

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Desarrollo de un repositorio de datos para el desarrollo de algoritmos en microscopía electrónica

**Autor: Unai Rubio Villate
Tutor: Carlos Oscar Sorzano Sanchez
Ponente: Roberto Marabini Ruiz**

Abril 2023

Agradecimientos

Agradecer a Carlos, tutor del proyecto por darme la oportunidad de realizar este Trabajo de Fin de Grado, por su gran ayuda y por abrirme las puertas a las instalaciones del Centro Nacional de Biotecnología (CNB) formado por un maravilloso grupo de trabajo.

En especial quiero mencionar a Daniel que ha estado acompañándome en este proyecto durante los últimos meses en los que cada vez que surgía una duda él estaba ahí para ayudar.

Por último, un último agradecimiento tanto a mi familia como a mis amigos que han sido un pilar fundamental durante toda mi vida.

Resumen

El Trabajo de Fin de Grado consiste en el desarrollo de un repositorio de datos para el análisis de microscopía electrónica.

Para llevar a cabo esta tarea, se ha utilizado el framework Scipion, una herramienta muy útil para la reconstrucción tridimensional de moléculas a partir de imágenes adquiridas en un microscopio electrónico. Gracias a este framework, se ha podido completar un protocolo que permite la recopilación de datos de manera sencilla y efectiva.

Una vez recopilados los datos, se procede a su procesamiento mediante una API creada en Django. Esta API permite la organización y tratamiento de los datos de manera eficiente, para que posteriormente puedan ser presentados de manera sencilla al usuario a través de una interfaz web.

Este repositorio de datos es una herramienta muy valiosa para la investigación en microscopía electrónica, ya que permite la obtención de información relevante de manera rápida y eficiente. Para garantizar la efectividad del repositorio, se han llevado a cabo una serie de pruebas en las que se ha verificado el resultado obtenido. De esta manera, se asegura que el repositorio cumple con los objetivos previstos y se garantiza su eficacia.

Palabras clave

Scipion, criomicroscopía electrónica, análisis partículas individuales, protocolo, flujo de trabajo, marco de trabajo, servidor, interfaz programación de aplicaciones, Django, Python

Abstract

This academic paper revolves around the development of a data repository for the analysis of electronic microscopy.

The framework Scipion is a very useful tool that has been used to accomplish this task, it allows us to make three-dimensional reconstructions of the images acquired from electronic microscopes. Thanks to this framework, it's been possible to create a protocol which can help us to collect data in a very simple and effective way.

After the data is collected, it is processed by an API created using Django. This API allows us to organise and treat this data in a very efficient way, so it can be presented to the user later in a very simple way through a web interface.

This data repository is a very useful tool for future investigations in electronic microscopy since it will allow the user to obtain relevant information very fast and efficiently. To guarantee the effectiveness of this repository, there have been several tests carried out where the result obtained has been indeed verified. This way, it is ensured that this repository will meet the intended goals and its efficiency is also guaranteed.

Keywords

Scipion, Cryogenic electron microscopy, single particle analysis, protocol, workflow, framework, server, application program interface, Django, Python

Índice

Agradecimientos.....	III
Resumen.....	V
Abstract.....	VII
Índice.....	IX
Índice figuras.....	XI
1. Introducción.....	1
1.1 Motivación.....	2
1.2 Objetivos.....	2
1.3 Estructura del proyecto.....	3
2. Estado del arte.....	5
2.1 Criomicroscopia electrónica.....	5
2.2 Historia.....	5
2.3 Ventaja y desventajas.....	6
2.4 Técnicas.....	7
3. Diseño.....	9
3.1 Introducción a scipion.....	9
3.1.1 Proceso reconstrucción 3D.....	11
3.2 Cliente:used ítems tracker.....	21
3.2.1 Generacion de resultados.....	22
3.2.2 Solicitud POST.....	24
3.3 Servidor Django.....	26
4. Implementación.....	29
5. Pruebas.....	29
6. Conclusiones y trabajo futuro.....	35
Bibliografía.....	36

Índice de figuras

Figura 2.1: Resolución 2013 vs Actualidad.....	6
Figura 2.2: Proceso SPA.....	8
Figura 3.1: Ventana principal de scipion.....	9
Figura 3.2: Flujo de procesamiento utilizado.....	10
Figura 3.3: Ventana de puglins.....	11
Figura 3.4: Proceso volumen 3D.....	12
Figura 3.5: Protocolo importar imagenes.....	14
Figura 3.6: Movie.....	15
Figura 3.7: Protocolo movie alignment.....	15
Figura 3.8: Resultado micrografías.....	16
Figura 3.9: Protocolo movie max shift.....	16
Figura 3.10: Protocolo estimación CTF.....	17
Figura 3.11: Estimacion de CTF.....	17
Figura 3.12: Protocolo extracción particulas.....	18
Figura 3.13: Entrenamiento particulas.....	19
Figura 3.14: Clasificación 2D. Izqda buena, dcha descartada.....	20
Figura 3.15: Volumen inicial.....	20
Figura 3.16: Protocolo Track used ítems Parte 1 de 2.....	21
Figura 3.17: Sets de imágenes generados por scipion.....	23
Figura 3.18: Imágenes separadas en categorías.....	24
Figura 3.19: Comunicación Cliente-Servidor.....	25
Figura 3.20: Estructura proyecto Django.....	29
Figura 5.1: Ventana de registro.....	30
Figura 5.2: Perfil del usuario.....	31
Figura 5.3: Protocolo Track used ítems Parte 2 de 2.....	32
Figura 5.4: Proyecto mostrado interfaz web.....	33
Figura 5.5: Categorías e imágenes de un proyecto.....	34

Introducción

La microscopía electrónica es una técnica ampliamente utilizada en la investigación científica para observar estructuras a nivel microscópico con una resolución mucho mayor que la que se puede obtener con un microscopio óptico. Esta técnica ha permitido importantes avances en muchas áreas de la biología, incluyendo la estructura de proteínas y otras biomoléculas, la caracterización de virus y células, y la observación de procesos celulares en vivo.

El desarrollo de algoritmos de análisis de imagen y procesamiento de datos puede aclarar la interpretación de los datos de microscopía electrónica. Para el desarrollo de algoritmos, es necesario contar con un repositorio de datos de alta calidad que contenga información detallada sobre los diferentes tipos de muestras, técnicas de microscopía utilizadas, así como las características y propiedades de las imágenes obtenidas.

Sin embargo, la recopilación y organización de los datos generados en este campo es un desafío. El procesamiento y análisis de las imágenes requieren de un software especializado, y los datos pueden estar dispersos en diferentes sistemas y formatos, lo que dificulta su acceso y uso por parte de la comunidad científica.

Para superar estos desafíos, se propone el desarrollo de un servidor de datos para el desarrollo de algoritmos en microscopía electrónica. Este repositorio permitiría recopilar, organizar y compartir datos de manera eficiente y garantizar la calidad de los datos generados.

Para su desarrollo, se utilizará el software Scipion ya que cuenta con numerosos protocolos que serán utilizados para extraer la información necesaria que será archivada posteriormente en el servidor. Esto permitirá tratar la información de manera eficiente y separar las muestras en etiquetas de si fueron utilizadas o no en la reconstrucción de partículas.

Para permitir una interacción más fluida y escalable con nuestro desarrollo web, se creará una API REST en Django que permitirá la recopilación y presentación de los datos en la aplicación web creada. Por último, se utilizará Apache para alojar la aplicación y pueda ser usada por más

1.1 Motivación

La motivación para el desarrollo de este repositorio de datos radica en la necesidad de mejorar la eficiencia y la calidad de la investigación científica en el campo de la microscopía electrónica. La recopilación, organización y validación de los datos son fundamentales para garantizar la calidad de los resultados obtenidos y para permitir el desarrollo de nuevas herramientas y técnicas de análisis.

Además, el acceso y la difusión de los datos son importantes para promover la colaboración y la innovación en el campo de la investigación científica. Un repositorio de datos bien estructurado y accesible puede facilitar la comparación de resultados entre diferentes investigadores y laboratorios, así como la validación de los hallazgos científicos por parte de la comunidad científica en general.

En este contexto, el uso de software especializado, como Scipion, puede mejorar significativamente la eficiencia y la calidad del procesamiento, así como también, la organización de los datos generados en los experimentos de microscopía electrónica. Además, el uso de un repositorio web basado en Django permitirá el almacenamiento y la presentación de los datos de manera clara y accesible para los usuarios, mientras que Apache asegurará la accesibilidad de la información en línea para cualquier usuario en cualquier lugar del mundo.

1.2 Objetivos

El objetivo principal de este proyecto es desarrollar un repositorio de datos para el análisis y la investigación en microscopía electrónica. Para lograrlo, se han establecido varios subobjetivos, que se detallan a continuación.

- Completar un protocolo en scipion que permite recopilar los datos de una reconstrucción tridimensional de la macromolécula que está bajo estudio para que pueda separarlos en etiquetas que indiquen si fueron utilizados o no en la investigación usando las relaciones que tienen los protocolos entre sí. Una vez separados y etiquetados, el mismo protocolo contará con las herramientas necesarias para establecer una comunicación con el servidor y enviar los resultados donde serán procesados y almacenados.
- Creación de un REST framework en Django que permita recoger y tratar toda la información que se envía desde scipion. Para ello, se aprovecharán las funciones y herramientas que proporciona este framework para trabajar con datos y crear modelos que permitan su organización y almacenamiento en una base de datos. De esta manera, se garantiza una gestión adecuada y ordenada de la información recibida.

- Lanzamiento del proyecto Django en un servidor Apache para que la aplicación pueda ser accesible al público en remoto. Esto se hace para asegurar que los usuarios puedan acceder a la información de manera rápida y segura, proporcionando un entorno seguro y estable donde los datos estén protegidos.

1.3 Estructura proyecto

El Trabajo de fin de grado consta de seis capítulos, a saber:

Capítulo 1. Introducción. Incluye una preparación al tema a tratar, junto a la motivación y la estructura del proyecto.

Capítulo 2. Estado del arte. Se introducen conceptos sobre la tecnología, junto con un breve contexto histórico y sus características.

Capítulo 3. Diseño. Explicación de Scipion y sus pasos principales. Elaboración de los datos enviados por el cliente y creación del repositorio.

Capítulo 4. Implementación. Lanzamiento del servidor al público

Capítulo 5. Pruebas. Apartado de explicación de como visualizar los resultados en la interfaz actuando como usuario.

Capítulo 6. Conclusiones y trabajo futuro. Explicación breve de los resultados obtenidos y una opinión de posibles mejoras.

2. Estado del arte

2.1. Criomicroscopía Electrónica

La criomicroscopía electrónica (CryoEM) es una técnica que utiliza microscopios electrónicos con la finalidad de estudiar la estructura molecular de las muestras biológicas a alta resolución y en tres dimensiones. Para ello, la muestra se congela rápidamente y se mantiene a temperaturas muy bajas (alrededor de los -180°C) para evitar la formación de cristales y la degradación de la muestra. Una vez que la muestra está lista se coloca en el microscopio y se le aplica un haz de electrones para obtener una imagen en 2D o 3D de su estructura. La técnica de criomicroscopía electrónica ha permitido a los investigadores obtener imágenes detalladas de macromoléculas biológicas, como proteínas y virus, y ha contribuido significativamente a nuestra comprensión de la estructura y función de estas moléculas en la biología molecular. [1]

2.2. Historia

CryoEM es una técnica relativamente nueva, que se ha desarrollado en las últimas décadas, aunque se basa en la técnica de microscopía electrónica creada en la década de 1930. Los primeros microscopios electrónicos eran capaces de obtener imágenes de alta resolución de secciones de muestras que, aunque contaban con ciertas limitaciones en la tecnología, permitió a los investigadores estudiar la estructura de las células [2]. Con el paso de los años fue mejorando la tecnología y con ello, se consiguieron mejores resultados.

El primer paso importante se dio en los años 70 cuando se desarrolló una técnica que permitía la congelación rápida de las muestras biológicas sin que se formasen cristales de hielo que pudieran dañar la estructura de la muestra. En los años posteriores se consiguió implementar una técnica de reconstrucción de imágenes, que permitía la reconstrucción de una estructura en 3D a partir de imágenes bidimensionales. [2]

En la década de 1990, se hizo un gran avance porque hasta el momento se creía que solo se podía usar los microscopios electrónicos para materia que estuviese muerta debido al haz de electrones que se usaba. La resolución mejoró significativamente gracias al avance tecnológico que permitió introducir detectores de alta velocidad en la captación de imágenes.

En 2013 se consiguió la resolución atómica que se buscaba Figura 2.1 gracias a una optimización de las piezas del microscopio lo que provocó en 2017 se otorgase el Premio Nobel de Química a esta técnica. [3]

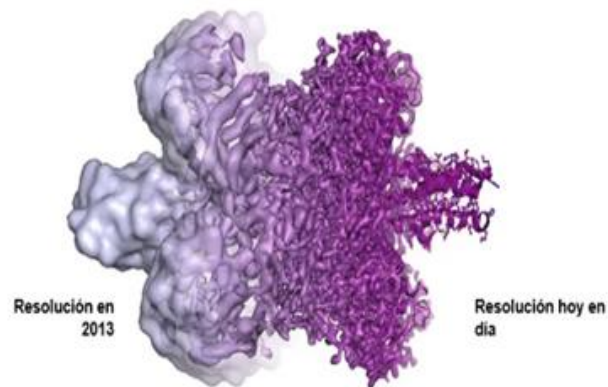


Figura 2.1: Resolución 2013 vs Actualidad

2.3. Ventajas y desventajas

En cuanto a las ventajas, es una técnica no invasiva ya que permite la observación de biomoléculas en su estado nativo, sin necesidad de incluir procesamientos químicos, cristalizaciones no exposiciones radiactivas, que puedan llegar a originar una alteración en la estructura o funcionalidad de las muestras [4]. Otra ventaja es que permite estudiar estructuras biológicas grandes y complejas, como las proteínas, en tres dimensiones y a alta resolución, lo que permite visualizar detalles estructurales que no llegan a ser visibles con otras técnicas [5].

Por el contrario, entre sus desventajas, destaca que requiere una gran cantidad de recursos informáticos para su funcionamiento de las imágenes y por tanto un proceso caro.

2.4. Técnicas

Algunas de las técnicas que incluye cryo-Em son:

- **EM de tinción negativa.** Se utiliza para contrastar las muestras biológicas, como proteínas y virus. La muestra se cubre con una solución de tinción negativa que rodea las partículas, lo que hace que aparezcan más oscuras en el fondo claro del microscopio [6].
- **Grabado por congelación EM.** Esta técnica se usa para visualizar muestras biológicas congelando rápidamente la muestra. El proceso de grabado permite visualizar la estructura de las moléculas y complejos de proteínas en su estado nativo [6].
- **Análisis de partículas individuales (SPA).** Esta técnica se utiliza para determinar la estructura tridimensional de proteínas y otros complejos biomoleculares. Se basa en la adquisición de múltiples imágenes de partículas individuales desde diferentes ángulos y orientaciones, para luego reconstruir una imagen 3D [6].
- **Servicios de cristalografía electrónica.** Esta técnica se utiliza para determinar la estructura de moléculas y macromoléculas complejas como proteínas a partir de cristales. La muestra se cristaliza y luego se irradia con electrones, que interactúan con los átomos de la muestra y producen patrones de difracción que se pueden analizar para determinar la estructura de la muestra [6].
- **Difracción de electrones de microcristales.** Esta técnica se utiliza para examinar estructuras cristalinas a nivel molecular. Los microcristales se obtienen y se exponen a un haz de electrones para obtener un patrón de difracción de electrones. A continuación, la estructura cristalina se puede determinar a partir del patrón de difracción [6].

De las técnicas vistas en el apartado vamos a centrarnos en explicar brevemente los pasos necesarios para el proceso de análisis de partículas individuales mostrados en la Figura 2.2.

En la preparación de la muestra para SPA, es crucial la purificación previa de la muestra, ya que esto asegura que solo se estén estudiando las partículas de interés. Una vez preparada la muestra, se procede a la congelación rápida a temperaturas muy bajas utilizando un dispositivo de congelación especializado.

Una vez la muestra está congelada se utiliza un microscopio electrónico de transmisión de criomicroscopía electrónica para adquirir imágenes de la muestra. A continuación, se seleccionan las partículas individuales para el análisis y se clasifican en diferentes clases en función de su forma y características. Esto permite identificar partículas con características similares, es decir, partículas que comparten la misma orientación de la muestra, lo que provoca una mejor resolución en las proyecciones de la muestra. Una vez clasificadas las partículas, se realiza una reconstrucción tridimensional para obtener una imagen 3D e identificar de forma más sencilla las partículas de interés [7].

Extraído de <https://www.creative-biostructure.com/single-particle-analysis-spa.htm>

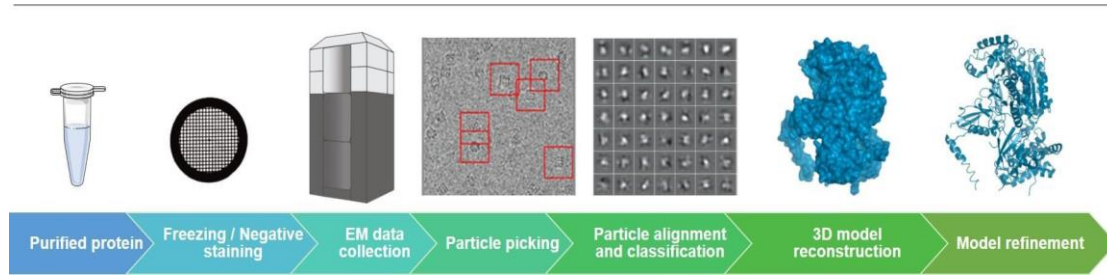


Figura 2.2: Proceso SPA

Cabe señalar que, por regla general, alguno de los pasos explicados se repetirá varias veces hasta conseguir el resultado deseado ya que se trata de un proceso complejo que requiere de experiencia previa y conocimientos técnicos.

Los resultados obtenidos son de una gran calidad, pero pueden verse influidos por la calidad de la muestra observada y la configuración del microscopio.

3. Diseño

3.1 Introduccion a scipion

Scipion es un framework de procesamiento de imágenes que trabaja para obtener modelos 2D y 3D de macromoléculas. Para ello, cuenta con una interfaz que permite crear flujos de trabajo automatizados combinando las herramientas de software que tiene. Es muy útil ya que guarda los resultados de cada uno de los pasos ejecutados y puede ser visualizado en cualquier momento [8].

Los proyectos se construyen utilizando protocolos, que son el conjunto de pasos o instrucciones que describen cómo procesar los datos. Los protocolos son piezas esenciales de Scipion ya que permiten reproducir los resultados y asegurar la calidad de los datos procesados. Además, Scipion también ofrece la posibilidad de crear flujos de trabajo o workflows que son una combinación de varios protocolos, permitiendo automatizar y simplificar aún más el proceso de análisis de datos.

De esta manera, Scipion se convierte en una herramienta poderosa y eficiente para el procesamiento de datos de criomicroscopía electrónica.

Una vez arrancamos el programa se abre una interfaz como en la Figura 3.1 donde se mostrarán todos los proyectos que se hayan creado.

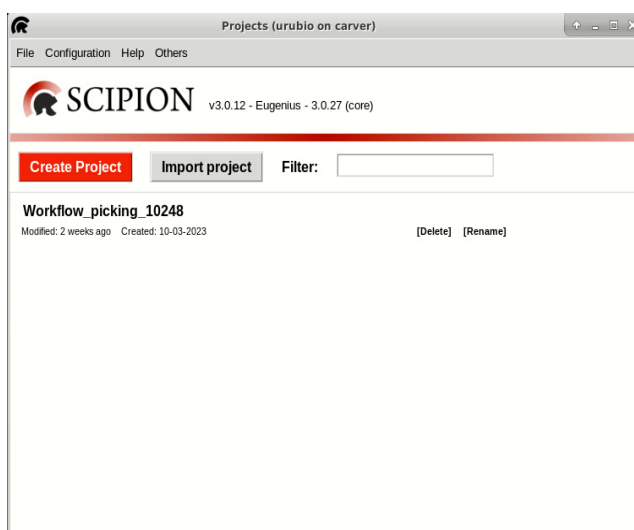


Figura 3.1: Ventana principal de Scipion

Si se selecciona el proyecto *workflow_picking_10248* se mostrará en la Figura 3.2 el ejemplo que contiene el flujo de procesamiento de reconstrucción 3D utilizado para la realización del proyecto y que será explicado más adelante. Cabe destacar que es posible que un mismo proyecto contenga varios flujos con sus respectivas pruebas y datos, ya que al ser independientes no interfieren entre sí.

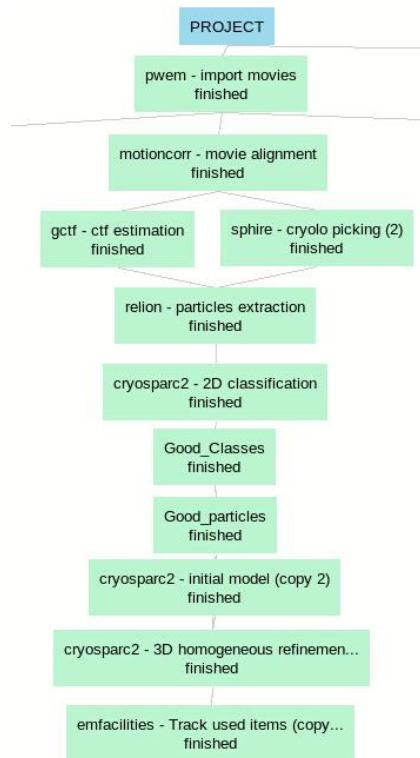


Figura 3.2: Flujo de procesamiento utilizado

Esto permite a los usuarios ver y editar la configuración de cada flujo. Además, también se pueden visualizar los resultados que genera cada uno de los protocolos, así como las imágenes procesadas y los modelos generados, en diferentes vistas 2D y 3D.

Cabe destacar que para la realización de los protocolos será necesario descargar su correspondiente plugin. Básicamente un plugin es un componente que integra software externo, en este caso dentro de Scipion, para añadir nuevas funcionalidades al programa [9]. Para ello, se accederá a la pestaña *others* de la Figura 3.1 para posteriormente entrar en la ventana de *Plugin manager* y descargar los que sean necesarios como en la Figura 3.3.

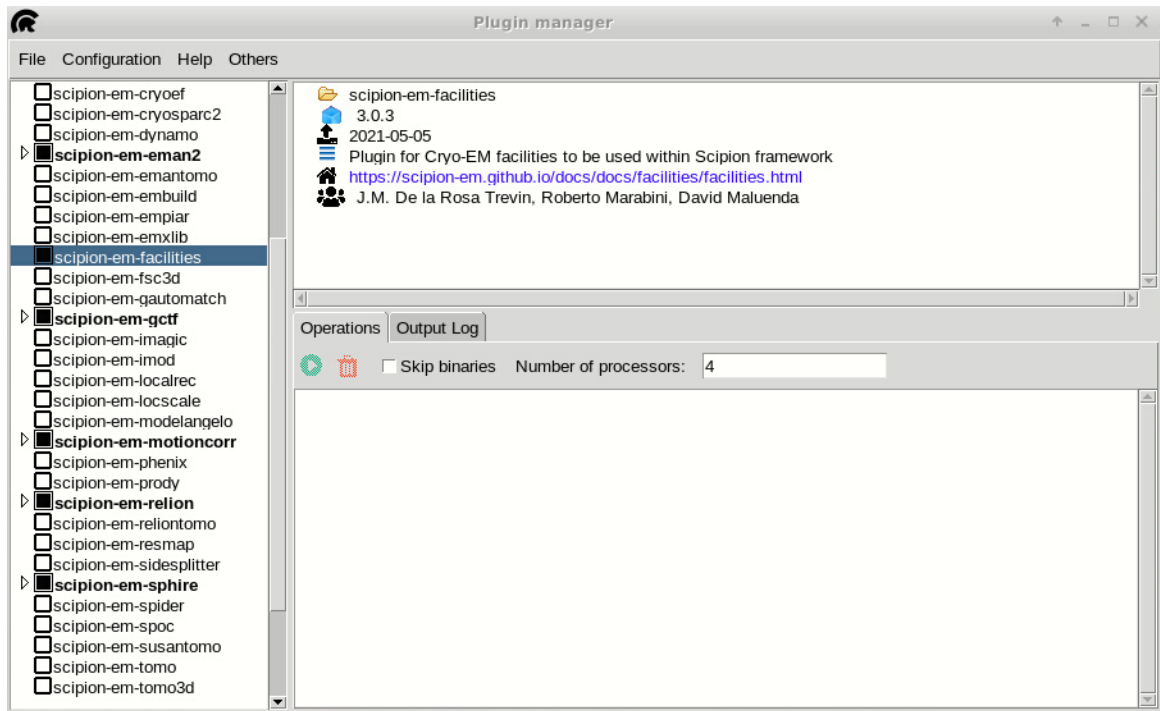


Figura 3.3: Ventana de puglins

3.1.1. Procesos reconstrucción 3D

El camino básico en un proyecto de Scipion comienza importando las imágenes en bruto (movies) hasta conseguir el volumen en 3D deseado y consta de varios pasos intermedios principales, Figura 3.4. Para entender el proceso se explicará en Scipion tanto su funcionamiento, como los datos que recibe y que genera:

Extraído de https://scipion-em.github.io/docs/release-3.0.0/static/pdfs/tutorial_scipion3_apoferritin.pdf

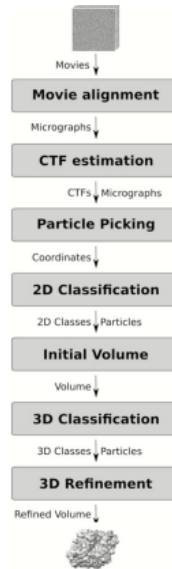


Figura 3.4: Proceso Volumen 3D

- **Importar imágenes.** Este es el primer paso en scipion, donde se permite a los usuarios cargar las imágenes generadas por el microscopio, revisar y seleccionar las que se usarán en el proyecto. A esas imágenes se les llama movies.
- **Alineamiento de movies.** En este paso, se corrige el movimiento de las partículas en el momento que son capturadas por el microscopio, alineando los frames de las movies para conseguir una mejora en la calidad de la muestra [10]. Para ello se utiliza el plugin *MotionCorr*.
- **Estimación de la función de transferencia de contraste (CTF).** Con las micrografías obtenidas en el paso anterior se estima la CTF para cada una de ellas. Este proceso implica estimar la distorsión que introduce el microscopio en el momento de capturar las imágenes y corregir aberraciones como ruido a nivel de partículas de la micrografía [10]. Se suele usar Ctfind4, Gctf, CryoSPARC para este proceso.
- **Extracción de partículas:** En este paso, las numerosas partículas que conforman cada micrografía se extraen para trabajar sobre ellas posteriormente. Este proceso puede ser automático o manual y se realiza mediante herramientas de selección que permiten seleccionar cuales son las partículas de interés como por ejemplo Relion, Gautomatch [10].

- **Clasificación 2D:** Una vez se ha hecho la extracción, se clasifican en 2D para identificar subconjuntos de partículas similares en función de sus características y se descartan aquellas que no sean verdaderamente partículas (ruido) [10]. Para agrupar las partículas en clases el algoritmo de clasificación más usado es CryoSPARC.
- **Volumen inicial:** En este paso, se genera un volumen inicial aproximado mediante CryoSPARC utilizando las partículas seleccionadas en el paso anterior. Este volumen inicial se utiliza como modelo para la siguiente etapa de clasificación 3D.
- **Clasificación 3D:** Una vez ya se conoce un volumen inicial es posible estimar una clasificación en 3D de las partículas usadas según sus características al igual que se hacía en 2D. Los algoritmos usados en este caso son Relion, CryoSPARC.
- **Refinamiento 3D.** Para finalizar, se realiza un refinamiento sobre el volumen inicial donde se vuelven a clasificar de forma iterativa las partículas para obtener una mejor clasificación de ellas, y por tanto una mejor resolución en el volumen 3D [10]. Al igual que los pasos anteriores se sigue utilizando el protocolo CryoSPARC.

Es importante mencionar que existen otros programas disponibles para realizar cada uno de los pasos anteriores y la elección de cada uno podrá depender de los conocimientos y experiencia del usuario, requisitos de hardware y la calidad de los datos.

Además, hay que destacar que hay pasos intermedios que se pueden añadir a los anteriores donde se incorporan otros tipos de cálculos según el tipo de datos que se quieran conseguir.

Para comprender mejor el funcionamiento de los pasos anteriores se explicarán en el entorno de Scipion viendo cuando sea posible como es la interfaz de los protocolos, y los resultados que generan.

Importar imágenes

Esta ventana es la primera que se muestra al comenzar el proceso de importación de imágenes. Podremos indicar el directorio donde se encuentran, las cuales han de estar en formato *TIFF* o *MRC* como se muestra en la figura 3.5.

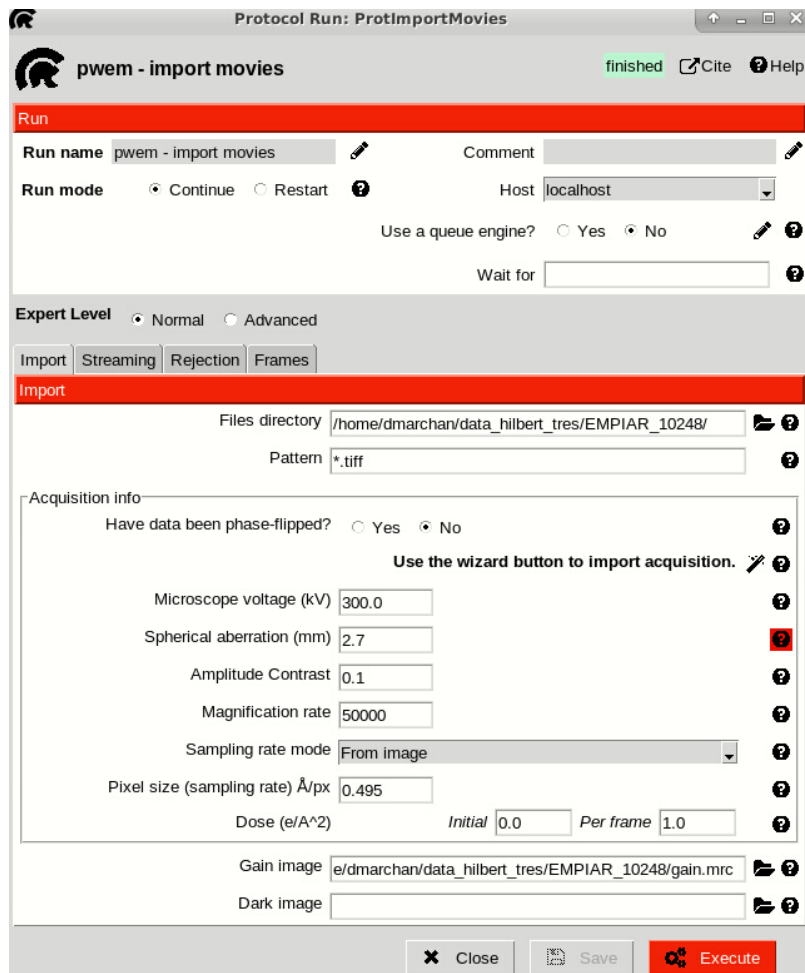


Figura 3.5: Protocolo importar imágenes

Una vez hecho, se presentan una serie de campos que deben rellenarse. Entre ellos se encuentra el voltaje del microscopio y el tamaño del píxel, que son los parámetros a tener en cuenta. Son importantes para que los resultados tengan sentido a nivel de dimensiones y resolución. Por último, podremos importar otro archivo que es la ganancia de la imagen para afinar la amplificación de la señal.

Con este ejemplo obtenemos una lista de 30 movies que podremos visualizar una a una como se muestra en la Figura 3.6 donde se enseña la primera de la lista. Cada una contendrá 50 frames con un tamaño de imagen 3710x3838 y un valor del pixel size de 0.495 como se indicó en la Figura 3.5.

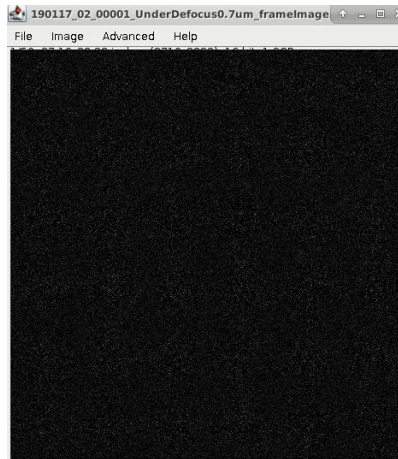


Figura 3.6: Movie

Creación micrografías

Una vez están cargadas las imágenes en el proyecto es momento de aplicar las correspondientes alineaciones. En este paso ya se presentan las primeras alternativas en los protocolos en función del objetivo.

- **Movie alignment.** Este método se encarga de corregir variaciones en la posición a nivel de frames en las movies en el momento que son tomadas con el microscopio. Para ello, se pasa como parámetro en la figura 3.7 las movies importadas en el apartado anterior y se toma como referencia artefactos que sean fáciles de seguir entre frames para calcular el desplazamiento entre frames en cada una de las imágenes [10].

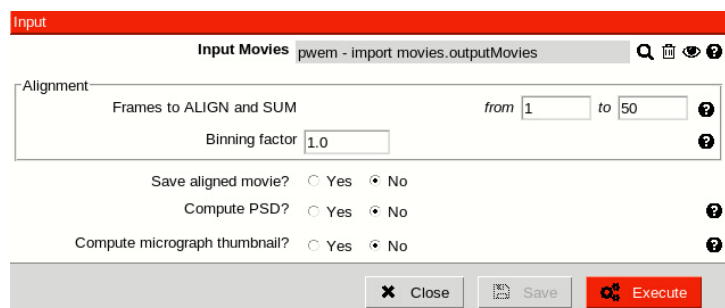


Figura 3.7: Protocolo movie alignment

Con esto se obtiene por cada imagen una gráfica que muestra el alineamiento entre el primer y el último frame (del 1 al 50) y su correspondiente micrografía en la Figura 3.8.

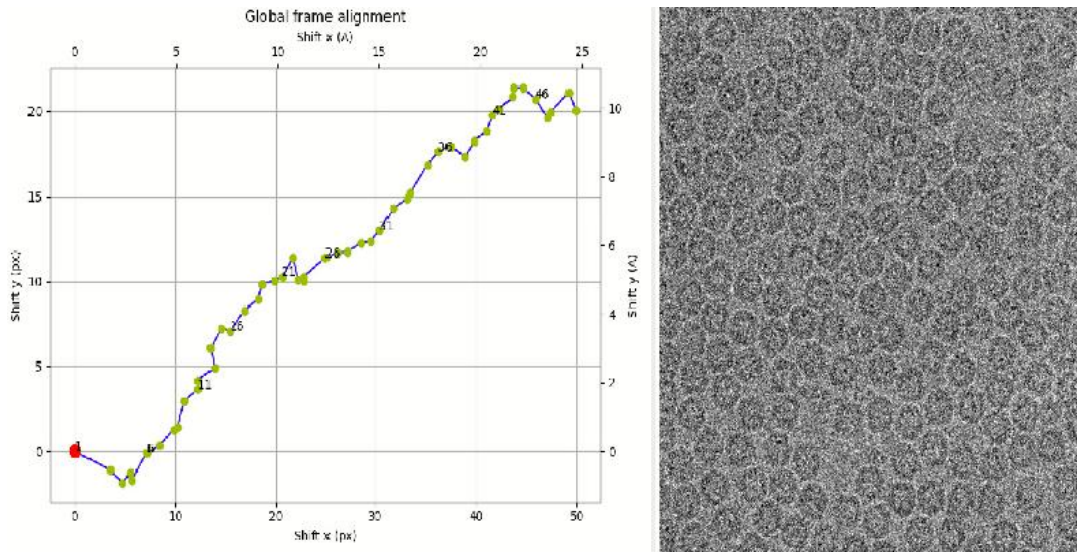


Figura 3.8: Resultado Micrografías

- **Movie maxshift.** Este protocolo aplica un filtro en función del alineamiento obtenido en el apartado anterior. Para ello filtra las movies que tienen un desplazamiento entre frames o global mayor al límite impuesto en la Figura 3.9, provocando que pueda darse el caso que se descarten imágenes que no cumplan las condiciones impuestas y por tanto no serán utilizadas para la creación del volumen final. En este ejemplo ninguna imagen se excede del límite establecido por lo que no se descartará ninguna.

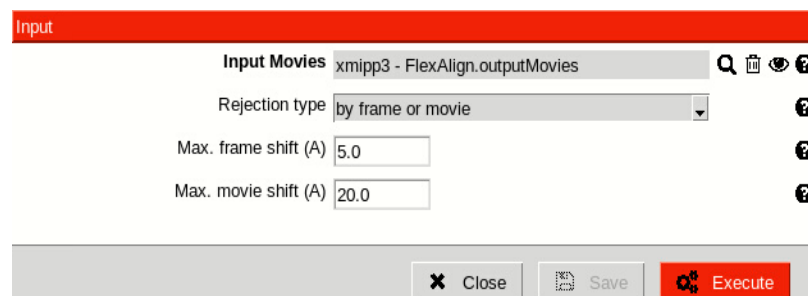


Figura 3.9: Protocolo movie maxshift

Estimación CTF

La estimación de la CTF es un paso fundamental en el procesamiento de imágenes. En la Figura 3.10 se muestra la interfaz del protocolo usado llamado *ctf-estimation* donde se introducen las micrografías obtenidas, y como parámetros para tener en cuenta, el *downsampling* que ayuda a ver mejor sus anillos de la y a determinar la resolución, y el *box size* que establece el tamaño en píxeles de la imagen donde se realiza la estimación.

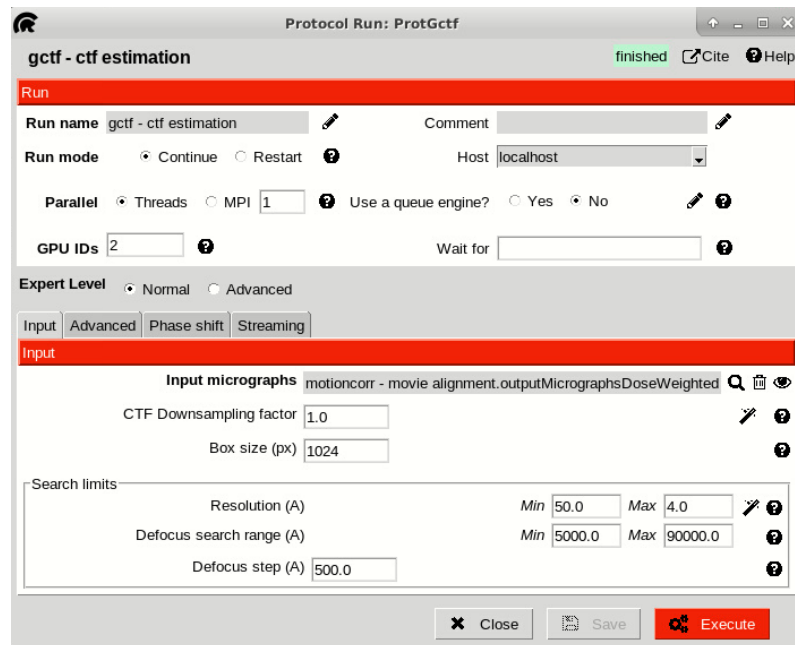


Figura 3.10: Protocolo estimación CTF

Una vez introducidos los parámetros, el resultado muestra la función de transferencia de fase del microscopio Figura 3.11, que refleja la calidad de la imagen y la resolución hasta donde se tiene información útil en la micrografía.

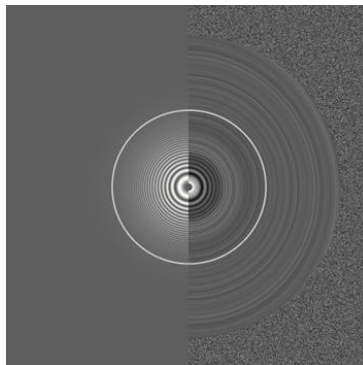


Figura 3.11: Estimación CTF

La estimación de la CTF permite corregir los efectos del desenfoque de la imagen y el ruido en la muestra, mejorando la resolución y la calidad de la imagen final.

Extracción partículas

Es otro de los pasos más importantes, que consiste en seleccionar las partículas que sean de interés de las micrografías usadas hasta el momento y extraerlas en imágenes individuales. Para ello previamente se puede hacer una selección manual donde el usuario indica de la micrografía un par de ejemplos de partículas buenas, para que posteriormente el algoritmo se vaya entrenando y elija de forma automática el resto de las partículas en todas las micrografías.

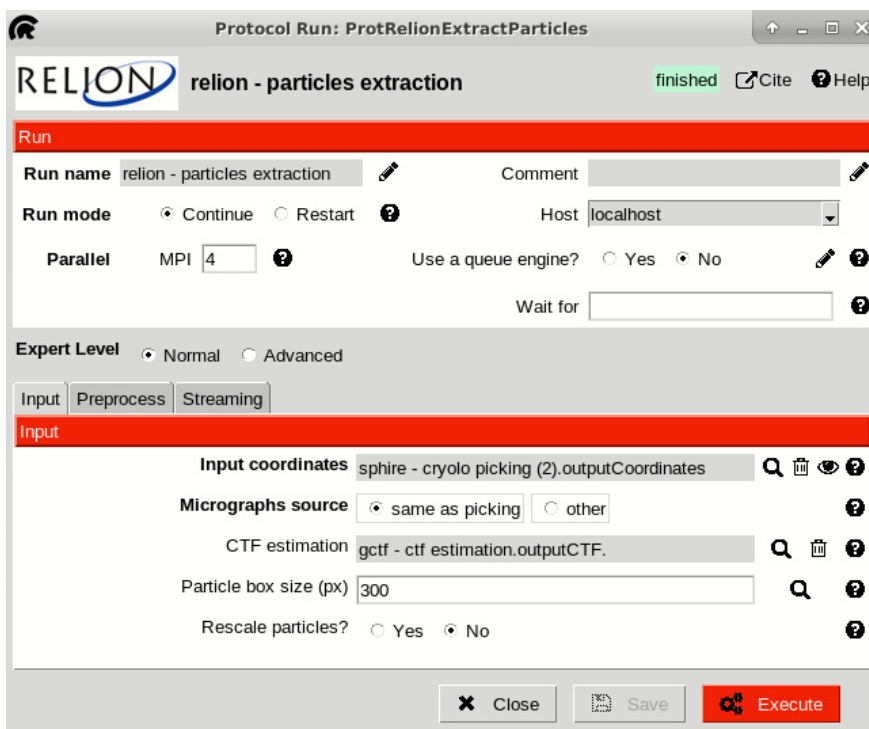


Figura 3.12: Protocolo extracción partículas

Una vez hecho, en la Figura 3.12, se usará el protocolo *relion -particle extraction* con la información de la CTF previamente calculada para corregir la distorsión del microscopio y generar una lista con todas las imágenes 2D de cada una de las partículas. Es posible que se dejen partículas sin marcar o que marque como correcta una que realmente no lo es como se ve en la Figura 3.13. Para disminuir el número de fallos se puede corregir el algoritmo indicando los lugares donde está fallando para que en el próximo entrenamiento se obtengan mejores resultados.

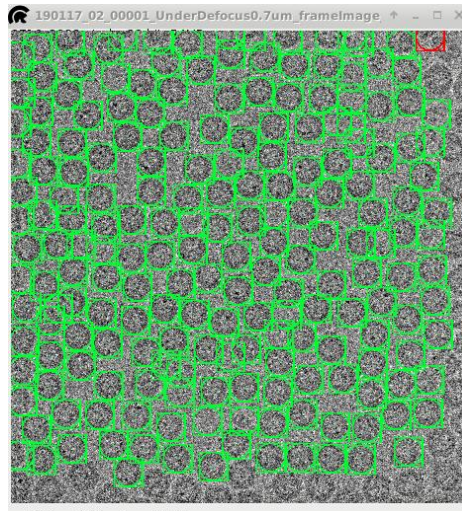


Figura 3.13: Entrenamiento partículas

Clasificación 2D

Este proceso se encarga de agrupar todas las partículas previamente extraídas en diferentes clases que comparten similitudes en su estructura y orientación. Para ello se utiliza un algoritmo de clasificación con el objetivo de maximizar la similitud dentro de una clase y minimizar entre clases diferentes [10].

El protocolo utilizado para la clasificación 2D es *cryosparc – 2D classification*, el cual cuenta con varios parámetros ajustables, entre ellos, el número de clases y el número de iteraciones que se realizan. Es crucial elegir los valores adecuados en ambos casos para tener una clasificación óptima. En el caso de las clases, si el número es demasiado bajo, las partículas pueden quedar agrupadas de forma que no se refleje adecuadamente las diferencias entre ellas, mientras que si es muy alto se podría generar una sobreclasificación dando lugar a clases vacías. Por otro lado, pocas iteraciones implica que no se alcance una convergencia en la clasificación, y excesivas generaría una sobrerefinación innecesaria.

Finalmente, se debe revisar el resultado de la clasificación y distinguir entre las clases válidas, como se puede observar en la Figura 3.14, y descartar aquellas que no sean apropiadas. Es importante realizar esta etapa con atención ya que de ello dependerá la calidad de los siguientes pasos en el procesamiento de la muestra.

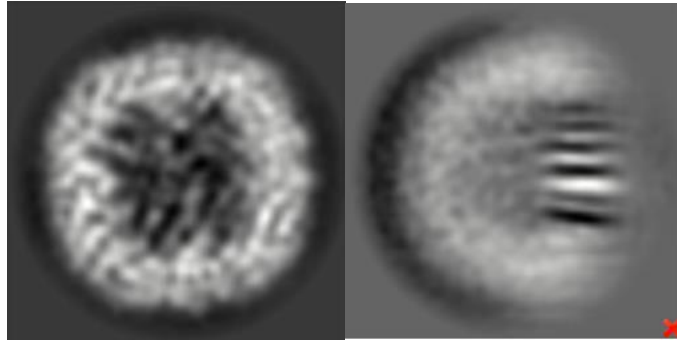


Figura 3.14: Clasificación 2D. Izqda buena, dcha descartada

Volumen inicial

Con las clases 2D ya creadas es momento de construir una primera aproximación en 3D de lo que se conoce como volumen inicial. Como no se conocen las orientaciones de las partículas que conforman las clases con respecto al volumen se hace una primera estimación dando lugar al volumen de la figura 3.15.



Figura 3.15: Volumen inicial

Refinamiento

El proceso de refinamiento tiene como objetivo mejorar su resolución para obtener una estructura más detallada y precisa. Para lograr esto, se vuelven a clasificar de forma iterativa las partículas para mejorar su clasificación y obtener una mejor calidad en la reconstrucción del volumen final [10]. Este proceso se lleva a cabo mediante el protocolo *cryosparc - 3D homogeneous refinement*.

3.2 Cliente:used ítems tracker

3.2.1 Cliente:Track used ítems protocol

Después de haber explicado detalladamente el funcionamiento de los procesos básicos para la creación de un volumen refinado a partir de la muestra obtenida con el microscopio. Es importante destacar que sobre este protocolo se añade la funcionalidad necesaria para que realice una reconstrucción tridimensional completa del volumen con el fin de obtener todos los elementos (micrografías, ctf, partículas, clases) utilizados en dicho proceso. Para conseguir mayor cantidad de detalle, el protocolo también incluye todos los datos que han sido descartados durante el proceso.

Para lograrlo, se realiza el camino inverso de cada uno de los procesos explicados anteriormente de forma que se vaya actualizando la información de forma automática acorde al protocolo correspondiente. Es decir, comenzando por la reconstrucción del volumen, le sigue la creación del volumen inicial, luego la clasificación 2D, selección de partículas, cálculo de CTF, alineamiento y, por último, la importación de las imágenes

Cabe destacar que para completarlo de forma exitosa solo es necesario introducir el volumen refinado en la Figura 3.16.

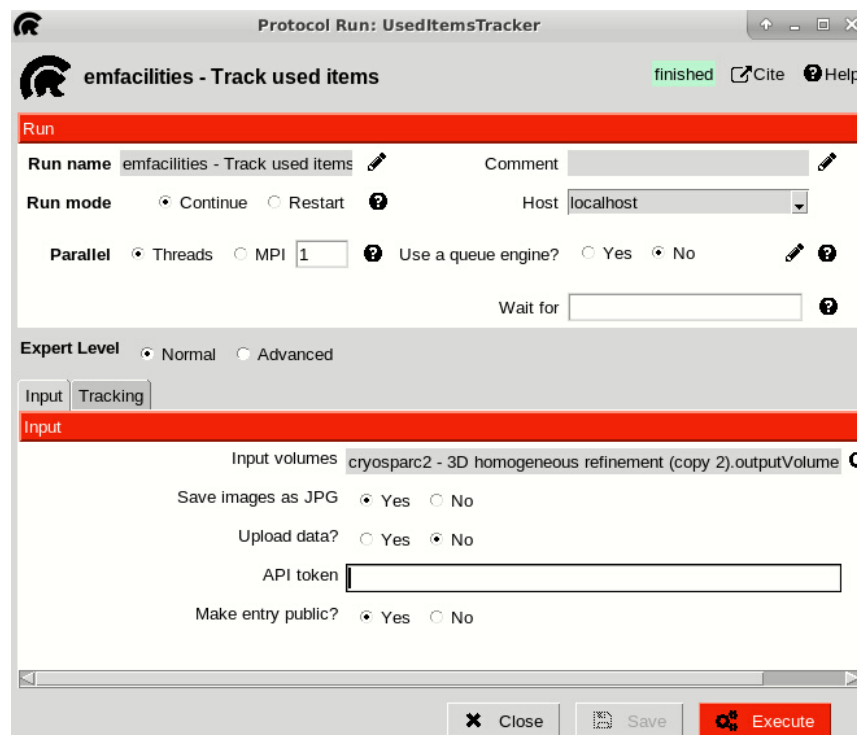


Figura 3.16: Protocolo Track used items. Parte 1 de 2.

Cuando se complete la primera parte del proceso, la segunda parte del objetivo incluye la comunicación un servidor externo. Esto permitirá el envío de todas las imágenes generadas junto con información relevante, como su nombre, categoría y etiqueta de uso, además de datos adicionales como el nombre del proyecto, el identificador del protocolo, entre otros.

Cuando el traspaso se complete de forma exitosa, el servidor añadirá la información recibida a su base de datos para posteriormente mostrarlo en una página web a modo de repositorio con una interfaz intuitiva. De esta forma, el usuario podrá visualizar de forma rápida y sencilla todos los proyectos que se hayan enviado al servidor tanto suyos como de demás personas.

3.1 Generación de resultados

El primer paso clave a nivel de código consiste en obtener el nombre del proyecto de Scipion, para crear varios diccionarios basados en dicho nombre, incluyendo uno que contiene la información de los protocolos incluidos en el proyecto, otro con las entradas necesarias para utilizarlos y el último con las salidas que generan. Utilizando toda esta información se lleva a cabo un proceso recursivo que va generando un grafo que representa la relación entre el protocolo actual y el primero de todos que actúa como nodo raíz.

Este grafo es una estructura de datos que representa un conjunto de nodos conectados por aristas. Cada arista contiene la dirección desde un nodo de origen a un nodo de destino. En este caso, el grafo tiene 11 nodos, que corresponden al número de protocolos que conforman el workflow utilizado y 12 aristas debido a sus conexiones. El número de aristas es mayor que el de nodos debido a que se introduce una bifurcación en el flujo.

Se inicia el recorrido en el protocolo actual, que es el último de todos, y se avanza hasta llegar a la raíz, que es el primero, pasando por todos los pasos intermedios. Cuando un nodo se encuentra con una bifurcación, se elige uno de los posibles caminos hasta alcanzar a la raíz, para posteriormente, volver al protocolo anterior a la bifurcación y elegir el camino opuesto. Así, se garantiza que se cubran todas las posibles rutas en el workflow y se llegue a la solución deseada.

Una vez está definida la estructura del grafo, el proceso de obtener los datos involucrados en el proceso de reconstrucción se facilita.

El proceso consiste en calcular el nodo más cercano a la raíz que contenga los datos que se desean conseguir y el nodo más lejano para el caso contrario. De esta manera se obtienen dos diccionarios, el primero al ser el más cercano contendrá los datos totales de la muestra y, por lo tanto, el más lejano al haber pasado por todos los procesos de clasificación está formado únicamente por las muestras usadas.

Con esta información el siguiente paso es comparar los dos diccionarios mediante su número de identificador para obtener la lista con los nombres de las partículas usadas y la lista con las no usadas, haciendo que la suma de ellas sea igual al total de las muestras. Para finalizar se crean los sets con su correspondiente nombre siempre y cuando haya al menos 1 muestra, con las imágenes en formato .jpg y su tamaño, el número de elementos y el tamaño del píxel. Del mismo procedimiento se calculan los sets para los datos restantes, pudiendo visualizarlos de forma individual para comprobar con mayor detalle los resultados. En la Figura 3.17 se muestra la salida que contiene la interfaz de Scipion donde están presentes todos los sets incluso para los que están vacíos. La forma de comprobar que están todos los datos es sumar los elementos usados y no usados de un mismo tipo.

<ul style="list-style-type: none"> ▽ Input inputVolumes (from cryosparc2 - 3D homogeneous refinement (copy 2) -> outputVolume [outputVolume]) ▽ Output ☞ emfacilities - Track used items (copy 10) -> usedParticles ☞ emfacilities - Track used items (copy 10) -> notUsedParticles ☞ emfacilities - Track used items (copy 10) -> usedMicrographs ☞ None ☞ emfacilities - Track used items (copy 10) -> usedCTFs ☞ None ☞ emfacilities - Track used items (copy 10) -> usedCoordinates ☞ emfacilities - Track used items (copy 10) -> notUsedCoordinates ☞ emfacilities - Track used items (copy 10) -> usedClasses2D ☞ emfacilities - Track used items (copy 10) -> notUsedClasses2D 	<ul style="list-style-type: none"> Volume (300 x 300 x 300, 0.49 Å/px) - w/halves SetOfParticles (3868 items, 300 x 300, 0.49 Å/px) SetOfParticles (1359 items, 300 x 300, 0.49 Å/px) SetOfMicrographs (30 items, 3710 x 3838, 0.49 Å/px) SetOfMicrographs (0 items, No-Dim, -999.00 Å/px) SetOfCTF (30 items) SetOfCTF (0 items) SetOfCoordinates (3868 items, 239 x 239) SetOfCoordinates (1359 items, 239 x 239) SetOfClasses2D (16 items) SetOfClasses2D (33 items)
--	--

Figura 3.17: Sets de imágenes generados por scipion

Para facilitar la visualización de todas las imágenes generadas, se crea en el directorio del proyecto una carpeta con el nombre del identificador del protocolo ejecutado y una lista de subcarpetas separadas por categorías. Como se observa en la Figura 3.18, estas contienen las mismas imágenes. Con esto, se cumple la primera parte del primer objetivo del protocolo.

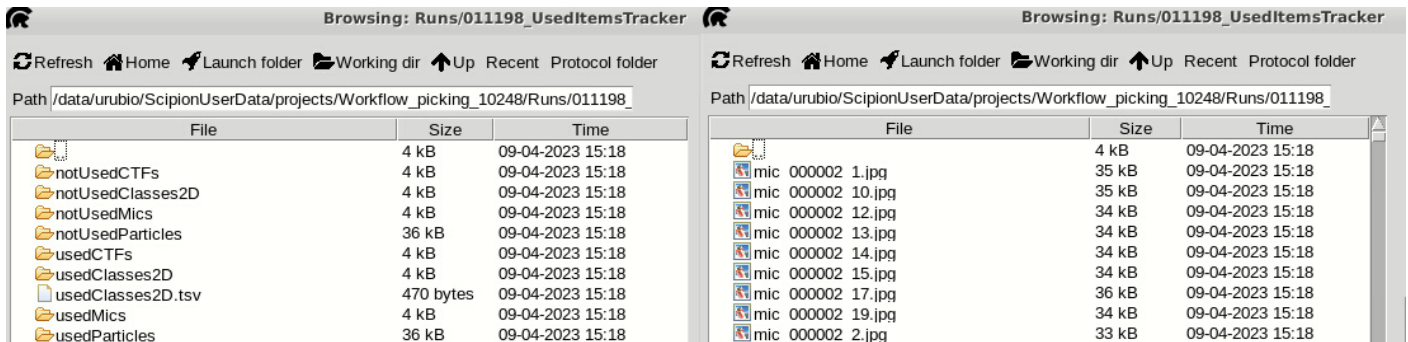


Figura 3.18: Imágenes separadas en categorías

3.2.2 Solicitud POST

La segunda parte del primer objetivo consiste en tras haber creado las carpetas mencionadas en el ejemplo anterior, enviar todas las imágenes generadas y la información relevante del proyecto a un servidor web en Django de manera automática al ejecutar el protocolo. Se utiliza una API creada para comunicarse con el servidor que se explicará en la siguiente sección.

Para enviar datos se utiliza el método POST procedente de la biblioteca *requests* de Python. A ese método se le pasan como argumentos 4 parámetros, los cuales son los siguientes:

El primer valor es establecer un *endpoint* del servidor web ya que es necesario conocer la ruta a la que se va a conectar y va a establecer una comunicación.

A continuación, se establece un encabezado de autorización utilizando un token que previamente se introduce al ejecutar el protocolo (en el apartado de pruebas se explica cómo generarlo). Esta comprobación es esencial para autenticar la solicitud del emisor y asegurar que el usuario que envía los datos está autorizado.

Posteriormente, se crea el diccionario que contendrá casi todos los datos que serán enviados al servidor. En este caso, este contendrá información de la categoría de cada imagen, el nombre del archivo y del proyecto, la resolución, el voltaje, si la imagen ha sido usada o no, entre otros parámetros

El último parámetro son las imágenes ya que al ser archivos no pueden incluirse dentro del diccionario.

El protocolo de comunicación que se sigue para hacer esta solicitud es HTTP, ya que es el estándar en la comunicación web entre clientes y servidores tal y como se muestra en la figura 3.19.

Está apoyado por el protocolo TCP donde el servidor está en constante escucha a la espera de una solicitud HTTP por parte del cliente. Cuando esto ocurre, el protocolo TCP se encarga de mantener la comunicación y asegurar que todos los paquetes lleguen al receptor [11]. Si la solicitud se completa de manera correcta, el servidor mandará la respuesta HTTP con el código de estado 200. En cambio, si se produce alguna anomalía durante el proceso se mostrará y se lanzará una excepción con un código diferente donde se indicará el origen del problema.

En este caso los errores que se tratan son si el token introducido no es válido devolviendo así el código *401*, o si el tipo de datos introducido en alguno de los campos no es el correcto con el código *400*.

Extraído de <https://blogs.ugr.es/tecweb/los-codigos-respuesta-del-servidor-http-claves-arquitectura/>

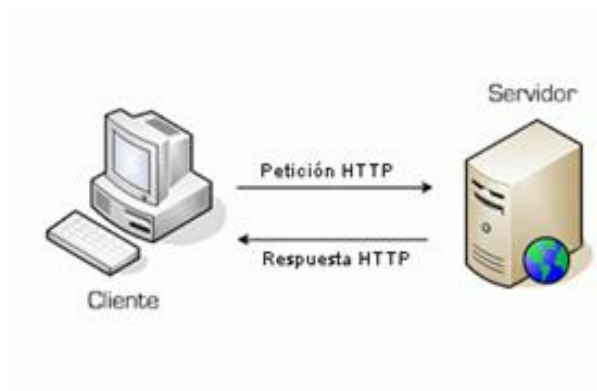


Figura 3.19: Comunicación Cliente-Servidor

3.3 Servidor Django

Para la creación del servidor Django REST framework es importante contar con la utilización de una API REST para permitir una comunicación fluida con el cliente.

Una API REST es un conjunto de protocolos y reglas que se utilizan para acceder a un servicio web y establecer una comunicación entre aplicaciones para compartir recursos. Su arquitectura se basa en un protocolo HTTP ya que es el más utilizado para enviar y recibir respuestas de un servidor web. Las solicitudes pueden ser de varios tipos dependiendo de la funcionalidad que se quiera conseguir en los recursos, GET (obtener), POST (crear recurso nuevo), PUT (para actualizar) y DELETE (eliminar) [12]. Se utiliza Django REST framework que es un paquete creado sobre Django que cuenta con herramientas y funciones que facilitan el proceso de creación de la API.

Django es un framework de alto nivel y código abierto escrito en Python que se utiliza para la creación de aplicaciones web complejas y escalables, con una arquitectura MVC (Modelo-Vista-Controlador) que separa la lógica de la aplicación, la interfaz de usuario y la base de datos [13].

Es decir, el proyecto está separado en 3 partes:

- **Modelo.** Define cómo va a organizarse la estructura de los datos enviados. Para ello se crea un archivo que crea todas las variables con el tipo de dato que ha de componer la aplicación web (booleano, entero, cadena de caracteres).
- **Vista.** Se encarga de la presentación de la información. Para ello se crean plantillas HTML para especificar cómo se muestran los datos del modelo.
- **Controlador.** Actúa como intermediario entre el modelo y la vista. Tiene como objetivo manejar las solicitudes del usuario, actualizar la información creada en el modelo y por consecuencia la vista.

Al crear el proyecto con nombre *repositorio* se genera el archivo *manage.py* que será útil en la línea de comandos para realizar migraciones en la base de datos para poder lanzar el servidor, y la base de datos *db.sqlite3* que contendrá toda la información sensible.

A su vez se crea una carpeta con el mismo nombre que el proyecto, que entre sus archivos contendrá uno llamado *settings.py* donde se especifica las aplicaciones instaladas, la ruta de las plantillas, base de datos y archivos estáticos, entre otros. Además, se encuentra el archivo *urls.py* que contendrá la configuración de las rutas de la aplicación.

El siguiente paso es crear la aplicación *ap1* que contendrá toda la lógica de programación para procesar los datos cuando llega una solicitud POST. Los archivos generados son los siguientes:

- ***models.py***: sirve para definir las clases que contendrán los variables a utilizar, con su tipo de datos y relaciones con otros modelos. En este caso se ha creado un solo modelo que contiene todas las variables enviadas en la solicitud POST y alguna extra como por ejemplo poder guardar la fecha a la que llega un paquete.
- ***views.py***: es el archivo más extenso ya que contiene todas las funciones del programa. En cada una de ellas se procesan las peticiones, se realiza la correspondiente gestión y se devuelve una respuesta. La forma de mostrar las respuestas se hace pasando los datos correspondientes a una plantilla HTML, que es renderizada y se devuelve una respuesta HTTP para que se le muestre al usuario.
- ***urls.py***: crea la relación entre la dirección URL que se mostrará en la página web y la función a la que estará asociada.
- ***forms.py***: sirve para definir los formularios de los datos recibidos para luego hacer las correspondientes comprobaciones de una forma más sencilla.
- ***admin.py***: se registran los modelos definidos, que en este caso solo será uno.

La carpeta *media* se ha creado con la única finalidad de almacenar todas las imágenes que son enviadas por parte del cliente. La forma hacerlo se ha establecido que sea mediante carpetas en función de su nombre de usuario para que sea más accesible su manipulación y no haya problemas de solapamiento. Dentro de cada usuario existen subcarpetas en función del proyecto y de la categoría a la que pertenecen las imágenes.

La carpeta static de la figura 3.2 contiene todos los archivos estáticos como Javascript y hojas de estilo CSS necesarios para el desarrollo.

Los archivos CSS sirven para dar un estilo a los elementos HTML y cambiar su formato para que sean más bonitos para los usuarios. En este caso, se han aplicado una serie de propiedades, como el formato a los botones para modificar su diseño dependiendo su función, introducción de colores, cambios en el tamaño de la fuente y márgenes en el texto.

Por otro lado, los archivos Javascript están para la creación de la tabla principal del programa, las características que lo componen y la funcionalidad de los botones, ya sea para borrar o descargar proyectos, e incluso mostrar las imágenes al ser seleccionadas. Se llaman archivos estáticos porque no cambian en ningún momento y son fijos para cualquier interacción del usuario.

La última de las carpetas creadas es templates, la cual contiene todos los archivos HTML que se utilizan. Cada archivo HTML define la estructura y el contenido que se mostrará en las diferentes vistas de la aplicación. Estos archivos se utilizan para generar las salidas y definir cómo se comportan las páginas web.

Para utilizar estas plantillas, se debe llamar a los archivos HTML correspondientes en las funciones de vista. En estas funciones, se pasan los datos necesarios para el manejo de la plantilla, lo que permite generar la página web dinámicamente. En el proyecto se incluyen diversas plantillas como, por ejemplo, una para mostrar un formulario de registro, otra para iniciar sesión, cambiar contraseñas, borrar usuarios y mostrar imágenes, entre otras.

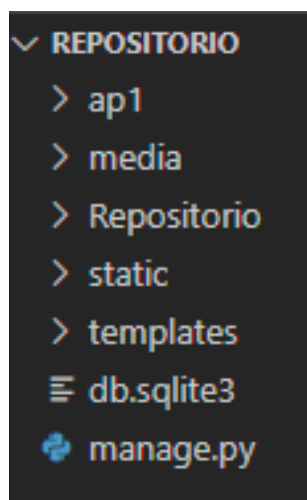


Figura 3.20: Estructura proyecto Django

4. Implementación

Para probar la aplicación y comprobar que funciona todo correctamente, es necesario lanzar el servidor en Django y acceder a la URL de localhost. De esta manera se puede usar el proyecto entero, pero únicamente desde el ordenador donde se lanza el servidor, es decir, no está abierto al público. Para convertirlo en un entorno público y permitir el acceso desde cualquier dispositivo se alojará en Apache, que es un servidor de código abierto de los más utilizados para alojar aplicaciones web.

Los pasos necesarios para completar el alojamiento consisten en instalar apache para el sistema operativo adecuado y el módulo de apache mod_wsgi para que se puedan ejecutar las aplicaciones de Django en el servidor. A continuación, hay que alojar el proyecto dentro de la carpeta donde se haya completado la instalación y modificar el archivo httpd.conf incluyendo las rutas de las dos instalaciones.

En cuanto a Django, es necesario modificar el archivo wsgi.py que se crea por defecto vacío e introducir el nombre del proyecto y en el archivo settings.py. Compartiendo mi IP pública se abre el servidor al público para que la gente lo pueda probar, aunque no es una práctica recomendable. Por lo tanto, a largo plazo es necesario comprar un dominio de internet y que sea así la forma en la que accedan los usuarios al servidor.

5. Pruebas

Este apartado servirá para conocer de una forma más visual el funcionamiento del trabajo, mediante una demo del funcionamiento del servidor explicando cual es el procedimiento para ejecutar los programas tanto por parte del cliente(scipion) y servidor.

El primer paso de todos será acceder al servidor y registrarnos introduciendo un nombre y una contraseña en la Figura 5.1.

Please register

Username: Required. 150 characters or fewer. Letters, digits and @/./+/_ only.

Password:

Password confirmation: Enter the same password as before, for verification.

Already have an account? [Login](#)

Figura 5.1: Ventana de registro

En el caso de que hubiese una discordancia entre la contraseña y su confirmación saldría un mensaje de error en la pantalla. Si completamos este paso correctamente aparecerá otra ventana similar donde habrá que introducir el nombre y contraseña de nuevo para poder iniciar sesión en la página web. Hay que destacar que el proceso del registro no es totalmente obligatorio, ya que se podrán observar los proyectos que se hayan subido de forma pública, aunque si no se lleva a cabo no se podrá subir ninguno.

Una vez iniciamos sesión ya se podrá observar todos los proyectos subidos, aunque por el momento no hay ninguno. Para poder hacerlo es muy importante acceder a la pestaña perfil en la parte superior de la barra de navegación para acceder a los datos de nuestro usuario, figura 5.2

Perfil

User: unai

API_Token: 3b3e9d4831ea31113e714dcb9cbdff9ee1e1c9ce

Eliminar cuenta

Old password:

New password:

New password confirmation:

Cambiar contraseña

Figura 5.2: Perfil del usuario

En esta ventana aparece la posibilidad de eliminar la cuenta si ya no queremos que esté en la base de datos, lo que conlleva que se borren todos los proyectos asociados con ese usuario. Está configurado de forma que no existan dos o más usuarios con el mismo nombre por lo que no habrá problemas si de verdad se quiere eliminar el usuario. También está la posibilidad de cambiar la contraseña introduciendo la antigua y la nueva que queremos que la sustituya.

La parte importante de la información que aparece es la que corresponde con *API_Token*. Este valor se genera de forma automática al registrarse en la página, y al igual que con los nombres, no habrá dos iguales. Cabe destacar que, si se borra el usuario, el token lo hará de forma automática y ya no será útil. Es un parámetro esencial ya que sin él no se podrá enviar ningún tipo de información.

Una vez ya tenemos el token generado es momento de acceder a scipion y en la interfaz del protocolo donde se introduce el volumen para generar los datos. Los parámetros que hay que rellenar son los siguientes:

El primero de todos llamado *Upload data* sirve para saber si el programa ha de hacer la solicitud POST y enviar los datos. Esto se hace por si el usuario quiere asegurarse primero que los datos se generan correctamente y que no existen fallos en el código a la hora de generar imágenes. En nuestro caso habilitamos la casilla para que si se llame a la función de envío de datos.

En el segundo parámetro *API_Token* hay que introducir el token mostrado en la Figura 5.2. Si no se rellena el campo o el valor introducido es incorrecto, saltará una excepción diciendo que el token no es válido ya que no existe en la base de datos de Django.

La última opción para rellenar *Make entry public* sirve para saber si el usuario quiere que los datos que se envíen se suban de forma pública o privada. Si los hace públicos, cualquier persona que se haya registrado o no, puede verlos, y por el contrario de forma privada solo lo puede ver la persona que inicie sesión con el nombre de usuario asociado al token. En este ejemplo se habilitará la casilla para que se suban de forma pública tal y como se muestra en la figura 5.3.

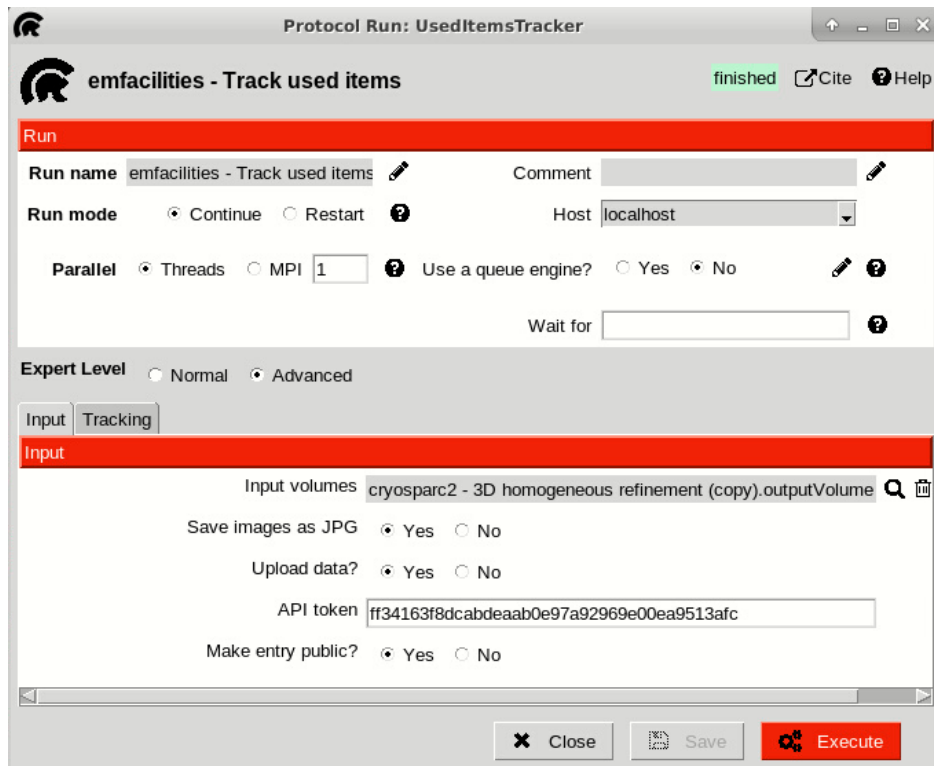


Figura 5.3: Protocolo Track used ítems. Parte 2 de 2.

Cuando el proceso termine y nos aseguremos que no se haya lanzado ningún error en la interfaz de scipion, es momento de volver al servidor, para empezar a ver toda la información del proyecto.

En la Figura 5.4 podemos observar una tabla en la que se muestran todos los proyectos asociados al usuario y la entrada de dicho proyecto con todas sus características separadas por columnas. En la primera, se muestra la fecha con día, mes y año en la que el proyecto fue mandado. La segunda indica el nombre del proyecto y actúa como un enlace para mostrar las carpetas que lo componen.

El botón azul permite al usuario poder descargar la carpeta del proyecto en formato .zip en su escritorio con todas sus imágenes.

El apartado descripción otorga una breve información de cuáles son las categorías que tiene y cuántas imágenes hay en cada una de ellas. El resto de las columnas dan información del propietario del proyecto, del valor del voltaje del microscopio con el que adquirieron las fotos y el tamaño del píxel. Además, existe un botón rojo que permite la opción de eliminar un proyecto si ya no queremos que se muestre en el servidor. La única condición para poder hacerlo es que el nombre del usuario que tiene la sesión iniciada sea igual al nombre del propietario del proyecto.

Existe la opción de buscar y filtrar por nombre o parámetros para facilitar la navegación y búsqueda de proyectos para usuarios que tengan un alto volumen de entradas.

Fecha	Proyecto	Descripción	Autor	Resolución	Voltaje	Borrar
2023-04-18	Workflow_picking_10248 	<ul style="list-style-type: none"> • notUsedClasses2D (33) • notUsedParticles (1359) • usedCTFs (30) • usedClasses2D (16) • usedMicrographs (30) • usedParticles (3868) 	unai	0.495 A/px	300 kV	Eliminar

Showing 1 to 1 of 1 entries

Previous 1 Next

Figura 5.4: Proyecto mostrado interfaz web

Si accedemos al nombre del proyecto, este nos llevará a la ventana mostrada en la figura 5.5 que contiene el nombre del proyecto, el identificador del protocolo de scipion ejecutado y un directorio por cada categoría que permite ver las imágenes una por una. En este caso se despliega la casilla de partículas y usadas para que aparezca un listado de todas las imágenes con su nombre. Al igual que el proyecto entero, incluye la opción de descargar la carpeta indicada, ya que es posible que un mismo proyecto incluya la ejecución de varios protocolos distintos y no queramos que se nos descarguen imágenes que no nos sean útiles.

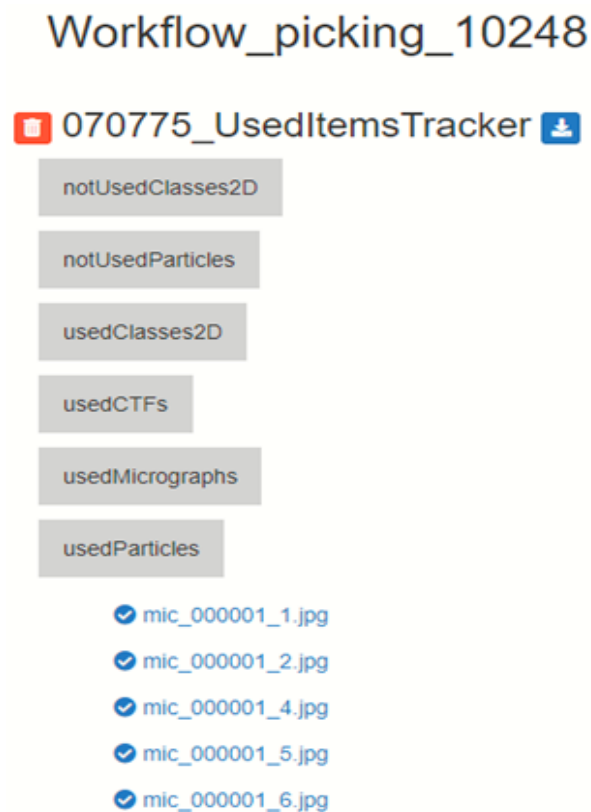


Figura 5.5: Categorías e imágenes de un proyecto

6. Conclusiones y Trabajo futuro

6.1 Conclusiones

En conclusión, el proyecto ha logrado su objetivo de crear un repositorio de datos para scipion mediante un protocolo eficiente que permite la creación y el envío de los datos.

Se ha desarrollado un servidor con su respectiva API para mostrar la información a los usuarios. Este trabajo de fin de grado ha permitido adquirir una valiosa experiencia en el desarrollo de soluciones tecnológicas innovadoras para la gestión de datos científicos como, por ejemplo, frontend y backend de una aplicación web desarrollada desde 0.

Además, el proyecto ha seguido los pasos necesarios para la reconstrucción de un volumen 3D a partir de la importación de imágenes. El protocolo desarrollado ha sido eficiente y ha permitido distinguir los datos utilizados de los que no, lo que ha permitido la creación de un repositorio de datos preciso y útil para el desarrollo de nuevos métodos que requieran de datos etiquetados (aprendizaje supervisado). La API desarrollada en Django ha sido clave en la recopilación y tratamiento de los datos, creando un servidor que facilita su acceso y uso en futuras investigaciones científicas.

6.2 Trabajo futuro

Una posible vía de mejora en cuanto a scipion podría consistir en ofrecer un mantenimiento continuo al plugin del protocolo utilizado, para mejorar la optimización de los resultados en cuanto al tiempo de ejecución.

Asimismo, en cuanto al servidor es importante considerar la seguridad y llevar a cabo mejoras que permitan controlar de forma más exhaustiva que no puedan subirse archivos maliciosos. Por otra parte, se deberían realizar revisiones periódicas a la cantidad de memoria usada de la base de datos, asegurándose de que se encuentre en óptimas condiciones. Esto permitiría garantizar la integridad de los datos y evitar problemas de rendimiento a largo plazo.

Bibliografía

- [1] UbiquisCiencia, "Cryo-EM o Criomicroscopía Electrónica," Medium, 17 diciembre 2019. [En línea]. Disponible: <https://ubiquisciencia.medium.com/cryo-em-o-criomicroscopía-electrónica-45d424bc2808>.
- [2] A. Bartesaghi, J. L. Aguerrebere, X. Kong, et al., "Direct Detection of Diverse Membrane Proteins in Virus-like Particles using CryoEM," PLoS One, vol. 11, no. 3, p. e0151274, 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4788751/>
- [3] C. M. Hernández, "Nobel de Química 2017 para la criomicroscopía electrónica," CienciaMundo, 04 octubre 2017. [En línea]. Disponible: <https://blogs.ciencia.unam.mx/cienciamundo/2017/10/04/nobel-de-quimica-2017-para-la-criomicroscopia-electronica/#:~:text=Entre%201975%20y%201986%20desarroll%C3%B3,revelar%20una%20estructura%20tridimensional%20definida>
- [4] M. C. Scott, "Cryo-EM for Dummies: Cryo-EM BPS 2020," Biophysical Society Blog, 25 febrero 2020. [Online]. Available: <https://www.biophysics.org/blog/cryo-em-for-dummies-cryo-em-bps-2020>
- [5] M. A. Sánchez et al., "Una nueva era del hielo: Cryo-EM," CONICET Mendoza Blog, 18 septiembre 2019. [En línea]. Disponible: <https://www.mendoza.conicet.gov.ar/blog/una-nueva-era-del-hielo-cryo-em/#:~:text=Una%20de%20las%20mayores%20ventajas,estructuras%20grandes%2C%20complejas%20y%20flexibles>

- [6] "Single Particle Analysis (SPA)," Creative Biostructure, [En línea]. Disponible: <https://www.creative-biostructure.com/single-particle-analysis-spa.htm>
- [7] A. Bhushan, "Introduction to cryo-EM," EMBL Hamburg, 2017. [Online]. Available: https://www.embl-hamburg.de/biosaxs/courses/embo2017/slides/EMBO_cryoEM_2_Bhushan.pdf.
- [8] Scipion Team. (2021). Scipion 3.0 documentation. [En línea]. Disponible en: <https://scipion-em.github.io/docs/release-3.0.0/index.html>
- [9] "¿Qué es un plugin y para qué sirve?," dinahosting, 19 octubre 2018. [En línea]. Disponible: <https://dinahosting.com/blog/que-es-un-plugin-y-para-que-sirve/>.
- [10] "CryoEM101 - Chapter 5," CryoEM101, [En línea]. Disponible: <https://cryoem101.org/chapter-5/>
- [11] J. A. Callejón, "Los códigos respuesta del servidor HTTP: claves arquitectura," Tec Web, 2014. [Online]. Available: <https://blogs.ugr.es/tecweb/los-codigos-respuesta-del-servidor-http-claves-arquitectura>.
- [12] "Creating a REST API with Django REST Framework," Stack Abuse, 22 enero 2021. [En línea]. Disponible: <https://stackabuse.com/creating-a-rest-api-with-django-rest-framework/#whatisarestapi>.
- [13] Mozilla. (2022). Introduction to Django [Online] available:<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>