.2. Imágenes de Resolución Múltiple

De una manera similar a la recién descrita, para evaluar el tiempo de generación de un conjunto de 5 imágenes multi resolución de tamaño 30×30 píxeles y con las resoluciones

definidas anteriormente, se obtendrá el promedio que demora el algoritmo en entregar N conjuntos de 5 imágenes multi resolución de tales características, con $N \gg 1$. Se tendrá como objetivo obtener el mismo estimador definido en (6.1), pero con la consideración de que ahora X_i denota el tiempo utilizado en la generación del conjunto i -ésimo de 5 imágenes multi resolución. Es importante notar que las coordenadas celestes $(\alpha, \delta)_i$ del i -ésimo conjunto de 5 imágenes de resolución múltiple son obtenidas de forma aleatoria con los mismos intervalos de valores ya mencionados en el caso anterior.

Estableciendo nuevamente el valor de $N = 1000$, se llega a que el tiempo esperado en la generación de un conjunto de 5 imágenes de resolución múltiple de tamaño 30×30 pixeles y con resoluciones $0.25''/\text{pixel}$, $0.5''/\text{pixel}$, $1''/\text{pixel}$, $2''/\text{pixel}$ y $4''/\text{pixel}$, se aproxima como:

$$\mu \approx \bar{\mu} = \frac{1}{N} \sum_{i=1}^N X_i = \frac{195.482[s]}{1000} = 195.482[ms]. \quad (6.4)$$

Teniendo en cuenta de que el proyecto ALERCE tiene como meta generar los conjuntos de 5 imágenes multi resolución en un tiempo aproximado de $4[ms]$ por conjunto, se puede ver que el tiempo promedio en la generación de las 5 imágenes para el caso del algoritmo del estudiante no cumple tal objetivo.

6.3. Mejoras Realizadas

Tal como se indicó en el final de Sección 6.2, el tiempo obtenido para la generación de un conjunto de 5 imágenes de resolución múltiple con las características que exige el algoritmo DELIGHT es lejano al rendimiento buscado. Con ánimos de lograr reducir el tiempo de ejecución del algoritmo creado, se realizó un análisis sobre cada paso de este mismo con el fin de verificar si se realiza alguna actividad innecesaria. En efecto, la apertura de archivos HiPS se realiza de forma ineficiente. El algoritmo abre el archivo HiPS que contiene la coordenada celeste recibida y sus 8 vecinos, y luego verifica si existe algún archivo faltante necesario para el cálculo de los pixeles de la imagen a retornar. Sin embargo nada afirma que los 8 archivos vecinos vayan a ser consultados en los pasos donde se calcula el pixel más cercano para cada coordenada celeste de interés. Lo que realmente se debiese hacer es que los archivos HiPS vayan siendo abiertos a medida que son necesarios en los cálculos, de una manera similar a la realizada en el caso de la apertura de la imagen HiPS faltante.

Así, se puede establecer una nueva versión del algoritmo incluyendo la mejora descrita recientemente. Este algoritmo supone ser el único y oficial como resultado del trabajo de memoria en lo que a generación de imágenes se refiere, y sus pasos se pueden listar de la siguiente manera:

1. Obtener la resolución por pixel asociada al orden ω de profundidad HiPS recibido.
2. Definir las propiedades importantes de la imagen a retornar, así como las coordenadas celestes centrales de la imagen, su resolución por pixel, y su tamaño en pixeles. Todo esto definido en un *fits header* dado que se desea retornar la imagen en extensión *.fits*.
3. Para cada pixel (x, y) de la imagen definida, calcular las coordenadas celestes (α, δ) que le corresponden. Para realizar tal acción, se utiliza el subpaquete *astropy.wcs* de la librería *astropy*.

4. Para cada (α, δ) obtenido en paso 3, obtener la imagen HiPS de orden ω que contiene a tal par de coordenadas mediante el módulo *healpy* y abrirla en caso de que no haya sido consultada anteriormente. Obtener a su vez el pixel más cercano a (α, δ) utilizando el subpaquete *astropy.wcs*.
5. Para cada pixel (x, y) de la imagen a retornar, establecer su valor según el valor del pixel más cercano a (α, δ) obtenido en paso anterior, donde (α, δ) son las coordenadas celestes de (x, y) calculadas en paso 2.

Se procedió entonces a realizar las mismas evaluaciones señaladas pero utilizando la versión actualizada del algoritmo ideado por el estudiante. Es importante mencionar que se espera que la mejora realizada se refleje de mayor manera en la generación de imágenes multi resolución en comparación a la de imágenes de resolución simple debido al tamaño en pixeles. Claramente una imagen de 480×480 pixeles requerirá una cobertura mucho más amplia del cielo que una imagen de 30×30 pixeles a igual resolución. Con un cálculo simple se ve que sólo la imagen de resolución $4''/\text{pixel}$ de tamaño 30×30 pixeles cubre la misma parte del cielo que la imagen de resolución $0.25''/\text{pixel}$ de tamaño 480×480 pixeles. Todas las demás imágenes multi resolución con resoluciones menores necesitarán una porción inferior del cielo. Además, es inmediato el hecho de que una imagen de tamaño 30×30 pixeles se genera de una forma más rápida que una de tamaño 480×480 pixeles.

6.3.1. Imágenes de Resolución Simple

Las nuevas mediciones para el tiempo utilizado en la generación de 1000 imágenes de resolución simple con resolución $0.25''/\text{pixel}$ y tamaño 480×480 pixeles entregan un estimador $\bar{\mu}$ con un valor de:

$$\mu \approx \bar{\mu} = \frac{1}{N} \sum_{i=1}^N X_i = \frac{215.319[s]}{1000} = 215.319[ms]. \quad (6.5)$$

Por lo tanto, se obtiene como resultado final que el algoritmo ideado por el estudiante entrega las imágenes de una forma que es aproximadamente 19 veces más rápida que la descarga de imágenes del servidor *Panstamps* por parte del algoritmo DELIGHT. La Tabla 6.1 muestra el avance en la reducción del tiempo para la obtención de una imagen de resolución simple de $0.25''/\text{pixel}$ y tamaño 480×480 pixeles.

Tabla 6.1: Reducción de tiempo para la obtención de imágenes *PanSTARRS*.

Algoritmo	Tiempo promedio[ms]
Panstamps	4000
Algoritmo sin mejora	293.565
Algoritmo con mejora	215.319

6.3.2. Imágenes de Resolución Múltiple

Para el caso de la generación de un conjunto de 5 imágenes multi resolución, las nuevas mediciones realizadas teniendo en consideración los cambios realizados al algoritmo arrojan

un tiempo promedio de:

$$\mu \approx \bar{\mu} = \frac{1}{N} \sum_{i=1}^N X_i = \frac{38.112[s]}{1000} = 37.765[ms]. \quad (6.6)$$

El aumento en el rendimiento del algoritmo mejorado para el caso de imágenes multi resolución con un tamaño de 30×30 pixeles es notable. El tiempo promedio por conjunto de 5 imágenes paso de ser $195[ms]$ a $38[ms]$ lo que evidencia una mejora importante en los accesos a disco para leer los distintos archivos HiPS. Este hecho se argumenta por la razón de que el algoritmo ideado sufrió modificaciones sólo en lo que a la lectura de imágenes HiPS se refiere. Si bien el tiempo obtenido no es igual a $4[ms]$, está bastante más cerca a cumplir los propósitos del proyecto ALerCE.

En este punto es importante recalcar el hecho de que todas las pruebas realizadas hasta este momento son de carácter secuencial. Al requerir un leve mejor rendimiento del algoritmo descrito por el estudiante en términos de tiempo de ejecución, es lógico pensar en como funcionaría este mismo utilizando paralelización. Dado que se han optimizado los accesos a disco por parte del algoritmo para leer los archivos HiPS, se espera que la ejecución en paralelo del algoritmo no tenga impactos significativos en los costos de tiempo de este mismo. Lo anterior se debe principalmente a que la apertura de archivos HiPS es la única instrucción por parte del algoritmo que realiza operaciones de escritura o lectura. Todas las demás partes del algoritmo se reflejan en cálculos de CPU los cuales pueden realizarse simultáneamente.

Así, se realiza una última evaluación la cual tiene como objetivo medir como escala el tiempo promedio en la generación de un conjunto de 5 imágenes multi resolución utilizando ejecuciones en paralelo del algoritmo. En primer lugar se realizó la medición de 2 ejecuciones en paralelo del algoritmo obteniendo un tiempo promedio de:

$$\bar{\mu}_2 = 38.561[ms]. \quad (6.7)$$

Donde $\bar{\mu}_2$ representa el tiempo máximo entre los tiempos de las 2 ejecuciones en paralelo. Realizando el mismo experimento con 3 ejecuciones del algoritmo se obtiene un tiempo de:

$$\bar{\mu}_3 = 39.117[ms]. \quad (6.8)$$

Donde $\bar{\mu}_3$ representa el tiempo máximo entre los tiempos de las 3 ejecuciones en paralelo. Comparando las ecuaciones (6.8) y (6.7) con el resultado (6.6) podemos inferir que la paralelización del algoritmo funciona de forma esperada, escalando aproximadamente $0.5[ms]$ en su tiempo de ejecución por rutina paralela agregada. Tal hecho se puede reafirmar observando la Tabla 6.2 donde se indica el tiempo obtenido para la evaluación en paralelo de N ejecuciones en paralelo, con $N \in [1, 6]$. Así, se espera que la paralelización del algoritmo ideado por el estudiante, por ejemplo con 10 ejecuciones en paralelo, si logre cumplir los costos de tiempo buscados por el proyecto ALerCE.

Tabla 6.2: Tiempo promedio máximo entre N ejecuciones en paralelo para la generación de 1,000 conjuntos de 5 imágenes multi resolución.

N	$\mu_N [ms]$
1	37.765
2	38.561
3	39.117
4	39.685
5	40.121
6	40.387

Capítulo 7

Conclusiones

El presente trabajo de título tuvo como enfoque realizar las distintas actividades necesarias para levantar una infraestructura digital la cual permita el almacenamiento y consulta rápida de grandes volúmenes de imágenes multi resolución astronómicas. Tales imágenes multi resolución deben contar con características específicas exigidas por la implementación actual del algoritmo DELIGHT.

Para lograr lo anterior se diseñó y desarrolló un servidor local utilizando el esquema HiPS para el almacenamiento de distintas partes del cielo a distintas resoluciones. A su vez, se ideó el algoritmo necesario para generar las imágenes de resolución múltiple que el algoritmo DELIGHT requiere para sus predicciones. Teniendo así acceso a un servidor HiPS local se logra implementar la rutina necesaria para generar las imágenes de resolución múltiple deseadas. Esta rutina significa un procesamiento en las imágenes primordiales de un servidor HiPS el cual tendrá como resultado imágenes centralizadas en las coordenadas celestes indicadas con tamaños en pixeles y resoluciones por pixel acordes a los necesarios por DELIGHT.

En primer lugar hablando sobre el análisis realizado al servidor *Panstamps* se concluye que tal servicio no es una forma eficiente de consultar las imágenes de resolución simple que utiliza la implementación actual del algoritmo DELIGHT. Así mismo se demuestra que el servidor es incapaz de entregar imágenes de resolución múltiple como las que DELIGHT requiere para realizar sus predicciones.

En cuanto a la creación de un servidor local HiPS se puede mencionar que se logra adquirir el conocimiento básico y necesario para poder generar un servidor de este tipo a partir de imágenes astronómicas de resolución simple. Se aprende a manipular las herramientas existentes para la generación de servidores HiPS a partir de imágenes entrantes y se conoce el uso de la librería *healpy* para recuperar los archivos HiPS de interés. Por su parte, se aprendió la escritura de rutinas para la descarga de imágenes astronómicas pertenecientes a la misión *PanSTARRS PS1* logrando descargar las partes del cielo que cubren todo el hemisferio norte de la esfera celeste.

Respecto al algoritmo ideado se puede concluir que se lleva a cabo de buena manera la rutina para generar imágenes astronómicas a distintas resoluciones centralizadas en las coordenadas celestes deseadas, logrando entregar estas imágenes de forma rápida. Además de lo anterior el algoritmo es capaz de entregar una imagen de resolución simple con resolución de $0.25''/\text{pixel}$ y de tamaño 480×480 pixeles la cual usaría la implementación actual del algoritmo DELIGHT para generar el conjunto de imágenes multi resolución, de una forma

mucho más eficiente que el rendimiento otorgado por la conexión hacia el servidor *Panstamps*.

7.1. Recuento de Objetivos

En cuanto a los objetivos del trabajo de memoria, se puede mencionar en primer lugar que se cumple de cierta forma el objetivo general del proyecto el cual pide un servidor que logre entregar el conjunto deseado de 5 imágenes multi resolución en aproximadamente 4[ms]. Si bien una única ejecución del algoritmo creado por el estudiante no es suficiente para cumplir con las metas establecidas, se demostró que ejecuciones en paralelo del mismo podrían eventualmente cumplir con las exigencias del proyecto ALerCE.

Refiriéndose sobre los objetivos específicos del trabajo se puede concluir en primer lugar que el análisis del servidor *Panstamps* fue exitoso, pudiendo demostrar que tal servidor no puede ser utilizado para cumplir con los propósitos exigidos por el algoritmo DELIGHT. De manera seguida se concluye que el estudio de estructuras de datos eficientes para el almacenamiento y consulta de imágenes multi resolución astronómicas fue realizado de buena manera, encontrando así el esquema HiPS el cual tuvo una relevancia importante en el trabajo de memoria. Por último, se puede concluir que se logró de manera exitosa crear un servidor HiPS local y complementarlo con rutinas las cuales permitan el procesamiento necesario de los archivos HiPS primordiales. El algoritmo logra la entrega de imágenes multi resolución de forma rápida, con un tiempo promedio cercano al propuesto desde un principio.

Es importante destacar que los resultados encontrados se lograron en gran parte gracias a las exigencias de tiempo que impone el proyecto ALerCE para la entrega de imágenes multi resolución. La búsqueda de la reducción de tiempo en la ejecución del algoritmo ideado permitió encontrar las mejoras necesarias en las primeras versiones del algoritmo para lograr una entrega eficaz y rápida de las imágenes de resolución múltiple. A su vez es importante hacer mención de las distintas herramientas disponibles en distintas librerías de *Python* las que permitieron realizar toda actividad necesaria en el algoritmo sin costos de tiempo provocados por el uso de tal lenguaje.

7.2. Impacto del Trabajo Realizado

El trabajo realizado en la memoria supone el punto de partida para la generación de imágenes de resolución múltiple para la identificación de galaxias en tiempo real. Antes del proyecto establecido el algoritmo DELIGHT no contaba con ninguna forma para obtener directamente las imágenes multi resolución que requiere para realizar sus predicciones. A su vez, la implementación actual de DELIGHT recupera la imagen de resolución simple que necesita para la generación de imágenes multi resolución en un tiempo el cual es 19 veces el tiempo que entrega el algoritmo ideado en la memoria. Por estos dos argumentos es que se concluye que el trabajo realizado tiene un impacto fuerte en los propósitos del proyecto ALerCE, pues en primer lugar se realiza una mejora importante al funcionamiento actual del algoritmo DELIGHT, y en segundo lugar se entrega una alternativa para levantar una infraestructura digital que permita la entrega de imágenes multi resolución de forma eficiente.

7.3. Trabajo Futuro

El trabajo futuro contemplado está fuertemente relacionado con el impacto recién comentado que tiene el trabajo hecho en la memoria. Lo realizado claramente abre la línea de investigación para buscar cumplir con los tiempos deseados por el proyecto ALERCE para el acceso de 10 mil conjuntos de 5 imágenes multi resolución. Así, se piensan en las siguientes alternativas que representan un trabajo futuro sobre el resultado obtenido en el presente trabajo de título:

1. Utilizar el servidor HiPS creado junto con el algoritmo ideado para entregar la imagen de resolución simple que requiere la implementación actual del algoritmo DELIGHT, y comparar el rendimiento con el obtenido al descargar imágenes a través del servidor *Panstamps*.
2. Modificar el algoritmo DELIGHT para que su entrada sea el conjunto de 5 imágenes de resolución múltiple que entrega el algoritmo ideado en la memoria. Evaluar los resultados proporcionados por DELIGHT y verificar si las predicciones resultantes son las mismas a las obtenidas con la implementación actual de DELIGHT.
3. Realizar un análisis avanzado al algoritmo escrito en el trabajo de título e intentar reducir los costos de tiempo de este mismo.
4. Revisar el funcionamiento de la librería *astropy.wcs* para leer archivos HiPS. Se descubre que la librería no logra procesar algunas imágenes del servidor local HiPS. Si bien lo anterior se encontró en un único archivo HiPS, idealmente todas las imágenes HiPS generadas deben ser posibles de procesar.
5. Generar una rutina que utilice paralelización para reducir los costos de tiempo del algoritmo ideado en la memoria.

7.4. Comentarios Finales

Como conclusiones finales, se puede decir en primer lugar que se cumplió en general con el objetivo propuesto, ya que se logra levantar una infraestructura digital para el rápido acceso a imágenes multi resolución astronómicas. Si bien el costo de tiempo del algoritmo ideado no cumple con el tiempo deseado por el proyecto ALERCE, está bastante cerca de serlo. Además se demostró que ejecuciones en paralelo del mismo eventualmente logran los tiempos exigidos.

Por otro lado, se puede concluir que el trabajo de memoria significó un aprendizaje considerable en el estudiante al poner en práctica principalmente las metodologías aprendidas en los cursos relacionados con la ingeniería de software. Si bien fue un desafío grande el organizar todo el trabajo de manera individual, las herramientas entregadas por tales cursos permitieron elaborar un plan de trabajo el cual resultó en su mayoría exitoso, permitiendo así entregar soluciones novedosas para resolver un problema interesante en el área de ciencias de la computación.

Bibliografía

- [1] Forster, F., “AleRCE - Automatic Learning for the Rapid Classification of Events,” 2019, <https://alerce.science/>.
- [2] Förster, F., Arancibia, A. M. M., Reyes-Jainaga, I., Gagliano, A., Britt, D., Cuellar-Carrillo, S., Figueroa-Tapia, F., Polzin, A., Yousef, Y., Arredondo, J., Rodríguez-Mancini, D., Correa-Orellana, J., Bayo, A., Bauer, F. E., Catelan, M., Cabrera-Vives, G., Dastidar, R., Estévez, P. A., Pignata, G., Hernández-García, L., Huijse, P., Reyes, E., Sánchez-Sáez, P., Ramírez, M., Grandón, D., Pineda-García, J., Chabour-Barra, F., y Silva-Farfán, J., “Delight: Deep learning identification of galaxy hosts of transients using multiresolution images,” *The Astronomical Journal*, vol. 164, 2022, [doi:10.3847/1538-3881/ac912a](https://doi.org/10.3847/1538-3881/ac912a).
- [3] Young, D. R., [doi:10.5281/zenodo.8037665](https://doi.org/10.5281/zenodo.8037665).
- [4] Tonry, J. L., Stubbs, C. W., Lykke, K. R., Doherty, P., Shivvers, I. S., Burgett, W. S., Chambers, K. C., Hodapp, K. W., Kaiser, N., Kudritzki, R. P., Magnier, E. A., Morgan, J. S., Price, P. A., y Wainscoat, R. J., “The pan-starrs1 photometric system,” *Astrophysical Journal*, vol. 750, 2012, [doi:10.1088/0004-637X/750/2/99](https://doi.org/10.1088/0004-637X/750/2/99).
- [5] Boch, T. y Fernique, P., “Creating and managing very large hips: The pan-starrs case,” *Astronomical Society of the Pacific Conference Series*, vol. 523, 2019.
- [6] Waters, C. Z., Magnier, E. A., Price, P. A., Chambers, K. C., Burgett, W. S., Draper, P. W., Flewelling, H. A., Hodapp, K. W., Huber, M. E., Jedicke, R., Kaiser, N., Kudritzki, R.-P., Lupton, R. H., Metcalfe, N., Rest, A., Sweeney, W. E., Tonry, J. L., Wainscoat, R. J., y Wood-Vasey, W. M., “Pan-starrs pixel processing: Detrending, warping, stacking,” *The Astrophysical Journal Supplement Series*, vol. 251, 2020, [doi:10.3847/1538-4365/abb82b](https://doi.org/10.3847/1538-4365/abb82b).
- [7] Flewelling, H. A., Magnier, E. A., Chambers, K. C., Heasley, J. N., Holmberg, C., Huber, M. E., Sweeney, W., Waters, C. Z., Calamida, A., Casertano, S., Chen, X., Farrow, D., Hasinger, G., Henderson, R., Long, K. S., Metcalfe, N., Narayan, G., Nieto-Santisteban, M. A., Norberg, P., Rest, A., Saglia, R. P., Szalay, A., Thakar, A. R., Tonry, J. L., Valenti, J., Werner, S., White, R., Denneau, L., Draper, P. W., Hodapp, K. W., Jedicke, R., Kaiser, N., Kudritzki, R. P., Price, P. A., Wainscoat, R. J., Chastel, S., McLean, B., Postman, M., y Shiao, B., “The pan-starrs1 database and data products,” *The Astrophysical Journal Supplement Series*, vol. 251, 2020, [doi:10.3847/1538-4365/abb82d](https://doi.org/10.3847/1538-4365/abb82d).
- [8] Boch, T., Fernique, P., Oberto, A., y Bonnarel, F., “Aladin - An Interactive Sky Atlas,” 2010, <http://aladin.cds.unistra.fr/aladin.gml>.
- [9] Pierre Fernique, Mark Allen, T. B. T. D. D. D. K. E. L. M. J. S. F. S., “HiPS - Hierarchical Progressive Survey. International Virtual Observatory Alliance Proposed Recom-

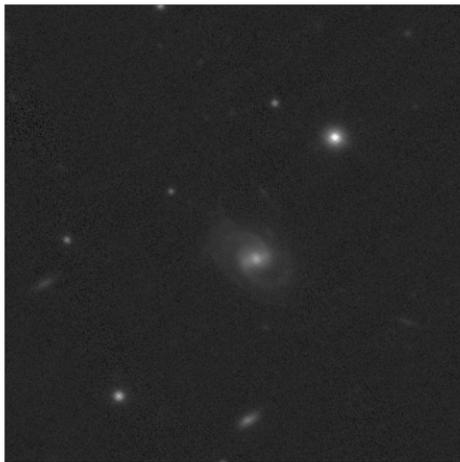
- mendation,” 2017, <https://www.ivoa.net/documents/HiPS/20170406>.
- [10] Gorski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., y Bartelmann, M., “Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere,” *The Astrophysical Journal*, vol. 622, 2005, doi:10.1086/427976.
- [11] Kunszt, P. Z., Szalay, A. S., y Thakar, A. R., “The hierarchical triangular mesh,” *Mining the Sky*, 2006, doi:10.1007/10849171_83.
- [12] O’Mullane, W., Banday, A. J., Górski, K. M., Kunszt, P., y Szalay, A. S., *Splitting the Sky - HTM and HEALPix*. 2006, doi:10.1007/10849171_84.
- [13] Zonca, A., Singer, L., Lenz, D., Reinecke, M., Rosset, C., Hivon, E., y Gorski, K., “healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in python,” *Journal of Open Source Software*, vol. 4, 2019, doi:10.21105/joss.01298.
- [14] Ginsburg, A., SipHocz, B. M., Brasseur, C. E., Cowperthwaite, P. S., Craig, M. W., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lim, P. L., Lockhart, K. E., Mommert, M., Morris, B. M., Norman, H., Parikh, M., Persson, M. V., Robitaille, T. P., Segovia, J.-C., Singer, L. P., Tollerud, E. J., de Val-Borro, M., Valtchanov, I., y Woillez, J., “astroquery : An astronomical web-querying package in python,” *The Astronomical Journal*, vol. 157, p. 98, 2019, doi:10.3847/1538-3881/aafc33.
- [15] Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., Günther, H. M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J. E., Singer, L., Fox, R., Weaver, B. A., Zabalza, V., Edwards, Z. I., Bostroem, K. A., Burke, D. J., Casey, A. R., Crawford, S. M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P. L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., y Streicher, O., “Astropy: A community python package for astronomy,” *Astronomy and Astrophysics*, vol. 558, 2013, doi:10.1051/0004-6361/201322068.
- [16] Boch, T., Fernique, P., Bonnarel, F., Chaitra, C., Bot, C., Pineau, F. X., Baumann, M., y Michel, L., “HiPS2FITS: Fast Generation of FITS Cutouts From HiPS Image Datasets,” en *Astronomical Data Analysis Software and Systems XXIX* (Pizzo, R., Deul, E. R., Mol, J. D., de Plaa, J., y Verkouter, H., eds.), vol. 527 de *Astronomical Society of the Pacific Conference Series*, p. 121, 2020.
- [17] “Numba: a high performance python compiler,” 2018, <https://numba.pydata.org/>.

Anexos

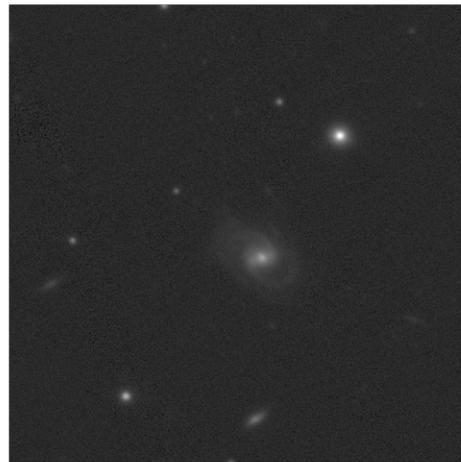
Anexo A. Ejemplos de Generación de Imágenes

A.1. Ejemplos de Generación de Imagen de Resolución Simple

A continuación es posible ver ejemplos de la generación de imágenes de resolución simple por parte del algoritmo ideado en el trabajo de memoria. Cada imagen de ejemplo es comparada con la imagen que obtendría la implementación actual del algoritmo DELIGHT a través del servidor *Panstamps*. La resolución de cada imagen es de $0.25''/\text{pixel}$ y cuentan con un tamaño de 480×480 píxeles.

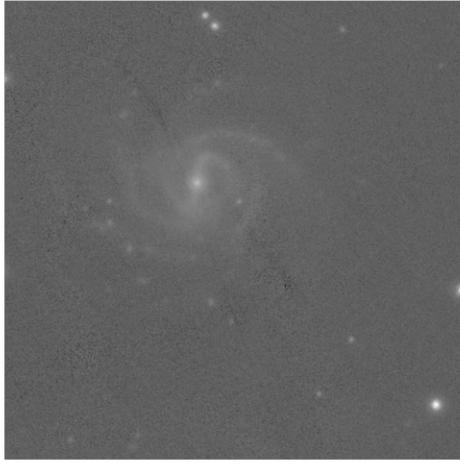


(a) Ejemplo de imagen de resolución simple generada.

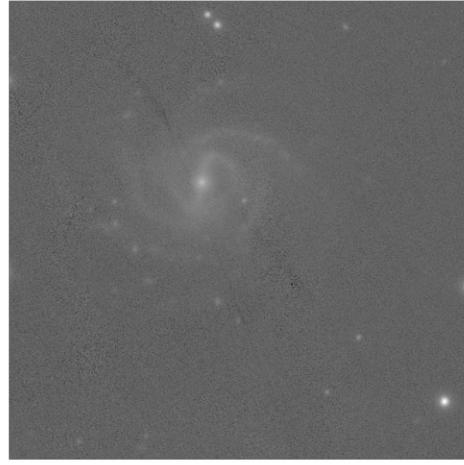


(b) Ejemplo de imagen entrante para algoritmo DELIGHT. Descargada desde servidor *Panstamps*.

Figura A.1: Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características. Coordenadas centrales (240.346160, 26.291187).

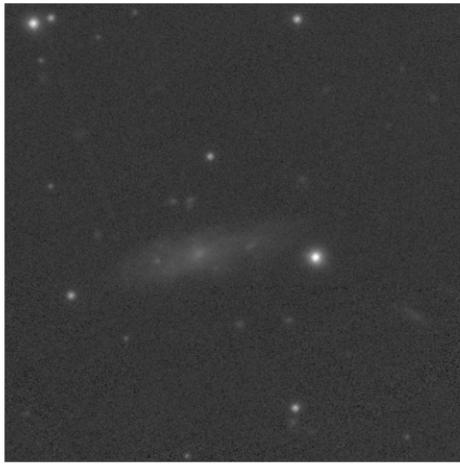


(a) Ejemplo de imagen de resolución simple generada.

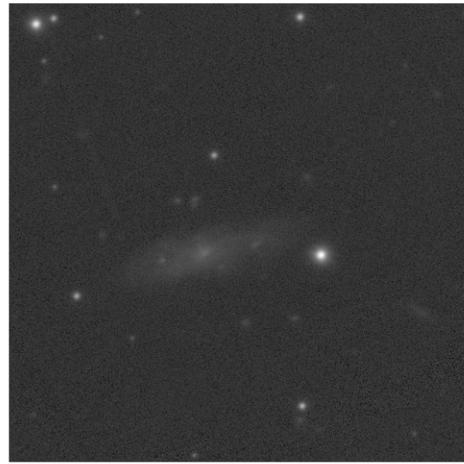


(b) Ejemplo de imagen entrante para algoritmo DELIGHT. Descargada desde servidor *Panstamps*.

Figura A.2: Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características. Coordenadas centrales (344.920080, 24.071530)



(a) Ejemplo de imagen de resolución simple generada.

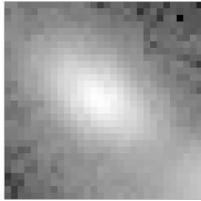


(b) Ejemplo de imagen entrante para algoritmo DELIGHT. Descargada desde servidor *Panstamps*.

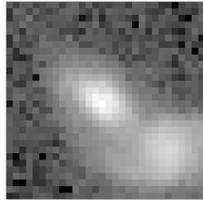
Figura A.3: Comparación de imagen de resolución simple generada v/s imagen Panstamps de las mismas características. Coordenadas centrales (60.866428, 19.181077).

A.2. Ejemplos de Generación de Imágenes de Resolución Múltiple

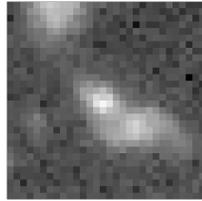
Las siguientes figuras denotan ejemplos de la generación de un conjunto de 5 imágenes multi resolución, junto con la imagen de resolución simple de resolución $0.25''/\text{pixel}$ y tamaño 480×480 pixeles asociada a tal conjunto. En la parte superior de cada figura se muestra el conjunto de 5 imágenes multi resolución junto con su resolución asociada. Cada una de ellas se caracteriza por un tamaño de 30×30 pixeles. En la parte inferior, se puede ver la imagen de resolución simple que utilizaría la implementación actual del algoritmo DELIGHT para generar el conjunto de imágenes de resolución múltiple indicado en parte superior.



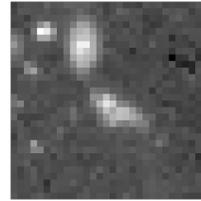
(a) $r=0.25''/\text{pixel}$.



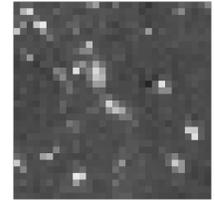
(b) $r=0.5''/\text{pixel}$.



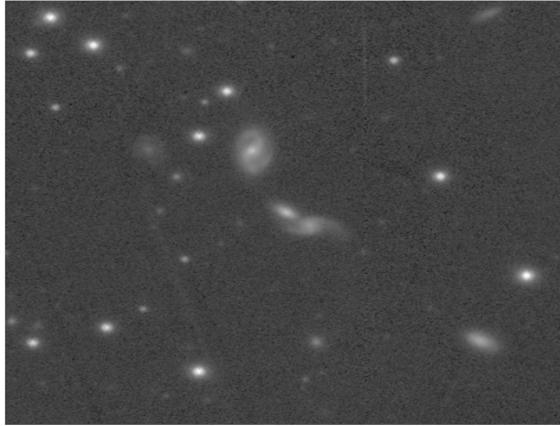
(c) $r=1''/\text{pixel}$.



(d) $r=2''/\text{pixel}$.

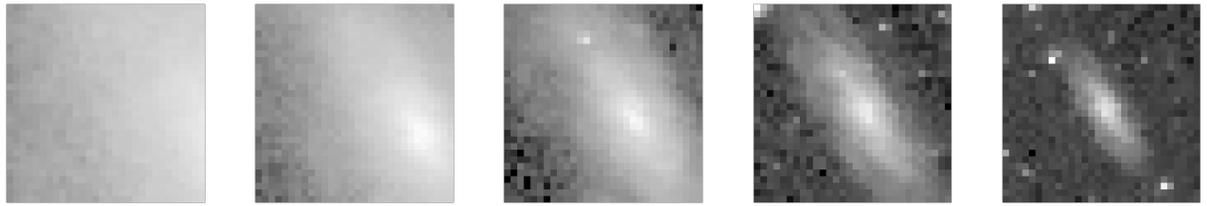


(e) $r=4''/\text{pixel}$.

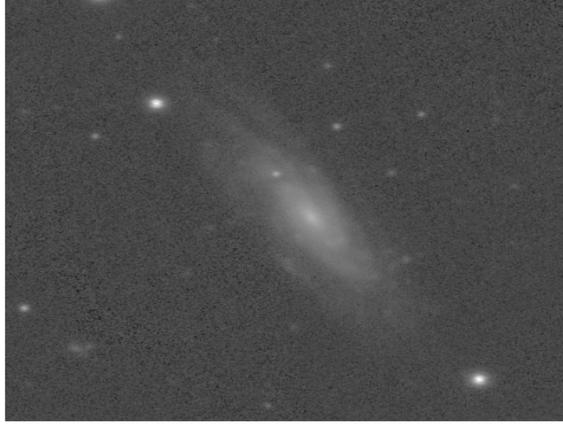


(f) $r=0.25''/\text{pixel}$. $N=480$ pixeles.

Figura A.4: Ejemplo de imágenes multi resolución. Cada imagen centralizada en las coordenadas celestes $(246.259991, 11.997976)$.

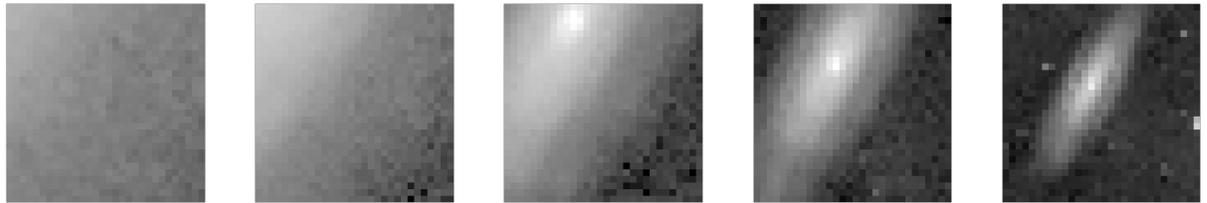


(a) $r=0.25''/\text{pixel}$. (b) $r=0.5''/\text{pixel}$. (c) $r=1''/\text{pixel}$. (d) $r=2''/\text{pixel}$. (e) $r=4''/\text{pixel}$.

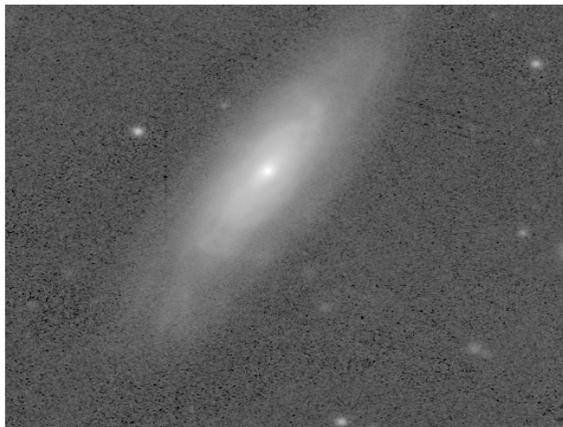


(f) $r=0.25''/\text{pixel}$. $N=480$ pixeles.

Figura A.5: Ejemplo de imágenes multi resolución. Cada imagen centralizada en las coordenadas celestes (16.541830, 31.406560).



(a) $r=0.25''/\text{pixel}$. (b) $r=0.5''/\text{pixel}$. (c) $r=1''/\text{pixel}$. (d) $r=2''/\text{pixel}$. (e) $r=4''/\text{pixel}$.



(f) $r=0.25''/\text{pixel}$. $N=480$ pixeles.

Figura A.6: Ejemplo de imágenes multi resolución. Cada imagen centralizada en las coordenadas celestes (179.613540, 10.017920).