

**Plataforma software extensible para el despliegue, ejecución y visualización de
algoritmos dedicados a la visión por computador**

JUAN SEBASTIAN MARCON CABALLERO

**Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga
2021**

Plataforma software extensible para el despliegue, ejecución y visualización de algoritmos dedicados a la visión por computador

JUAN SEBASTIAN MARCON CABALLERO

Una tesis presentada en cumplimiento de los requisitos para el grado de:
Ingeniero de Sistemas e Informática

Director:

FABIO MARTÍNEZ CARRILLO, Ph.D

Codirector:

GABRIEL RODRIGO PEDRAZA FERREIRA, Ph.D

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería de Sistemas e Informática
Bucaramanga

2021

Agradecimientos

Los autores expresan su agradecimiento:

A mi madre Ana Mercedes Caballero, por su sacrificio y apoyo desmesurado, a quien le debo este y todos mis logros.

A mis hermanos Vannesa Marcon y Erwing Marcon que son parte fundamental de mi vida, por su apoyo incondicional en cada paso de mi infancia y desarrollo profesional.

A mi amiga, pareja y compañera de vida Paola Caicedo, por ayudarme en cada escalón de mi desarrollo personal y profesional.

A mis amigos Andres, William, Edgar, Oscar, Henry, Nicolas, Santiago, Karen, Ruben con los cuales viví esta gran etapa de mi vida, en especial a Eduard Caballero con quien compartí muchos logros y derrotas.

A mi director Fabio Martínez y codirector Gabriel Pedraza por los cargos que desempeñaron durante el desarrollo de este proyecto, por su paciencia y por no rendirse en el último paso de esta etapa.

A todas aquellas personas que influyeron en mi vida personal y profesional.

Juan Sebastian Marcon Caballero

Contenido

	Pág
Introducción	15
1. Planteamiento y justificación del problema	18
2. Objetivos	19
2.1. Objetivo general	19
2.2. Objetivos específicos	19
3. Marco de referencia	20
3.1. Micro-servicios	20
3.2. <i>Docker</i> : la virtualización y los contenedores	21
3.3. Procedimientos remotos: <i>gRPC</i>	24
3.4. Estado del arte: soluciones de software para el despliegue de aplicaciones científicas	25
4. Desarrollo del proyecto	29
4.1. Principales componentes de la solución	29
4.1.1. Tipos de usuario	29
4.1.2. Algoritmos y sus funcionalidades en la plataforma	30
4.2. Planificación de requerimientos	33
4.2.1. Requerimientos funcionales	33
4.2.2. Definición de los requerimientos no funcionales	35
4.3. Diseño del prototipo	35
4.4. Arquitectura propuesta	39

5. Plan de pruebas 43
5.1. Pruebas de integridad 43
5.2. Pruebas de aceptación 48
5.2.1. Prueba de usabilidad del sistema 49

6. Conclusiones y perspectivas 52

Bibliografía 54

Anexos 57

Lista de figuras

	Pág
Figura 1. Arquitecturas de software de un algoritmo de sustracción de fondo. En a) se presenta una arquitectura monolítica enfocada a realizar los procesos de manera uniforme y en conjunto. En b) se representa la arquitectura basada en micro-servicios, los cuales reportan procesos independientes que permiten el despliegue independiente de aplicaciones.	21
Figura 2. Arquitectura lógica de las máquinas virtuales <i>VM</i> . Virtualización de hardware por aplicación, requiere de un <i>SO</i> por aplicación	23
Figura 3. Arquitectura lógica de los contenedores <i>Docker</i> . Virtualización de software por aplicación, solo se requiere el <i>SO</i> de anfitrión.	24
Figura 4. Arquitectura de una implementación de servicios web con la librería <i>Flask</i> . Aislación entre las tecnologías de los algoritmos nula, obliga a usar tecnologías compatibles entre algoritmos.	27
Figura 5. Diagrama de actores del sistema	37
Figura 6. Diagrama de casos de uso - Modulo: Algoritmos	38
Figura 7. Diagrama de casos de uso - Modulo: Ejecuciones	39
Figura 8. Diagrama de casos de uso - Modulo: Usuarios	40
Figura 9. Arquitectura física de la plataforma <i>Deeptools</i> . Representación de las interacciones entre las capas de la arquitectura (capa de presentación, capa de servidor. . . .	41
Figura 10. Inicio del flujo básico de la plataforma <i>Deeptools</i> . a) Interfaz gráfica para registrar usuarios en cumplimiento de CU02 (Tabla 9). b) Interfaz gráfica para iniciar sesión en cumplimiento de CU00 (Tabla 7).	45

Figura 11.Despliegue de un algoritmo. a) Registro del protocolo de toma de datos, se refiere al registro del paso a paso para la toma de datos compatibles con el algoritmo.	
b) Selección de los elementos del algoritmo. c) Toma de metadatos del algoritmo. d) Proceso de compilación del algoritmo: Seguimiento paso a paso durante la compilación del algoritmo	46
Figura 12.Ejecución de un algoritmo. Barra de progreso en tiempo real.	47
Figura 13.Resultados - Escala de SUS	50
Figura 14.CU00 - Iniciar Sesión	133
Figura 15.CU02 - Registrar usuario	134
Figura 16.CU03 - Ver perfil	134
Figura 17.CU04 - Editar perfil	135
Figura 18.CU05 - Ver perfil de desarrolladores	135
Figura 19.CU06 - Listar notificaciones	136
Figura 20.CU08 - Suscribirse	136
Figura 21.CU09 - Cancelar suscripción	137
Figura 22.CU10 - Listar suscripciones	137
Figura 23.CU11 - Desplegar algoritmo Paso 1	138
Figura 24.CU11 - Desplegar algoritmo Paso 2	138
Figura 25.CU11 - Desplegar algoritmo Paso 3	139
Figura 26.CU11 - Desplegar algoritmo Construyendo	139
Figura 27.CU12 - Iniciar algoritmo	140
Figura 28.CU13 - Detener algoritmo	140
Figura 29.CU14 - Eliminar algoritmo Selección del algoritmo	141
Figura 30.CU14 - Elimina algoritmo Confirmar eliminación	141
Figura 31.CU15 - Administrar usuarios	142
Figura 32.CU15 - Administrar usuarios Agregar suscripción	142
Figura 33.CU15 - Administrar usuarios Eliminar suscripción	143

Figura 34.CU16 - Ejecutar algoritmo	143
Figura 35.CU16 - Ejecutar algoritmo Selección de archivos	144
Figura 36.CU16 - Ejecutar algoritmo Prueba con archivos probados por el desarrollador	144
Figura 37.CU16 - Ejecutar algoritmo Algoritmo en ejecución	145
Figura 38.CU17 - Listar ejecuciones	145
Figura 39.CU18 - Ver ejecución	146
Figura 40.CU19 - Eliminar ejecución Selección de la ejecución	146
Figura 41.CU19 - Eliminar ejecución Confirmar eliminación	147
Figura 42.CU20 - Clonar ejecución Selección de ejecución a clonar	147
Figura 43.CU20 - Clonar ejecución Selección de archivos	148
Figura 44.CU20 - Clonar ejecución Algoritmo en ejecución de el experimento clonado .	148
Figura 45.CU21 - Filtrar algoritmos	149
Figura 46.CU22 - Listar algoritmos	149
Figura 47.CU23 - Descargar elementos multimedia	150

Lista de tablas

	Pág
Tabla 1. Requerimientos funcionales	33
Tabla 2. Requerimientos no funcionales	35
Tabla 3. Casos de uso	36
Tabla 4. Incidencias registrando un usuario	44
Tabla 5. Incidencias durante el despliegue de un algoritmo	46
Tabla 6. Incidencias durante la visualización de los resultados de una ejecución	48
Tabla 7. Caso de uso: CU00 - Iniciar sesión	151
Tabla 8. Caso de uso: CU01 - Cerrar sesión	151
Tabla 9. Caso de uso: CU02 - Registrar usuario	151
Tabla 10.Caso de uso: CU03 - Ver perfil	152
Tabla 11.Caso de uso: CU04 - Editar perfil	153
Tabla 12.Caso de uso: CU05 - Ver perfil de desarrolladores	153
Tabla 13.Caso de uso: CU06 - Listar notificaciones	154
Tabla 14.Caso de uso: CU07 - Notificar novedades	155
Tabla 15.Caso de uso: CU08 - Suscribirse	155
Tabla 16.Caso de uso: CU09 - Cancelar suscripción	155
Tabla 17.Caso de uso: CU10 - Listar suscripciones	156
Tabla 18.Caso de uso: CU11 - Desplegar algoritmo	156
Tabla 19.Caso de uso: CU12 - Iniciar algoritmo	157
Tabla 20.Caso de uso: CU13 - Detener algoritmo	158

Tabla 21.Caso de uso: CU14 - Eliminar algoritmo	158
Tabla 22.Caso de uso: CU15 - Administrar usuarios	159
Tabla 23.Caso de uso: CU16 - Ejecutar algoritmo	159
Tabla 24.Caso de uso: CU17 - Listar ejecuciones	160
Tabla 25.Caso de uso: CU18 - Ver ejecución	160
Tabla 26.Caso de uso: CU19 - Eliminar ejecución	161
Tabla 27.Caso de uso: CU20 - Clonar ejecución	161
Tabla 28.Caso de uso: CU21 - Filtrar algoritmos	162
Tabla 29.Caso de uso: CU22 - Listar algoritmos	162
Tabla 30.Caso de uso: CU23 - Descargar elementos multimedia	162

LISTADO DE APÉNDICES

	Pág
Apéndice A. Plan de pruebas	57
Apéndice B. Casos de prueba	63
Apéndice C. Incidencias	68
Apéndice D. Cuestionario SUS	70
Apéndice E. Manuales del usuario	75
Apéndice F. Manuales del desarrollador	103
Apéndice G. Implementación del prototipo	133
Apéndice H. Casos de uso	133

Resumen

Título: Plataforma software extensible para el despliegue, ejecución y visualización de algoritmos dedicados a la visión por computador *

Autores: Juan Sebastian Marcon Caballero **

Palabras claves: algoritmos dedicados, visión por computador, aprendizaje de máquina, despliegue, ejecución, visualización, compatibilidad, extensible, estructura de software.

Descripción: Hoy en día, la visión por computador e inteligencia artificial han permitido soportar y dar solución a un gran número de problemas, en diversas áreas del conocimiento. Muchos de estos métodos han evolucionado a aplicaciones robustas que operan con éxito en la industria. En su mayoría, las aplicaciones están relacionadas con la solución a problemas relacionados con el análisis de datos y la simplificación de tareas repetitivas. Las herramientas características de las ciencias de la computación están centradas en la solución de problemas de optimización y modelamiento de algoritmos de aprendizaje de máquina. Sin embargo estos algoritmos carecen de una estructura de software clara y uniforme, lo que conlleva a prototipos funcionales, pero con poca usabilidad, inclusive por parte de otros expertos en la misma área. Entre otras, estos algoritmos funcionales tienen múltiples dependencias de software que generan problemas de incompatibilidad y dificultad de operar en diferentes ambientes y sistemas operativos. De hecho, muchas de estas propuestas no evolucionan como aplicaciones tecnológicas debido a la dificultad existente en el proceso de despliegue y ejecución de los diferentes modelos que los componen, requiriendo así el diseño de software que ofrezca una infraestructura adecuada para el despliegue, la administración de la complejidad, el manejo de la heterogeneidad y la extensibilidad de la solución. En este trabajo se diseñó y se desarrolló la plataforma de software llamada *Deeptools* que permite desplegar algoritmos de aprendizaje de máquina y visión por computador, los cuales han sido entrenados para tareas específicas. El despliegue de estos algoritmos se hace mediante una arquitectura de micro-servicios, y el uso de contenedores *docker*s que permite la independencia de recursos y dependencias propias de las aplicaciones. Luego, se diseñaron vistas generalizadas que permiten visualizar y entender el comportamiento de las aplicaciones, con datos de entrenamiento, pero además con nuevas interacciones por parte de un usuario. La plataforma propuesta fue

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Fabio Martínez Carrillo, Ph.D. Codirector: Gabriel Rodrigo Pedraza, Ph.D.

validada con más de cinco algoritmos, desarrollados en el grupo de investigación *BIVL²ab*, mostrando robustez y flexibilidad para el despliegue de aplicaciones heterogéneas, así como usabilidad desde el punto de vista de usuario de la aplicación como el desarrollador del modelo.

Abstract

Title: Extensible software platform for the deployment, execution and visualization of dedicated computer vision algorithms *

Author: Juan Sebastian Marcon Caballero **

Keywords: dedicated algorithms, computer vision, machine learning, deployment, execution, visualization, compatibility, extensibility, software structure.

Description: Nowadays, computer vision and artificial intelligence support and provide solutions to a large number of problems in various areas of knowledge. Many of these methods have evolved into robust applications that operate successfully within the industry. The majority of the applications are related to data analysis and the simplification of repetitive tasks. The characteristic tools of computer science are focused on solving optimization problems and modeling machine learning algorithms. However, these algorithms lack a clear and uniform software structure, which leads to functional prototypes with little usability, even by domain experts. Moreover, these functional algorithms have multiple software dependencies that generate problems of incompatibility and difficulty to operate in different environments and operating systems. In fact, many of these proposals do not evolve as technological applications due to the existing difficulty in the process of deployment and execution of the different models that compose them, thus requiring the design of software that offers an adequate infrastructure for deployment, complexity management, heterogeneity management and extensibility of the solution. In this work, the software platform called *Deeptools* was designed and developed to deploy machine learning and computer vision algorithms, which have been trained for specific tasks. The deployment of these algorithms is done through a micro-services architecture, and the use of docker containers that allows the independence of resources and dependencies of the applications. Subsequently, generalized views were designed to visualize and understand the behavior of the applications with training data, as well as with new user interactions. The proposed platform was validated with more than five algorithms, developed in the *BIVL²ab* research group, showing robustness and flexibility for the deployment of heterogeneous applications, in addition to usability from the point of view of the application user and the model developer.

* Degree work

** Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering. Advisor: Fabio Martínez Carrillo, Ph.D. Co-Director: Gabriel Rodrigo Pedraza, Ph.D.

Introducción

Las nuevas tecnologías buscan acoplarse de forma natural en escenarios industriales, resultando su interacción transparente en muchas actividades y procesos cotidianos. El escalamiento de estas tecnologías ofrecen hoy en día nuevas funcionalidades que resultan naturales al nivel del usuario, permitiendo el prolongado uso de la aplicación. Por ejemplo, muchas de las aplicaciones de inteligencia artificial y visión por computador han ganado interés en el mercado gracias a los sistemas que permiten embeber estas aplicaciones, logrando que gran parte de su persistencia de un valor agregado, permitiendo una interacción natural y ganancia en el uso de estos sistemas, desde el nivel de usuario. Particularmente, las aplicaciones que incluyen modelos de aprendizaje de máquina han permitido soportar y dar solución a un sin número de problemas en diversas áreas del conocimiento y han sido usadas en un amplio número de campos ¹, incluyendo diagnóstico médico, comercio de acciones, control robótico, aviación, exploración espacial, leyes, percepción remota, descubrimientos científicos, juegos, entre otros. De hecho, muchas de estas aplicaciones ha logrado brindar un aporte diferencial, que permite una mejor adaptación en la sociedad y un uso intrínseco de cada una de sus funcionalidades, sobresaliendo de otras aplicaciones que cumplen el mismo objetivo técnico.

A pesar de su gran aplicabilidad y alta demanda, algunos algoritmos de visión por computador e inteligencia artificial no entran a su etapa de producción debido a la complejidad que lleva el despliegue. De hecho, muchos de estos algoritmos son basados en computación distribuida, lo cual requiere la factorización del código fuente ². La solución a lo anterior es la implementación de software especializado para cada algoritmo y el entendimiento de la topología que impli-

¹ Iyad Rahwan. *Argumentation in Artificial Intelligence*. Springer, 2009, <https://link.springer.com/content/pdf/10.1007/978-0-387-98197-0.pdf>.

² William Gropp y col. *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. MIT press, 1999.

ca un correcto entendimiento de la aplicación ³. Sin embargo, la creación de cada aplicación puede evolucionar en múltiples soluciones y requerir la convergencia de diversas tecnologías de desarrollo, lo cual conlleva una evaluación tecnológica rigurosa y exhaustiva para garantizar la compatibilidad de sus componentes ⁴. este enfoque de desarrollo en cada aplicación particular puede ser restrictivo, lo cual impone un paradigma de estandarización de tecnologías para el desarrollo de aplicaciones específicas, de alto nivel, que involucren nuevas tendencias en ciencias de la computación. De hecho, los recientes avances en aprendizaje de máquina, aprendizaje profundo y visión por computador han impulsado nuevas metodologías de desarrollo que permitan la generalización de componentes, y el desarrollo ágil de aplicaciones, teniendo como factor común los datos utilizados y las respuestas que evidencian el desempeño en cada aplicación. A pesar de que dichos estándares son cada vez mas comunes, aún existe la necesidad en desarrollar soluciones que permitan el despliegue ágil de prototipos y la validación de diversas aplicaciones, que compartan un marco de modelamiento en común, esto nos da a entender que su comportamiento es similar, dando pie a la estandarización de estas herramientas.

Esta fase del desarrolló de software es imprescindible dada la necesidad de acceder a este tipo de herramientas desde cualquier parte del mundo, además aporta a la comunidad una nueva herramienta lista para usar. De hecho esta fase del desarrollo de software no garantiza la centralización de estas herramientas, dando la necesidad de encontrar un software centralizado capaz de desplegar este tipo de algoritmos.

En este trabajo se desarrolló *DeepTools*, una herramienta web que permite desplegar y ejecutar algoritmos de inteligencia artificial y visión por computador, permitiendo la visualización y el monitoreo de estas estrategias mediante un aplicación web. Esta aplicación cuenta con una interfaz de usuario amigable, que permite la interacción de los modelos desarrollados con

³ Pei Fan y col. “A topology-aware method for scientific application deployment on cloud”. En: *International Journal of Web and Grid Services* 10.4 (2014), pág. 338. DOI: 10.1504/ijwgs.2014.064937.

⁴ Rafael Rodríguez Puente Carlos Abel Capeáns Hurtado. “Compatibility technology in of Enterprise Systems implementation”. En: (2015).

dos tipos de usuarios: 1) el desarrollador de estos algoritmos y 2) el usuario que desea utilizar estos algoritmos. Tanto el usuario como el desarrollador tienen tipos de acceso diferentes. El usuario, en el rol de visitante, únicamente puede visualizar e interactuar con los algoritmos y programas existentes en *DeepTools* según los parámetros de cada uno de estos. Por otro lado, el desarrollador tiene un perfil con información personal requerida, además de ser el único tipo de usuario que puede agregar los algoritmos de visión por computador e inteligencia artificial. La ejecución de este proyecto brindó una solución específica para fortalecer la difusión científica, permitiendo desplegar, ejecutar y visualizar resultados y desempeños de algoritmos relacionados con la inteligencia artificial, el aprendizaje de máquina y la visión por computador. El prototipo desarrollado se logró llevar a un nivel de maduración **TLR-4**, contando con pruebas en un ambiente real, así como también, generando los documentos de usuario y desarrollador para su uso, despliegue y posterior escalamiento. La herramienta *DeepTools* fue sometida al comité de propiedad intelectual de la institución para solicitar su respectivo registro.

1. Planteamiento y justificación del problema

El despliegue de software es una parte fundamental en el desarrollo de arquitecturas computacionales, considerando que en esta fase se pone a disposición una herramienta para su uso en un ambiente real. Sin esta fase muchas de las herramientas que conocemos al día de hoy no existirían. Los desarrolladores de soluciones basadas en software científico tienen un interés central en la funcionalidad de la solución, dejando de lado el proceso de ingeniería de software ⁵, y disminuyendo así la calidad de la solución en otros aspectos diferentes a su función. Además estas consideraciones no facilitan el uso de la solución a usuarios ajenos a la autoría del algoritmo. Solo pocos de los creadores de software científico no siguen esta tendencia, una encuesta sobre la práctica de la ciencia de la computación ⁶ mostró un pequeño conjunto de programas científicos que se destacaron tanto en términos de mejores prácticas de ingeniería de software como de altos estándares de reproducibilidad, esto constituye una problemática para la estandarización del desarrollo de software científico. Particularmente, el grupo de investigación *BIVL²ab* recientemente se ha enfocado a proponer múltiples soluciones científicas basadas en representaciones de aprendizaje profundo, inteligencia artificial y técnicas orientadas a la visión por computador. A pesar de su demostrado valor científico y demostrado carácter resolutivo, muchas de estas tecnologías presentan dificultades para ser visualizadas y validadas por expertos en otras áreas, así como también resulta difícil la generalización de las herramientas para ser representada en un catálogo de software donde múltiples tecnologías, dado esto se hace necesario una herramienta que permita que estas herramientas puedan converger y ser validadas como un nivel de maduración **TLR 3 y 4**.

⁵ Zeeya Merali. “Error: why scientific programming does not compute”. En: *Nature* 467.7317 (2010), págs. 775-777.

⁶ P. Prabhu y col. “A survey of the practice of computational science”. En: *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. 2011, págs. 1-12. DOI: 10.1145/2063348.2063374.

2. Objetivos

2.1. Objetivo general

Desarrollar una plataforma software extensible que permita el despliegue, ejecución y visualización de resultados de algoritmos computacionales basados en aprendizaje de máquina y visión por computador.

2.2. Objetivos específicos

1. Identificar los requerimientos de un ambiente de cómputo capacitado para el despliegue, la ejecución y la visualización de algoritmos de aprendizaje de máquina y visión por computador.
2. Modelar una arquitectura de software extensible que permita soportar los requerimientos encontrados.
3. Implementar un prototipo de software extensible para el despliegue, ejecución y visualización de algoritmos de aprendizaje de máquina y visión por computador.
4. Validar el prototipo de software con 3 algoritmos de aprendizaje de maquina y visión por computador desarrollados por los integrantes del grupo de investigación *BIVL²ab*.

3. Marco de referencia

A continuación se detallan algunos componentes, conceptos y tecnologías claves en el desarrollo del presente trabajo, permitiendo ser la base del conocimiento para el desarrollo de la solución propuesta ante el problema planteado.

3.1. Micro-servicios

Durante mucho tiempo el desarrollo de aplicativos se centraron en el funcionamiento autónomo e independiente. Estas características se lograron con el diseño sin modularidad (figura 1.a). Este tipo de sistemas son llamados monolíticos y constituyen una serie de problemas durante su desarrollo y mantenimiento⁷, aumentando los costos del aplicativo. La problemática más clara se da al momento de hacer modificaciones, estos cambios pueden afectar la funcionalidad completa del aplicativo, ya que este requiere una compilación e implementación completa de toda la aplicación⁸. En este sentido, para un despliegue industrial de una aplicación en funcionamiento esto puede acarrear latencias en los procesos de una entidad, así como también puede ocasionar pérdida de información, requiriendo además procesos independientes de verificación y adaptación a protocolos de servicio.

Para suplir dicho esquema, hoy en día existen diferentes arquitecturas enfocadas a la modularización de funcionalidades o servicios aislados que se exponen a través de un paradigma *API*. Una de las más nombradas en la actualidad es la arquitectura de micro-servicios (figura 1.b), haciendo su nombre énfasis en la implementación de un sistema único como un conjunto de servicios pequeños, cada uno de los cuales se ejecuta en su propio proceso y se comunica con

⁷ Chris Richardson. “Pattern: monolithic architecture”. En: *Posjećeno* 15 (2014), pág. 2016.

⁸ Martin Fowler y James Lewis. “Microservices a definition of this new architectural term”. En: *URL: <http://martinfowler.com/articles/microservices.html>* (2014), pág. 22.

mecanismos ligeros ⁹. Esta arquitectura nos permite desplegar cada servicio de forma independiente, además pueden estar programados en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos ⁹.

Arquitecturas de software de un algoritmo de sustracción de fondo

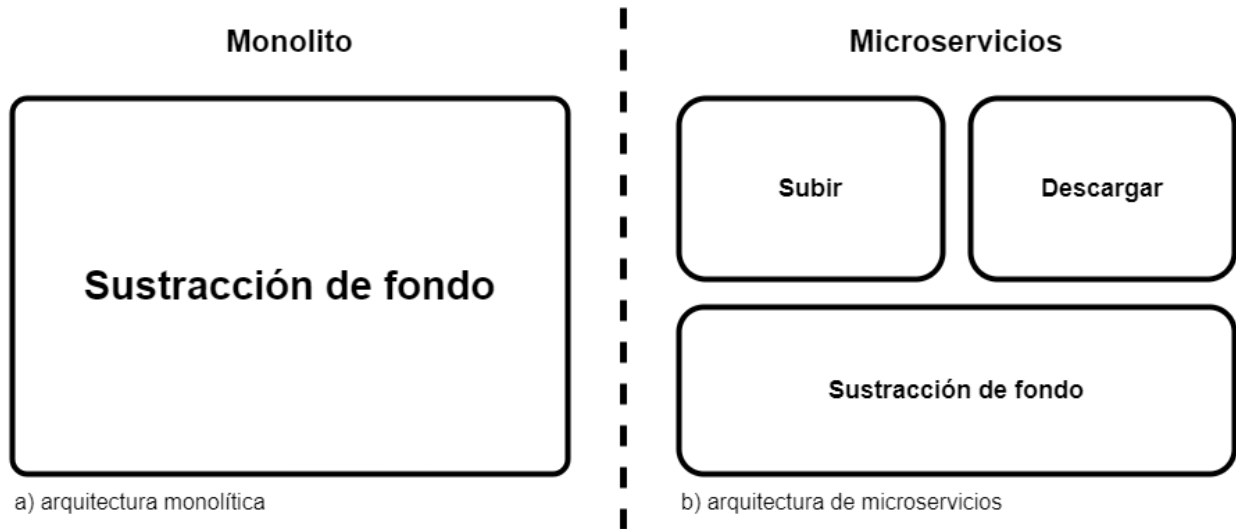


Figura 1. Arquitecturas de software de un algoritmo de sustracción de fondo. En a) se presenta una arquitectura monolítica enfocada a realizar los procesos de manera uniforme y en conjunto. En b) se representa la arquitectura basada en micro-servicios, los cuales reportan procesos independientes que permiten el despliegue independiente de aplicaciones.

3.2. Docker: la virtualización y los contenedores

Hoy en día han emergido nuevas aplicaciones que necesitan cada vez más recursos computacionales, creando de forma inherente una tendencia mayor en la demanda de hardware y requerimientos específicos de funcionamiento. Estas aplicaciones han sido diseñadas sobre esquemas de funcionamiento con demanda máxima, sin embargo, muchas de estas aplicaciones no utilizan

⁹ G. Granchelli y col. “Towards Recovering the Software Architecture of Microservice-Based Systems”. En: *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. 2017, págs. 46-53. DOI: 10.1109/ICSAW.2017.48.

todo el potencial demandado en el diseño, generando un desuso en los recursos. Además, algunos requerimientos especiales en el hardware pueden ser contraproducentes para la usabilidad masiva de estas aplicaciones.

Nuevos paradigmas de computo que brindan y administran *recursos virtuales* para la simulación de hardware base, han emergido en el estado del arte, permitiendo así que cada aplicación use la fracción necesaria de los recursos físicos y que muchas aplicaciones vuelvan a hacer operables, en distintas configuraciones físicas.

Esta simulación que permite compartir recursos físicos se le conoce como virtualización, derivándose en diferentes alternativas según las demandas de usabilidad y los requerimientos típicos de sistemas específicos. Por ejemplo, la virtualización tradicional tiene como base virtualizar sistemas operativos con ilusión de recursos dedicados. También la virtualización alternativa tienen una percepción de ambientes aislados, conocidas como maquinas virtuales (*VM*) y contenedores correspondientemente. Las *VM* se rigen bajo el concepto del hardware físico que transforma un servidor en muchos servidores. Su arquitectura (figura 2) se enfoca en la virtualización del hardware a diferencia de los contenedores. El hipervisor permite que varias máquinas virtuales se ejecuten en una sola máquina. Cada *VM* incluye una copia completa de un sistema operativo, la aplicación, los binarios y bibliotecas necesarias, que ocupan decenas de *Gigabites*¹⁰.

Por otra parte, los contenedores son un concepto a alto nivel que agrupa el código y las dependencias juntas. En este sentido, se pueden ejecutar varios contenedores en la misma máquina y compartir el núcleo del sistema operativo con otros contenedores, cada uno de los cuales se ejecuta como procesos aislados en el espacio del usuario. Los contenedores ocupan menos espacio que las *VM*, pueden manejar más aplicaciones y requieren menos sistemas operativos.

Particularmente, *Docker* es un proyecto enfocado a la automatización del despliegue de apli-

¹⁰ James E Smith y Ravi Nair. “The architecture of virtual machines”. En: *Computer* 38.5 (2005), págs. 32-38.

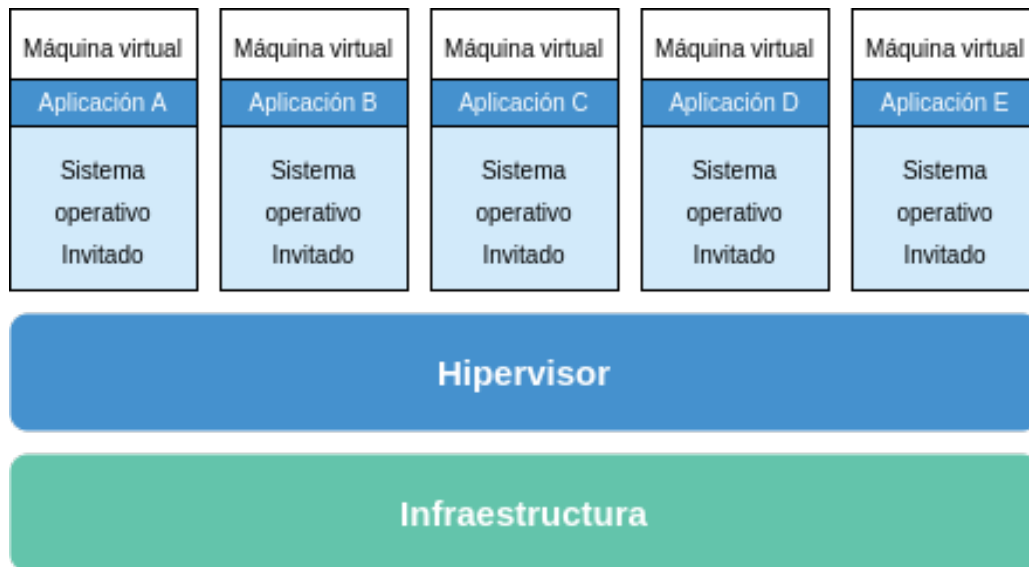


Figura 2. Arquitectura lógica de las máquinas virtuales *VM*. Virtualización de hardware por aplicación, requiere de un *SO* por aplicación

caciones virtualizadas dentro de contenedores de software aislados ¹¹¹². La tecnología de *Docker* se centra en los requisitos de los desarrolladores y operadores de sistemas para separar las dependencias de las aplicaciones de la infraestructura ¹²¹³. Los contenedores y las *VM* tienen beneficios similares de aislamiento y asignación de recursos, pero funcionan de manera diferente, ya que los contenedores virtualizan el sistema operativo en lugar del hardware ¹².

¹¹ C. Anderson. “Docker [Software engineering]”. En: *IEEE Software* 32.3 (2015), págs. 102-c3. DOI: 10.1109/MS.2015.62.

¹² Inc Docker. *What is a Container?* URL: <https://www.docker.com/resources/what-container>. (accessed: 26.02.2020).

¹³ James Turnbull. *The Docker Book: Containerization is the new virtualization*. James Turnbull, 2014.

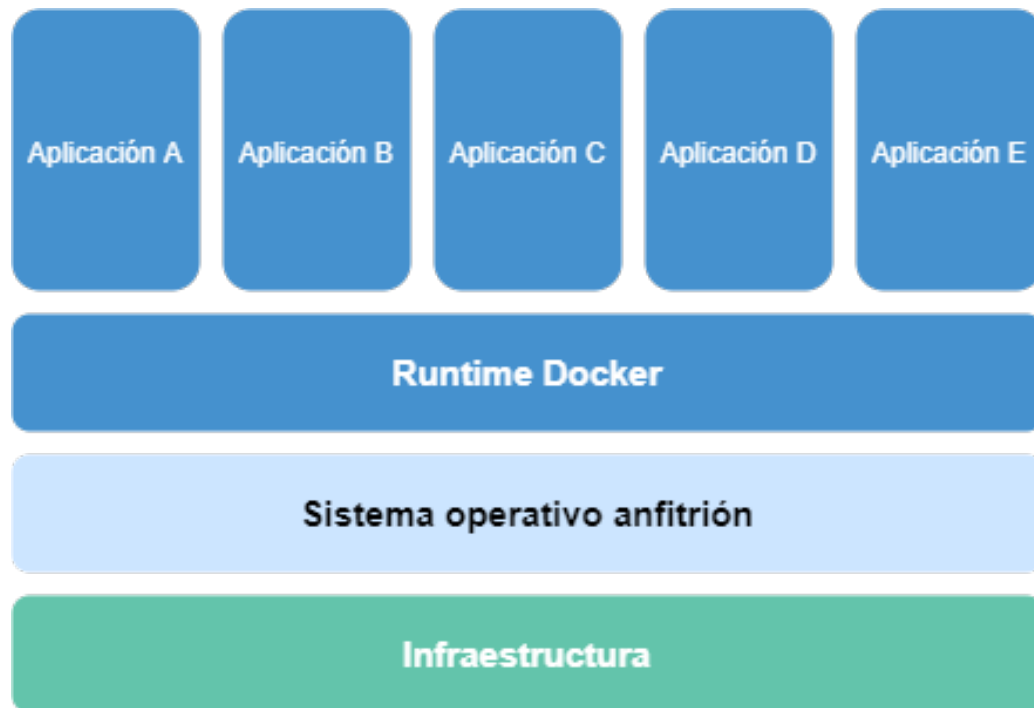


Figura 3. Arquitectura lógica de los contenedores *Docker*. Virtualización de software por aplicación, solo se requiere el *SO* de anfitrión.

3.3. Procedimientos remotos: *gRPC*

Las llamadas a procedimientos remotos (*RPC*) son un paradigma útil para proporcionar comunicación a través de una red entre programas escritos en un lenguaje de alto nivel ¹⁴. El modelo atraviesa dos procesos: el proceso del cliente y un proceso del servidor. El cliente envía primero un mensaje de llamada al proceso del servidor y espera un mensaje de respuesta. El mensaje de llamada incluye los parámetros del procedimiento, y el mensaje de respuesta incluye los resultados del procedimiento. Una vez recibido el mensaje de respuesta, se extraen los resultados del procedimiento y se reanuda la ejecución de la llamada. En el lado del servidor, un proceso está inactivo esperando una llamada del cliente, una vez llega el proceso servidor extrae los los

¹⁴ Asociación de Maquinaria Informática, ed. *Implementación de llamadas a procedimientos remotos*. DOI: 10.1145/800217.806609.

parámetros del procedimiento, calcula los resultados, envía un mensaje de respuesta y queda a la espera el siguiente mensaje de llamada.

Existen implementaciones basadas en *RPC* con características deseables, tales como la solución *gRPC*, que es comúnmente usada para la conexión de micro-servicios basado en *Protocol Buffers* y *HTTP/2*, permitiendo conexiones con muy baja latencia para grandes cantidades de datos. Esta tecnología proporciona características como autenticación, transmisión bidireccional y control de flujo, enlaces bloqueantes o no bloqueantes, cancelaciones y tiempos de espera, además es independencia del lenguaje(multilenguaje) y multiplataforma ¹⁵.

3.4. Estado del arte: soluciones de software para el despliegue de aplicaciones científicas

En la actualidad existen múltiples plataformas que permiten desplegar modelos de aprendizaje automático como *TensorFlow Serving* ¹⁶. Este es un sistema flexible y de alto rendimiento diseñado para entornos productivos, facilita la implementación de nuevos algoritmos. Esta plataforma tiene una implementación *REST* al momento de desplegar los distintos modelos *ML*, en otras palabras permiten crear un punto final para cada implementación de un modelo *ML* permitiendo mantener la arquitectura cliente-servidor. Teniendo en cuenta su arquitectura de base, esta solución requiere ser implementada con *TensorFlow* ¹⁷, limitando las tecnologías de desarrollo. De hecho, con los recientes avances en las estrategias computacionales para aprendizaje profundo, existen diversas librerías y alternativas bastante específicas propuestas desde los autores, que no necesariamente tienen una implementación en *Tensorflow* o requieren múltiples dependencias de otras librerías de aprendizaje de máquina. También existen propuestas de algoritmos, implementados en esquemas híbridos, que usan alternativas con diferentes dependencias.

¹⁵ Cloud Native Computing Foundation. *What is a gRPC?* URL: <https://grpc.io>. (accessed: 15.03.2021).

¹⁶ *MLflow*. 2022.

¹⁷ *Tensorflow*. 2022.

Así, muchas de estas soluciones no pueden ser exploradas o desplegadas en *TensorFlow Serving* debido a sus requerimientos de base.

Otra plataforma de gran acogida en la comunidad es **Cortex**¹⁸, que es una herramienta de código abierto flexible, que brinda servicios de monitoreo de modelos, clusterización, integración continua, distribución continua, servidor de modelos, compatibilidad con distintos servicios de computación en la nube como *AWS*, *Azure DevOps*, entre otros. Esta plataforma incluye servicios que soportan procesamiento en *GPU* y *CPU*, además como la implementación de múltiples *frameworks* (*TensorFlow*, *Pytorch*, *scikit-learn*, *Keras*, entre otros), para lograr así una rápida interacción en las soluciones implementadas. Esta herramienta sin embargo puede presentar algunas limitaciones para la adopción en equipos multidisciplinarios que desarrollan algunas estrategias aisladas que requieren componentes muy específicos, escritos en librerías que no son fácilmente adaptables. Además su arquitectura basada en micro-servicios depende de un buen entrenamiento del equipo para que no resulten aislados en las soluciones o no puedan interactuar con los demás componentes de la plataformas.

Algunas de las soluciones más rústicas pero comunes, es la implementación de servicios web compatibles con el modelo desarrollado, un ejemplo común ocurre con modelos de aprendizaje automático desarrollados con *Python*, en el cual usan librerías como **Flask**¹⁹ para envolver los modelos. Estos desarrollos permiten facilitar la creación del servicio web, requiriendo una implementación específica para cada modelo (figura 4), con este tipo de arquitecturas no se aíslan los distintos algoritmos limitando el uso de tecnologías, ya que requiere compatibilidad de todas las herramientas, Este tipo de implementaciones se suelen complementar con servicios de computación en la nube como *Microsoft Azure*²⁰ que permiten desplegar contenedores y configurar llamadas *API*, también está compuesta por múltiples productos y servicios que

¹⁸ *Cortex. Cloud infrastructure for machine learning at scale. 2022.*

¹⁹ *Flask. Web development, one drop at a time. 2022.*

²⁰ *Microsoft Azure. Cloud Computing Services: Microsoft Azure. (accessed: 12.03.2021).*

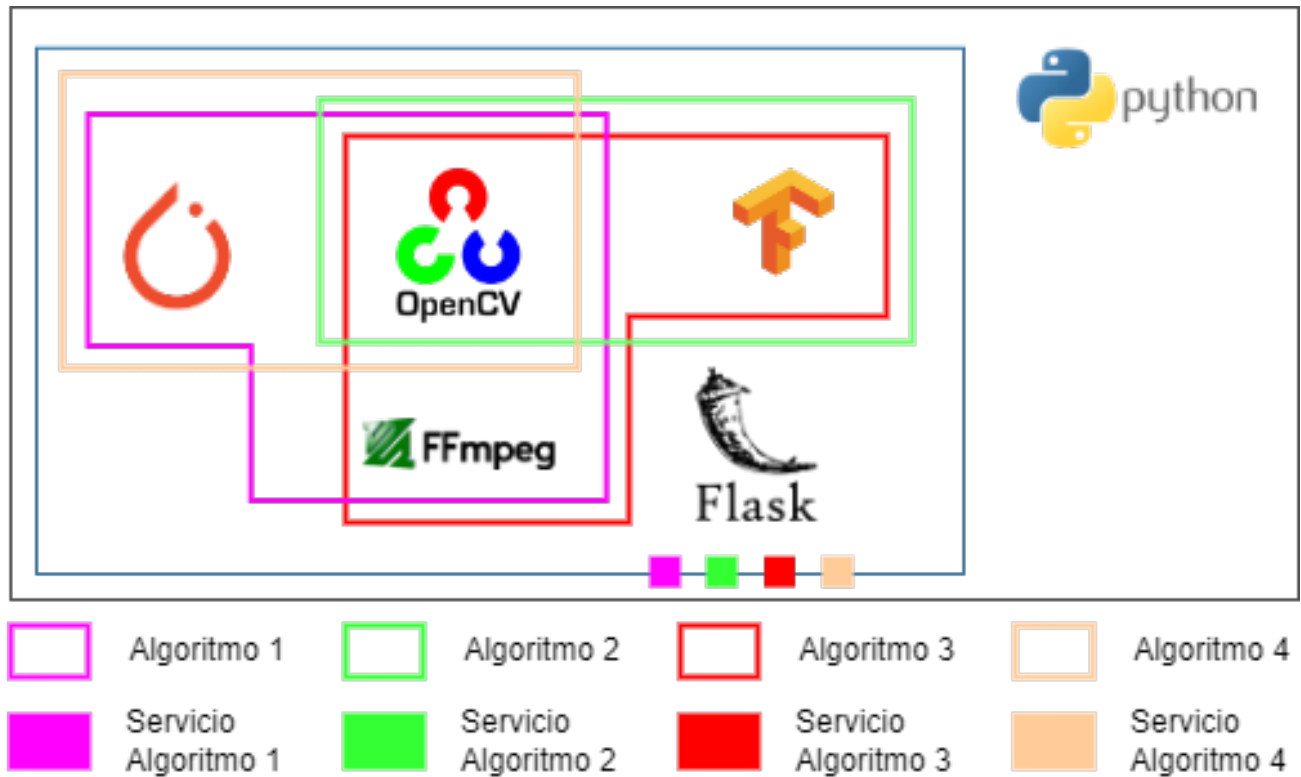


Figura 4. Arquitectura de una implementación de servicios web con la librería *Flask*. Aislación entre las tecnologías de los algoritmos nula, obliga a usar tecnologías compatibles entre algoritmos.

propone nuevas soluciones y resuelve dificultades comunes. Este tipo de productos son utilizados normalmente para implementar arquitecturas basadas en micro-servicios con el fin de proveer una alta escalabilidad y disponibilidad de las soluciones.

También, en el estado del arte se han destacado soluciones más orientadas en el despliegue de modelos *ML*, como *Algorithmia*²¹ el cual provee un servicio de despliegue, administración y monitoreo de tráfico de modelos *ML* servidos desde un *API*, implementando *MLOps* para garantizar un flujo repetible, escalable y eficiente durante el ciclo de vida del modelo *ML*.

²¹ Algorithmia. *What is Algorithmia?* (accessed: 26.02.2021).

Existen plataformas similares tal como *MLFlow*²², su factor común se basa en la búsqueda de administrar el ciclo de vida de un modelo *ML*, incluyendo la experimentación, reproducibilidad, implementación y centralización del modelo.

²² *MLflow. An open source platform for the machine learning lifecycle. 2022.*

4. Desarrollo del proyecto

En este capítulo se describe como desde una fase de análisis se planeó una arquitectura para dar a la solución *DeepTools* que permite desplegar algoritmos de aprendizaje de máquina y visión por computador, diseñados en diferentes tecnologías, para diferentes propósitos, y los cuales han sido validados en componentes en términos científicos en el grupo de investigación *BIVL²ab*. La arquitectura propuesta cuenta con tres capas basadas en micro-servicios para garantizar la escalabilidad del software. Además la solución incluye el uso de contenedores *Docker* que permite la independencia de recursos y dependencias propias de las aplicaciones. En general para el desarrollo de la arquitectura se utilizó *Django*, *Docker 20.10*, *gRPC 1.35* y *Redis 5.0* como las principales tecnologías de la capa de servidor, *MySQL 8.0* en la capa de datos y *React JS 16* en la capa de presentación.

4.1. Principales componentes de la solución

A continuación se mencionan los principales componentes definidos dentro de la solución, los cuales han permitido organizar el diseño, definir procedimientos claros de ejecución y lograr desarrollar una solución flexible que permite involucrar múltiples actores dentro de la plataforma, diversas soluciones, así como también, hacer efectiva la interacción y despliegue de diferentes servicios pretendidos ser expuestos por el grupo de investigación *BIVL²ab*.

4.1.1. Tipos de usuario Con base en los requerimientos de la solución planteada se identificaron perfiles relevantes para el ciclo de vida de la plataforma, partiendo de este se definieron los siguientes roles de usuario dentro del sistema:

1. **Usuario.** Este rol comprende el consumidor o público general que participa como visitante casual en la pagina web, puede visualizar los algoritmos activos y utilizarlos si lo requiere con los correspondientes elementos establecidos por el autor de este. Este usuario no

requiere ningún tipo de conocimiento técnico, además tampoco necesita conocer la lógica del algoritmo, ni mucho menos el marco tecnológico usado en su desarrollo. Este usuario es totalmente ajeno al desarrollo y despliegue, pero su intervención podría nutrir los distintos algoritmos desplegados, esto debido a que el número de ejecuciones es proporcional a los escenarios a los que se enfrenta el algoritmo aumentando la fiabilidad y estabilidad del mismo.

2. **Desarrollador.** Este rol enmarca el tipo de usuario que enriquecen la herramienta *DeepTools* con contenido relevante y actualizado en torno al área de la visión por computador, sus últimos desarrollos, algoritmos y acercamientos relacionados con esta temática. Aunque este usuario requiere conocimiento de ciertas tecnologías necesarias para el despliegue, este no tiene la necesidad de conocer un marco tecnológico completo, ni mucho menos ser un experto en desarrollo web, pues una de las metas de este proyecto es facilitar el trabajo de cualquier tipo de usuario creador de software científico en este proceso específico para la difusión científica.

4.1.2. Algoritmos y sus funcionalidades en la plataforma Un algoritmo es una pieza de código o conjunto de instrucciones que represente una solución a una problemática específica, pero en este caso para nosotros es un poco más acotada, cuando hablamos de algoritmo nos referimos a una pieza de código que implemente técnicas de IA y/o Visión por computador, esto debido a que este sistema fue diseñado solamente para estos algoritmos. Dada la naturaleza de los algoritmos durante el diseño del prototipo de los algoritmos se definieron distintos estados que definen el ciclo de vida de los algoritmos afectando su visibilidad y utilización en la plataforma:

- **Desplegado.** Es un algoritmo que está registrado en la plataforma y cumple con los requerimientos para ser usado, pero este aún no ha sido probado. En otras palabras está en su fase de pruebas. Este estado es un filtro a algoritmos que no cumplan con las pruebas menores mínimas (Ejecución *dummy*), por esto su aporte a la plataforma *DeepTools* es nulo. En este estado el algoritmo no merece ser visible, ni debe usarse, por ello estas

opciones solo están disponibles para su autor, el cual debe encargarse de probar que mediante las ejecuciones de este mismo que puede usarse.

- **Activo.** Es un algoritmo que anteriormente fue desplegado y fue probado en diferentes escenarios por su creador, esto para garantizar la integración de este con la plataforma. En esta etapa el algoritmo aporta valor a la plataforma *DeepTools*, este fue usado en distintos entornos, y merece ser probado en muchos mas, por ello sera visible para cualquier usuario de la plataforma, permitiendo la posibilidad de suscripción a aquellos interesados. En este estado el algoritmo puede ser usado por su creador y por los usuarios suscritos.
- **Desactivado.** Es un algoritmo que anteriormente estuvo activo pero por distintas razones fallo para un escenario especifico. Este caso ocurre cuando un algoritmo falla para cualquier usuario, el aplicativo se encarga de notificar a su creador de la falla y automáticamente cambia de estado, desactivándose hasta que su creador solucione la falla. Una vez solucionada la falla cambiara al estado ‘Desplegado‘ para garantizar que el cambio no fallara con pruebas *dummy*.

Además los algoritmos pueden ser accedidos y ejecutados según las necesidades y solicitudes de un usuario. Teniendo en cuenta esto, en el diseño de la arquitectura hemos definido los siguientes funcionalidades que puede tener un usuario:

- **Suscripción.** La suscripción representa el interés de un usuario hacia un algoritmos en especifico, bajo esta lógica los usuarios solo se van a suscribir a algoritmos que deseen utilizar. Un usuario puede suscribirse a muchos algoritmos y un algoritmo puede tener múltiples usuarios suscritos.
- **Ejecución.** La ejecución se refiere al registro que deja un algoritmos especifico en la base de datos de la plataforma. La cardinalidad de las ejecuciones es 1 a n, luego un usuario suscrito a un algoritmo puede tener muchas ejecuciones por algoritmo.

- **Visualización.** La visualización de un algoritmo se refiere la acción de observar los resultados de una ejecución de un algoritmo desplegado. Esta se limita a los elementos del algoritmo, dado esto solo se podrán visualizar algoritmos compatibles con los elementos preestablecidos por el desarrollador antes de su desplégue.

Por otra parte, algunos algoritmos de *IA* requieren analizar imágenes, audio, vídeos, entre otros tipos de datos para solucionar alguna problemática, además de esto la respuesta a este problema también requiere alguno de estos. Estos son llamados elementos y se refiere a todo dato que el usuario debe proveer al algoritmos y los datos que el algoritmo responde representado de una manera amena al usuario. Cada algoritmo tiene distintos elementos los cuales son seleccionados previamente por el desarrollador. El usuario debe preestablecer los elementos, el tipo de dato y su cantidad antes de desplegar un algoritmo. Debido a la alta aplicabilidad de estos algoritmos se tiene que acotar estos elementos para la finalización de la primera versión de este prototipo. A continuación se mencionaran cada uno de los elementos que puede tener un algoritmo y sus tipos de datos:

- **Entrada:** Los elementos de entrada son los elementos multimedia necesarios para un algoritmo, estos van a ser proveídos por el usuario. Los elementos de entrada pueden ser imágenes o vídeos con cualquier formato compatible con el navegador *Google Chrome v89.0.4389.114 (jpg, png, h264)*.
- **Salida:** Los elementos de salida son los elementos multimedia generados por un algoritmo como resultado del algoritmo, estos van a ser proveídos por el algoritmo, dado esto el desarrollador deberá encargarse de que el algoritmo logre generar estos. Los elementos de salida pueden ser imágenes o vídeos con cualquier formato compatible con el navegador *Google Chrome v89.0.4389.114 (jpg, png, h264)*.
- **Gráficos:** Los elementos gráficos son todos lo elementos que deben ser representados como gráficos o diagramas, estos van a ser proveídos por el algoritmo, dado esto el desarrollador

deberá encargarse de que el algoritmo logre generar estos. Los elementos gráficos pueden ser cualquier gráfico en dos dimensiones representado por la librería *HIGHCHARTS v9.0.1*.

- **Respuesta:** Los elementos de respuesta son todos los elementos que no pueden ser representados por los anteriores, estos son proveídos por el algoritmo, por ello el desarrollador deberá pensar en como representar estos de manera ordenada y concisa para el usuario. Estos elementos es cualquier tipo de datos que puede representarse por medio de *Markdown* y *HTML*.

4.2. Planificación de requerimientos

En esta fase se definió el alcance del proyecto mediante reuniones con el director, codirector e integrantes del grupo de investigación *BIVL²ab* para analizar las distintas problemáticas en el despliegue de algoritmos de visión por computador. También se generaron de modelos y documentos que facilitaron proponer una visión general de la arquitectura de software y la producción del plan de acción de las siguientes fases de desarrollo.

4.2.1. Requerimientos funcionales En esta etapa se realizó el levantamiento de requerimientos funcionales para continuar con cumplimiento del objetivo general planteado, llevando a cabo reuniones integrantes del grupo de investigación *BIVL²ab* ya que cuentan con la experiencia en la creación de algoritmos de visión por computador. Estos requerimientos fueron validados por los directores del trabajo, los cuales cumplían con los objetivos planteados durante el desarrollo del proyecto.

Tabla 1. Requerimientos funcionales

Código	Nombre	Usuarios	
		D	U
RF00	Permitir el registro de nuevos usuarios	✓	-
RF01	Permitir iniciar sesión a los distintos usuarios	✓	✓

RF02	Permitir el cierre de sesión	✓	✓
RF03	Permitir la modificación de la cuenta	✓	✓
RF04	Permitir el despliegue de un algoritmo que cumpla con indicaciones establecidas en la documentación	✓	-
RF05	Permitir la visualización de los detalles de los algoritmos desplegados	✓	✓
RF06	Permitir eliminar algoritmos desplegados	✓	-
RF07	Permitir la visualizar de los diferentes algoritmos activos	✓	✓
RF08	Permitir la ejecución de los algoritmos desplegados a usuarios suscritos a un algoritmo	✓	✓
RF09	Proveer durante el despliegue de un algoritmo el seguimiento en tiempo real del proceso	✓	-
RF10	Notificar errores durante la ejecución de algoritmos	✓	✓
RF11	Permitir la suscribirse a un algoritmo activo	✓	✓
RF12	Permitir consultar y eliminar ejecuciones de su autoría	✓	✓
RF13	Permitir la descarga de elementos multimedia de ejecuciones de su autoría	✓	✓
RF14	Permitir ejecutar pruebas sencillas de un algoritmo para probar su funcionamiento	✓	✓
RF15	Proveer durante la ejecución de un algoritmo el seguimiento en tiempo real del proceso	✓	✓
RF16	Notificar al usuario cuando haya finalizado las ejecuciones de un algoritmo activo	✓	✓
RF17	Permitir agregar y eliminar usuarios suscritos a un algoritmo de su autoría	✓	-

D: Desarrollador U: Usuario

4.2.2. Definición de los requerimientos no funcionales También se definieron ciertos requerimientos no funcionales debido a especificaciones puntuales por los integrantes del grupo *BIVL²ab* durante la fase de análisis del prototipo.

Tabla 2. Requerimientos no funcionales

Código	Nombre
RNF01	Interfaz de usuario adaptable a distintos dispositivos.
RNF04	Interfaz de usuario reactiva.

Hoy en día la experiencia de usuario es una parte muy importante para el usuario final, al final del día el usuario tomara la herramienta con que se sienta mas cómodo, por esto se implemento un diseño web adaptativo, esta es una técnica de diseño web que busca la correcta visualización de una misma página en los distintos dispositivos con tamaños estandarizados. Su corazón se basa en la redimensión y reorganización de los componentes gráficos de la web dependiendo de el ancho de cada dispositivo permitiendo una mejor experiencia de usuario. La razón de este requerimiento se basa a la alta utilización de los dispositivos móviles, la accesibilidad de la plataforma sin la necesidad de un ordenador es una funcionalidad mas que deseable, podría considerarse considerarse un requisito para cualquier tipo de usuario.

Otra de las implementaciones mas deseados es el diseño web reactivo, esto debido a la experiencia que provee al usuario, al igual que el diseño web adaptativo es mas un requisito que un valor agregado para los usuarios.

4.3. Diseño del prototipo

Una vez se abordaron los requerimientos se prosiguió con la definición y realización de casos de uso, diagramas de actividades, diagramas de clases y arquitectura de software. Todo lo anterior constituye las bases para el diseño del prototipo. Con el propósito de detallar el comportamiento del sistema mediante la interacción con el usuario se definieron los siguientes casos de uso (Tabla 3). En la tabla se detallan las principales acciones comprendidas durante el diseño y que

resultan imprescindibles en la valoración de la solución para el despliegue de soluciones valoradas únicamente desde el punto de vista de ciencias de la computación. Con los diferentes casos de uso planteados se pudo aprovechar un diseño basado en suscripciones e interacción de una comunidad científica, la cual puede ser útil para validar diferentes aproximaciones desarrolladas en el grupo de investigación, sin acarrear costos mayores de cómputo o redundancia de la información y ejecución de soluciones parciales.

Tabla 3. Casos de uso

Id	Nombre	RF
CU00	Iniciar sesión (Tabla 7)	RF01
CU01	Cerrar sesión (Tabla 8)	RF02
CU02	Registrar usuario (Tabla 9)	RF00
CU03	Ver perfil (Tabla 10)	-
CU04	Editar perfil (Tabla 11)	RF03
CU05	Ver perfil de desarrolladores (Tabla 12)	-
CU06	Listar notificaciones (Tabla 13)	-
CU07	Notificar novedades (Tabla 14)	RF10, RF17
CU08	Suscribirse (Tabla 15)	RF11
CU09	Cancelar suscripción (Tabla 16)	RF12
CU10	Listar suscripciones (Tabla 17)	-
CU11	Desplegar algoritmo (Tabla 18)	RF04
CU12	Iniciar algoritmo (Tabla 19)	-
CU13	Detener algoritmo (Tabla 20)	-
CU14	Eliminar algoritmo (Tabla 21)	RF06
CU15	Administrar usuarios (Tabla 22)	RF18
CU16	Ejecutar algoritmo (Tabla 23)	RF08, RF15
CU17	Listar ejecuciones (Tabla 24)	-

CU18	Ver ejecución (Tabla 25)	RF09
CU19	Eliminar ejecución (Tabla 26)	RF13
CU20	Clonar ejecución (Tabla 27)	RF05
CU21	Filtrar algoritmos (Tabla 28)	RF06
CU22	Listar algoritmos (Tabla 29)	RF07
CU23	Descargar elementos multimedia (Tabla 30)	RF14

También, durante el diseño del prototipo se incluyeron diagramas de actores para identificar los principales usuarios que interactúan en el sistema, según los casos de uso reportados en la Tabla 3. En la figura 5 se evidencia el diagrama de actores, donde se muestra las similitudes entre los roles definidos. El usuario ‘Invitado’ es aquel usuario que no se ha registrado, el acceso de los módulos a este usuario es nulo, solo puede registrarse en la plataforma. Por otro lado el rol ‘Desarrollador’ hereda del rol ‘Usuario’, dándonos a entender que el rol ‘Desarrollador’ es una versión avanzada del rol ‘Usuario’, ya que tiene las mismas funcionalidades, y se agregan otras funcionalidades.

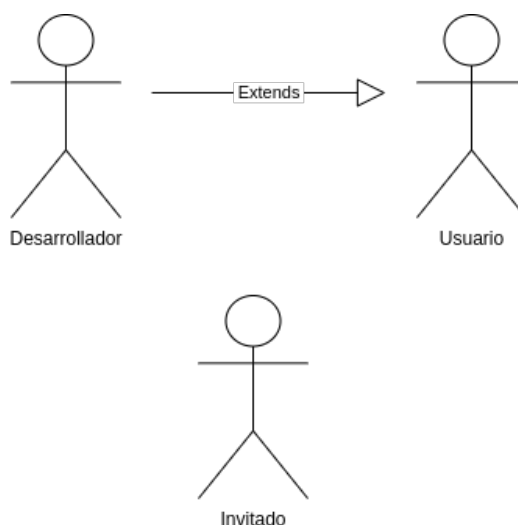


Figura 5. Diagrama de actores del sistema

Una vez identificados los actores y los requerimientos funcionales se pudo diseñar los respectivos diagramas de casos de uso. En las siguientes figuras se evidencia el diagrama de casos de uso con el fin de representar de manera simplificada y modularizada las funciones de la plataforma *DeepTools*. Por ejemplo en la Figura 6 se ilustra las funcionales pertenecientes al modulo de los algoritmos, siendo en este caso el rol de ‘Desarrollador’ esta encargado de administrar los algoritmos en cuanto el despliegue, inicialización y eliminación de los algoritmos, así como también estará encargado de administrar los suscriptores de sus algoritmos. Por otra en este modulo el ‘Usuario’ es responsable de las acciones que involucran la administración de sus suscripciones, este se encarga de crear, visualizar, listar, eliminar y usar las distintas suscripciones.

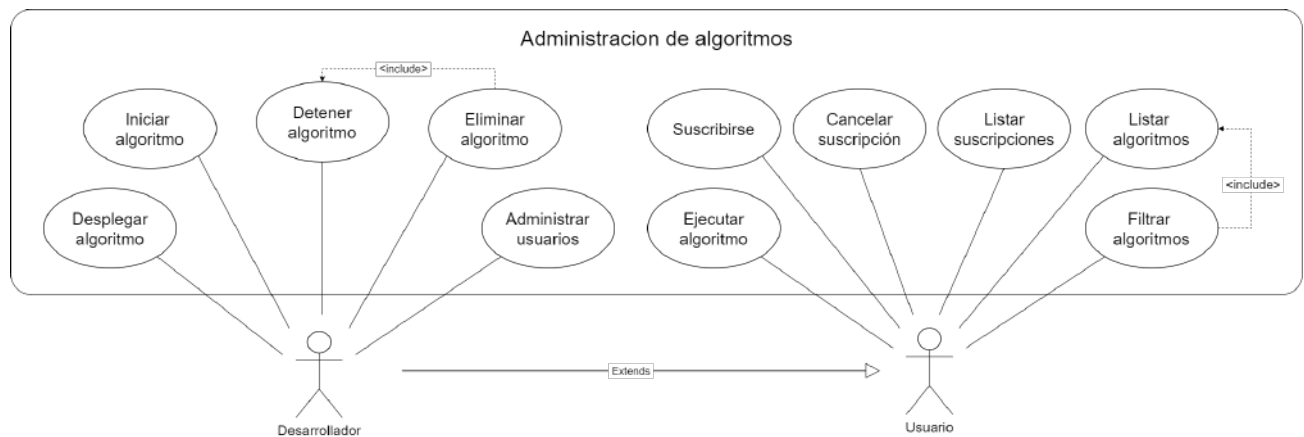


Figura 6. Diagrama de casos de uso - Modulo: Algoritmos

En el diagrama de la Figura 7 se muestra el modulo de ejecuciones, este modulo el principal protagonista es el ‘Usuario’, ya que es el ente de la plataforma que consume los productos y/o servicios desplegados. El ‘Usuario’ podrá crear, eliminar, clonar y visualizar las distintas ejecuciones de su autoría.

En el diagrama de la Figura 8 se muestra el modulo del usuario, en este modulo el principal protagonista es cualquier usuario registrado, ya que requiere información propia de cada usuario, tal como e-mail, nombre de usuario, entre otros.

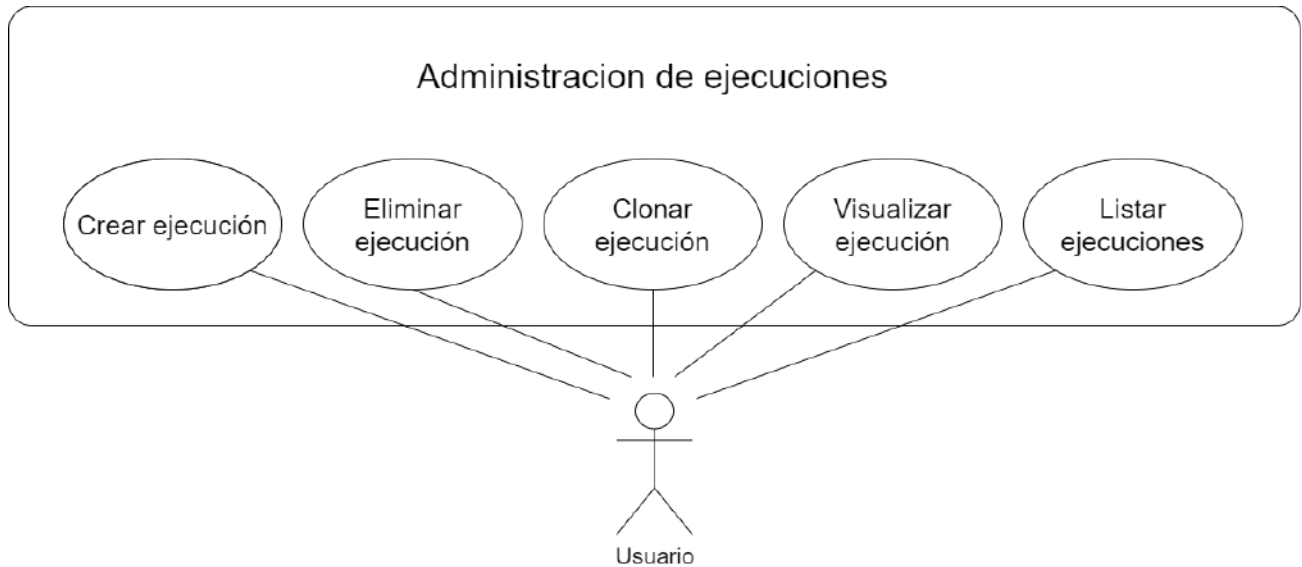


Figura 7. Diagrama de casos de uso - Modulo: Ejecuciones

4.4. Arquitectura propuesta

La arquitectura propuesta en este trabajo se ilustra en la Figura 9. Para este proyecto se realizó una arquitectura de 3 capas basada en micro-servicios, para simplificar la comprensión y la organización del desarrollo. De esta manera cada capa tiene su propia función, separando la capa lógica de la capa de aplicación y de la interfaz de usuario. Así en la arquitectura se añade una flexibilidad al diseño de la aplicación, el cual se puede construir y desplegar múltiples interfaces de usuario sin cambiar la lógica de la aplicación. A continuación se detallan cada una de las capas consideradas en esta arquitectura:

- **Capa de presentación.** encargada de la interacción con el usuario, esta fue construida con un SPA con *React JS* con el fin de brindar al usuario una plataforma mas dinámica y amigable, esta también cuenta con *Responsive Design* para ampliar el acceso desde distintos dispositivos. Su comunicación con la capa de servidor es mediante un *restful API* y conexiones *WebSockets* para proporcionar una comunicación con *Realtime*.
- **Capa del servidor.** Esta capa esta a cargo de varias funciones del ciclo de vida del

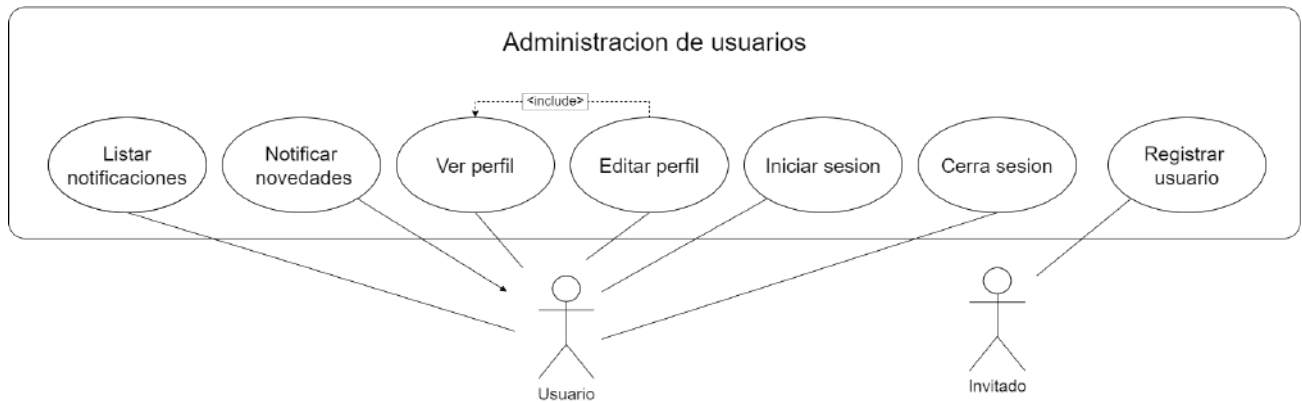


Figura 8. Diagrama de casos de uso - Modulo: Usuarios

aplicativo, como lo son:

- Responder al cliente: Se encarga de recibir las peticiones de la capa de presentación, procesa la información y envía las respuestas por el canal correcto.
- Peticiones a los contenedores: Esta capa tendrá la responsabilidad de crear peticiones a los contenedores y enviar las respectivas respuestas a la capa de presentación.
- Peticiones de información: Se comunicara con la capa de datos para obtener información de interés para la plataforma.

Esta capa es administrada por *Django* y *Django REST* para habilitar la compatibilidad con la capa de presentación. Se usaron contenedores *docker* con los diferentes servicios desplegados por los usuarios aprovechando la automatización durante el despliegue de aplicaciones virtualizadas, el enfoque en los requisitos de los desarrolladores y su aislamiento. Se utilizo *Redis* como canal para la transmisión de datos de alta velocidad, proporcionando una mejor experiencia en la mensajería instantánea.

La capa de servidor se dividió e 4 grandes módulos, cada uno cumple una tarea específica en el modelo de negocio del prototipo. El modulo **enrutador** es el encargado de llamar el modulo indicado dependiendo de la solicitud del cliente, es e encargado de la comunicación

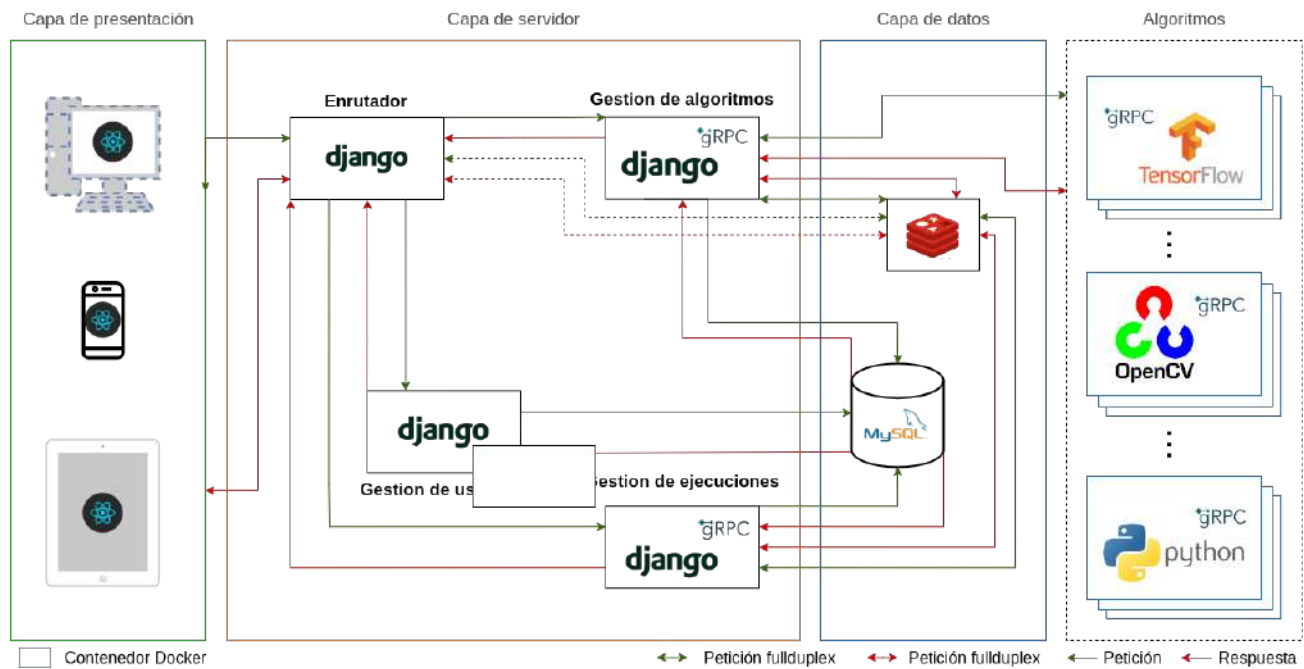


Figura 9. Arquitectura física de la plataforma *Deeptools*. Representación de las interacciones entre las capas de la arquitectura (capa de presentación, capa de servidor).

entre la capa de presentación y la capa de servidor.

El módulo de **gestión de usuarios** se encarga de administrar los datos de la sesión de cada usuario, sirviendo funciones tales como, inicio de sesión, cierre de sesión, registro, edición de perfil, cambio de contraseña, notificaciones, entre otras.

El módulo con la lógica mas pesada o compleja es el módulo de **gestión de algoritmos**, el cual se encarga de montar cada uno de los algoritmos desplegados por los desarrolladores, además gestiona los accesos de los distintos usuarios en los algoritmos.

Y por ultimo, pero el mas importante es el módulo de **gestión de ejecuciones**, el cual se encarga de probar, guardar y visualizar las distintas ejecuciones de los algoritmos, esto permite dar al usuario un entorno donde puede revisar cada una de las ejecuciones, evitando la fatiga del usuario guardar la información de la ejecución localmente. Esto es sumamente potente, ya que se puede acceder desde cualquier parte del mundo a los resultados de las ejecuciones.

- **Capa de datos.** Esta capa es un objeto que contiene toda la información dinámica que contiene el software, actúa como una matriz que almacena datos de manera organizada. Teniendo en cuenta las funcionalidades y requerimientos planteados para el prototipo, se planteo una base de datos relacional (*MySQL*), debido a la necesidad de usar transacciones, lo cual hace complejo el uso de base de datos no relacionales.

Para detallar el flujo que conecta estas tres grandes capas simularemos la ejecución de un algoritmo. Este flujo siempre empieza desde la capa de presentación (cliente). El cliente le notifica al servidor *Django* (capa de servidor) que desea ejecutar un algoritmo, El servidor *Django* verifica los datos con la capa de datos (base de datos), después de verificados inicia una conexión *full-duplex* entre el micro-servicio y el servidor *Django* si esta conexión se realiza con éxito devuelve al cliente la confirmación del inicio de la ejecución. Esta acción abre una conexión *full-duplex* con el servidor *Redis* y el cliente, creando un puente entre el micro-servicio del algoritmo desplegado y el cliente. Con el paso del tiempo el estado de la ejecución cambiara a completado, el cliente reaccionara a esta suscripción y mostrara los resultados de esta ejecución. Con este ultimo paso la ejecución se proclama finalizada y se notificara al cliente el nuevo estado de la ejecución.

5. Plan de pruebas

El plan de pruebas se realizó con el objetivo de verificar la calidad del prototipo desarrollado, así como también para verificar con cada uno de los requerimientos funcionales y no funcionales establecidos en el análisis propuesto por la metodología. Este plan de pruebas estará definido según la Norma ISO 29119 ²³, el cual esta documentado en detalle el anexo 1. A continuación se detallan las pruebas de integridad y las pruebas de aceptación y usabilidad llevadas a cabo en el presente proyecto.

5.1. Pruebas de integridad

El propósito de estas pruebas fue verificar si el prototipo de software se encuentra en una etapa de desarrollo madura. Con este plan se descubrieron errores que limitaban la funcionalidad del software, los cuales fueron debidamente atendidos. . En el enfoque de las pruebas de integridad se realizó pruebas de extremo a extremo de la construcción sin entrar en detalles más finos de los módulos.²⁴.

Para realizar estas pruebas se escogieron los principales casos de uso. Estos son parte vital para el funcionamiento general del software. Los casos de uso reconocidos como críticos para el software son:

1. CU00 - Iniciar sesión (Tabla 7)
2. CU02 - Registrar usuario (Tabla 9)
3. CU07 - Notificar novedades (Tabla 14)

²³ Trycore. *Norma ISO 29119*. (accessed: 5.01.2022).

²⁴ Vinod Kumar CHAUHAN. "Smoke Testing". En: *Int. J. Sci. Res. Publ* 4.1 (2014), págs. 2250-3153.

4. CU08 - Suscribirse (Tabla 15)
5. CU11 - Desplegar algoritmo (Tabla 18)
6. CU12 - Iniciar algoritmo (Tabla 19)
7. CU16 - Ejecutar algoritmo (Tabla 23)
8. CU18 - Ver ejecución (Tabla 25)

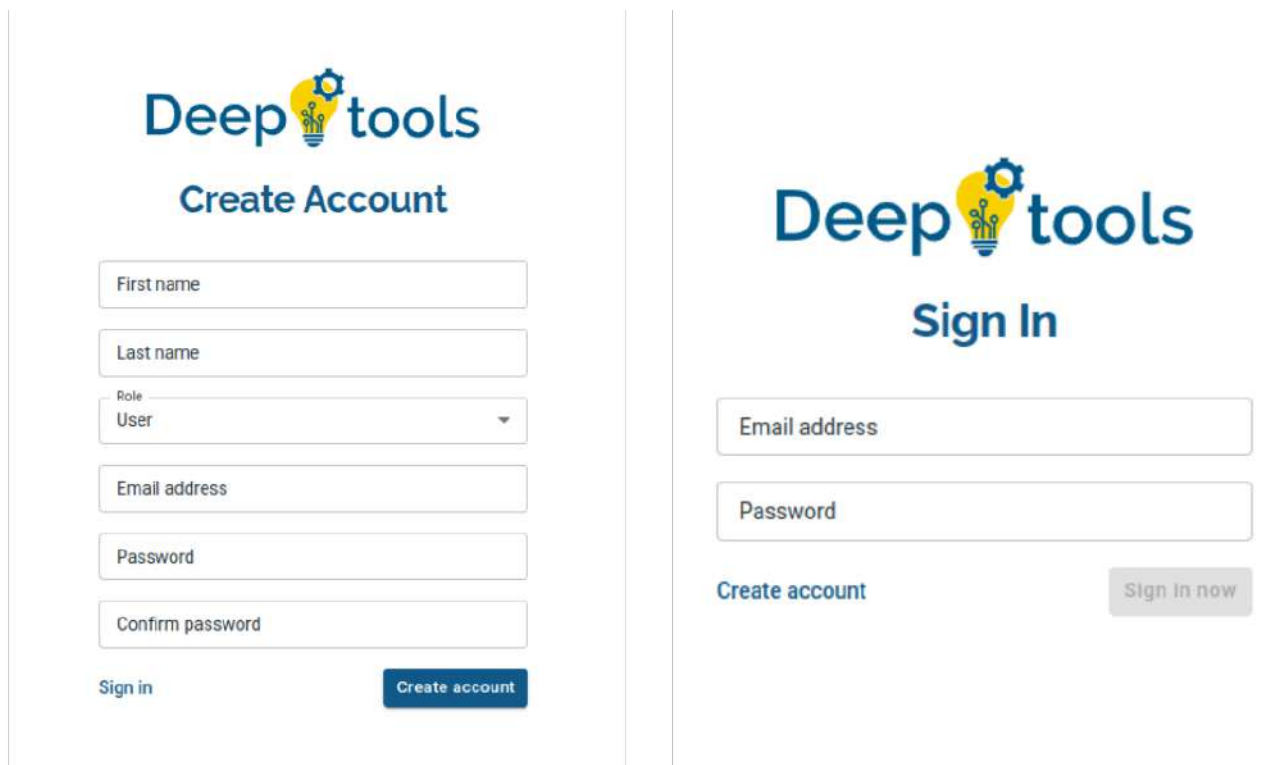
Dichos casos de uso representan el flujo básico requerido para su funcionamiento en otras palabras, los casos de usos que requieren más atención dada su importancia en el software. De los 8 casos de uso seleccionados anteriormente 6 de estos tienen una dependencia directa con CU00 (Tabla 7) y este tiene una dependencia con CU02 (Tabla 9). Luego los datos necesarios para crear un usuario son mínimos, información (nombre, email, contraseña) que cualquier internauta tiene a la mano.

Estos casos de uso representan el inicio de la plataforma *Deeptools*, nos lleva al inicio del flujo completo del prototipo, mencionado esto se debe llenar el formulario (Figura 10.a) para realizar el registro de usuario, una vez que se realizó este proceso se continuó completando el siguiente formulario (Figura 10.b) con el fin de iniciar sesión y continuar con los demás casos de uso.

Durante la ejecución de estas pruebas los usuarios reportaron algunos errores (Tabla 4) los cuales contribuyeron a la mejora del prototipo. Este se produjo gracias a la falta de validación del campo email, el cual debe ser exclusivo para cada usuario.

Tabla 4. Incidencias registrando un usuario

Código	Nombre	Descripción
B01	No se muestra <i>feedback</i> con usuario ya registrados.	Al registrar un usuario ya existente no se muestra error, la pantalla no cambia en absoluto.



a) Registro de usuario (CU02)

b) Inicio de sesión (CU00)

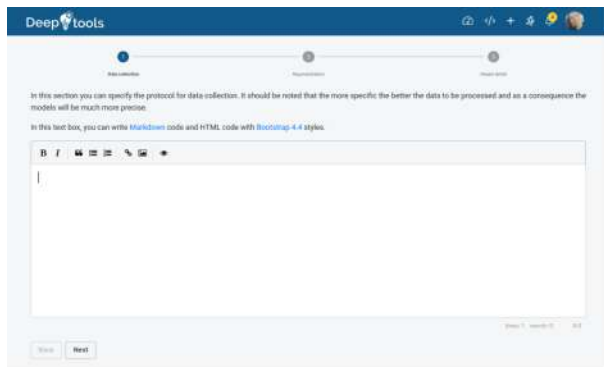
Figura 10. Inicio del flujo básico de la plataforma *Deeptools*. a) Interfaz gráfica para registrar usuarios en cumplimiento de CU02 (Tabla 9). b) Interfaz gráfica para iniciar sesión en cumplimiento de CU00 (Tabla 7).

Una vez agotada estas pruebas, se continuó con CU11 (Tabla 18) para los usuario desarrolladores. Cabe destacar que el objetivo principal de estos es poblar el software con distintos algoritmos, y no la utilización algoritmos que no son de su autoría.

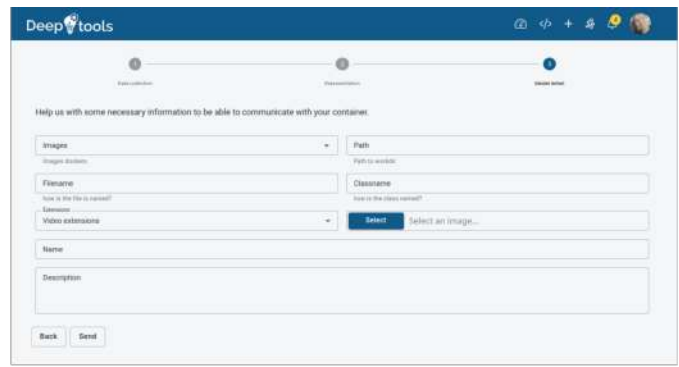
Todos los usuario reportaron un error (Tabla 5) durante el proceso de compilación del algoritmo (Figura 11.d), aunque el algoritmo se desplegaba no se mostraba el *feedback* correcto, dando al usuario una mala experiencia de usuario, dado el desconocimiento del estado del proceso de compilación. Este error se generaba por falta de configuración en las conexiones *WebSockets*. Este inconveniente fue debidamente atendido en la configuración del sistema y detallado en los manuales de usuario, que reposan como anexo en este documento.

Tabla 5. Incidencias durante el despliegue de un algoritmo

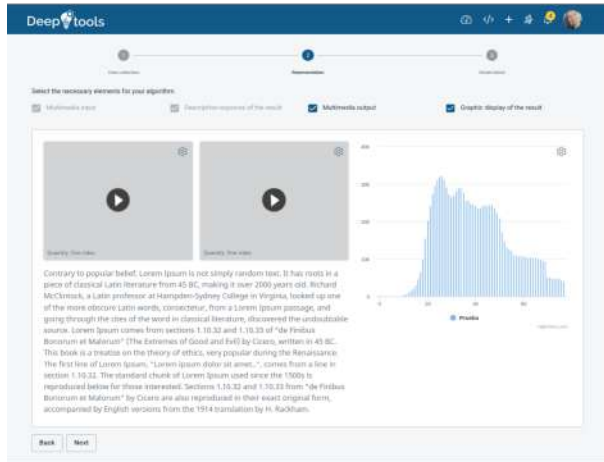
Código	Nombre	Descripción
B05	No se muestra una barra de progreso.	Al momento de desplegar un algoritmo no se muestra la barra de progreso en tiempo real.



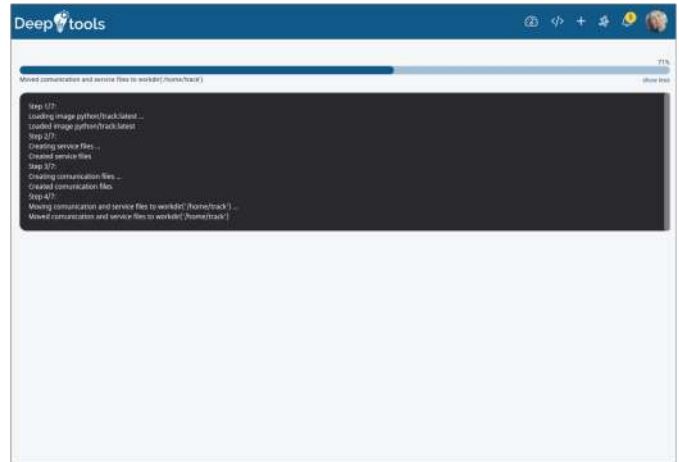
a) Paso 1: Registro del protocolo de toma de datos



c) Paso 3: Toma de metadatos del algoritmo.



b) Paso 2: Selección de los elementos del algoritmo



d) Paso 4: Proceso de compilación del algoritmo

Figura 11. Despliegue de un algoritmo. a) Registro del protocolo de toma de datos, se refiere al registro del paso a paso para la toma de datos compatibles con el algoritmo. b) Selección de los elementos del algoritmo. c) Toma de metadatos del algoritmo. d) Proceso de compilación del algoritmo: Seguimiento paso a paso durante la compilación del algoritmo

Al finalizar el despliegue se realizó CU12 (Tabla 19) para exponer el algoritmo desplegado, con esta acción se libera el algoritmo para ser usado por los usuarios suscritos a este.

Los usuario se suscribieron (Tabla 15) para poder usar un algoritmo, esto habilito la posibilidad de realizar distintas pruebas o ejecuciones del algoritmo. Esto nos dio paso a probar CU16 (Tabla 23), este nos mostró una barra de progreso (Figura 12) y información del proceso de ejecución en tiempo real.

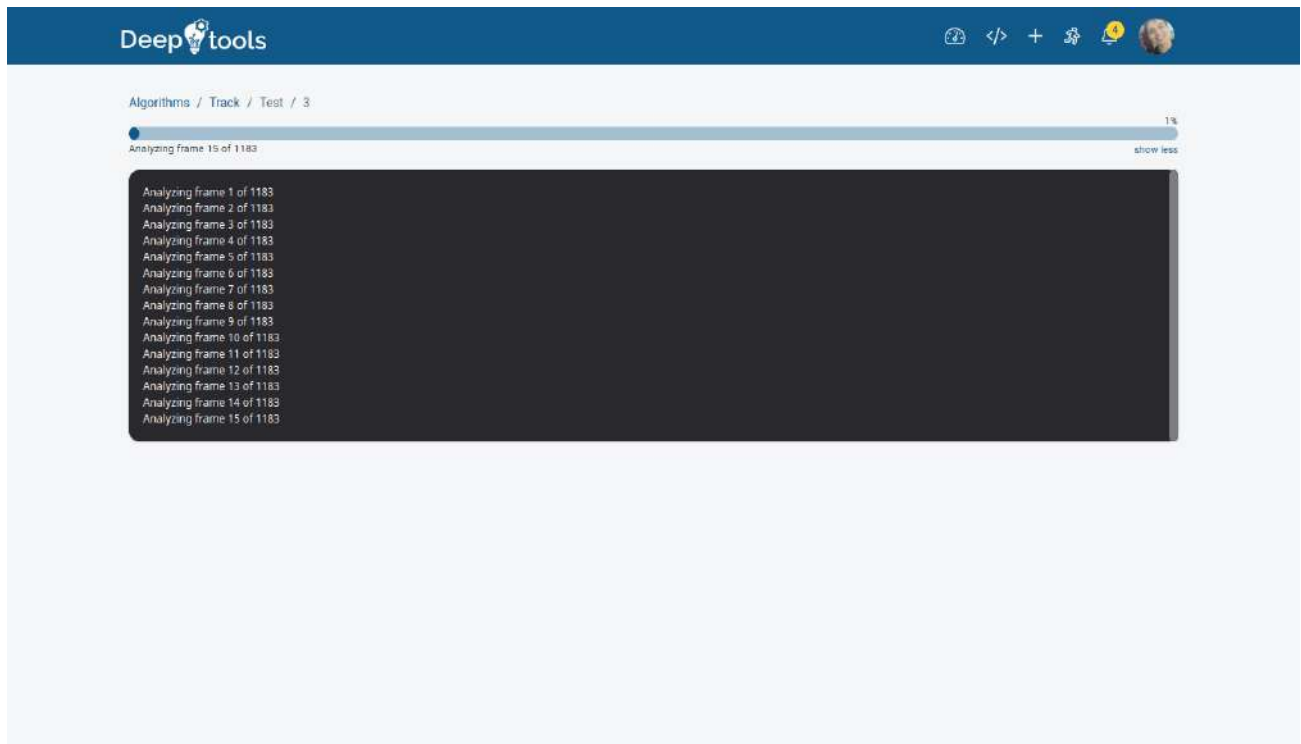


Figura 12. Ejecución de un algoritmo. Barra de progreso en tiempo real.

Cuando inicio CU16 (Tabla 23) se desplegó un observable listo para notificar novedades (Tabla 14), el software se encargo de notificarle el proceso que recorrió su prueba, y le notifico cuando finalizo. Después de esto los usuarios reportaron un error (Tabla 6) al momento visualizar los detalles de la ejecución (Tabla 25). Es importante aclarar que esta prueba de integridad es la que se realizó con mayor frecuencia en el software, pero a medida que se realizaba el desarrollo del software se estaban haciendo pruebas de integridad más detalladas para probar funcionalidades

pequeñas a medida que se iban desarrollando.

Tabla 6. Incidencias durante la visualización de los resultados de una ejecución

Código	Nombre	Descripción
B08	No se ven los detalles de la ejecución del algoritmo.	Al ver los detalles de la ejecución de un algoritmo, no se muestra información alguna, se ve una pantalla en blanco.

Las limitaciones y los reportes realizados por los usuarios fueron debidamente atendidos y solucionados. Esto conlleva a mejorar la funcionalidad del sistema, pero además todos los procedimientos llevados a cabo fueron documentados, como se menciona en el anexo 3.

5.2. Pruebas de aceptación

El objetivo de las pruebas de aceptación es verificar si el prototipo se creó según las normas, criterios y requisitos especificados antes de ser entregado a los usuarios finales.²⁵ Para ello, estas pruebas se realizaron con 3 integrantes del grupo *BIVL²ab*. En este caso, cada integrante lidera un proyecto de investigación, implementando una solución de inteligencia artificial y visión por computador, según las necesidades dispuestas en el proyecto. Entonces, los estudiantes se les solicitó desplegar el algoritmo en la plataforma desarrollada, teniendo como indicaciones únicamente el manual de usuario, propuesto para la aplicación. Además, también se consideró un cuarto usuario encargado de ejecutar estos algoritmos.

1. **Usuario 1, 2, 3** → Desarrollador
2. **Usuario 4** → Usuario

En una primera instancia se solicitó realizar a los 4 usuarios 3 distintas tareas:

²⁵ Abhijit A SAWANT, Pranit H BARI y PM CHAWAN. "Software testing techniques and strategies". En: *International Journal of Engineering Research and Applications (IJERA)* 2.3 (2012), págs. 980-986.

1. Registro
2. Inicio de sesión
3. Actualización de datos
4. Validar la actualización de datos

Una vez finalizadas las tareas de autenticación se continuo con el siguiente flujo:

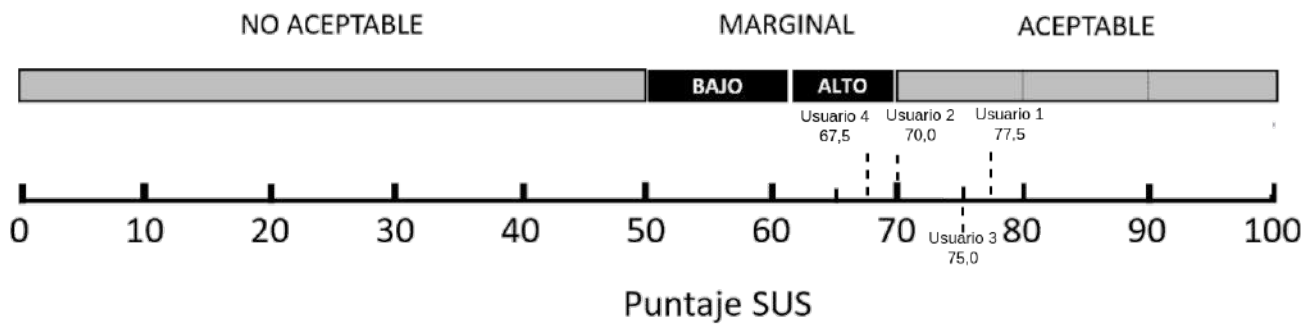
1. Desplegar un algoritmo → **Usuario 1, 2, 3**
2. Iniciar un algoritmo → **Usuario 1, 2, 3**
3. Listar los algoritmos → **Usuario 4**
4. Suscribirse a tres algoritmo → **Usuario 4**
5. Detener un algoritmo → **Usuario 2**
6. Eliminar un algoritmo → **Usuario 1**
7. Listar mis suscripciones → **Usuario 4**
8. Ejecutar un algoritmo → **Usuario 4**
9. Ver los detalles y resultados de una ejecución → **Usuario 4**
10. Eliminar una ejecución → **Usuario 4**

5.2.1. Prueba de usabilidad del sistema En la prueba de usabilidad se utiliza el cuestionario del sistema de escala de usabilidad (*System Usability Scale*) que consiste en aplicar 10 preguntas a los usuarios, a estas respuestas se aplica la fórmula de SUS. El valor no tiene una interpretación porcentual, debe clasificarse según la escala de usabilidad.

En el siguiente gráfico podemos observar los puntajes de los usuarios según la puntuación SUS (Figura 13).²⁶

Figura 13. Resultados - Escala de SUS

Preguntas	Usuario 1	Usuario 2	Usuario 3	Usuario 4
Creo que usaría el sistema Deeptools frecuentemente	3	4	4	2
Encuentro el sistema Deeptools innecesariamente complejo	1	2	2	2
Creo que el sistema Deeptools fue fácil de usar	4	5	4	4
Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar el sistema Deeptools	3	3	2	2
Las funciones del sistema Deeptools están bien integradas	5	4	3	3
Creo que el sistema Deeptools es muy inconsistente	1	2	1	1
Pienso que la mayoría de la gente aprendería a usar el sistema Deeptools rápidamente	5	3	2	4
Considero que el sistema Deeptools es muy difícil de usar	2	2	1	3
Me siento confiado al usar el sistema Deeptools	4	3	5	4
Necesité aprender muchas cosas antes de ser capaz de usar el sistema Deeptools	3	2	2	2
Ecuacion SUS	77,50	70,00	75,00	67,50



²⁶ Fabio DEVIN. *Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? | UXpañol*. (Accessed on 04/15/2021).

Los resultados de la escala de aceptabilidad nos muestra que los usuarios 1, 2 y 3 se encuentra en una escala aceptable, el usuario 4 en marginal alto. La calificación del usuario 4 denota una cierta inconformidad en comparación al resto de usuarios. Una de las razones de esto podría el rol del usuario además de las responsabilidades que cumplió cada tipo de rol durante el desarrollo del prototipo. *Deeptools* fue creado de la mano del grupo de investigación *BIVL²ab*, este con integrantes mayormente creadores de algoritmos científicos, cumpliendo con el papel de ‘Desarrollador’, generando un mayor acople a las necesidades de este usuario, por otro lado el contacto con el otro rol de la plataforma fue menor, limitando *feedback* de este.

Cabe mencionar que la plataforma *Deeptools* tiene un nivel de maduración **TLR-L4** (Validación de componente y/o disposición de los mismos en entorno de laboratorio), como prototipo de software se planteo un mínimo producto viable. Este sistema esta en proceso de registro de software, atendiendo los lineamientos institucionales requeridos desde la Vicerrectoría Académica. En general podemos decir que el prototipo se encuentra en una escala aceptable en la que los usuarios percibieron efectividad, eficiencia y satisfacción.²⁷

²⁷ Jorge Montiel A. *De la Escala de Usabilidad de Sistemas | by Jorge Montiel Anguiano | Medium.* (Accessed on 04/15/2021).

6. Conclusiones y perspectivas

En este trabajo se presentó el diseño y desarrollo de la plataforma de software *Deeptools* que permite el despliegue, ejecución y visualización de algoritmos dedicados a la visión por computador, el aprendizaje de máquina y las representaciones de aprendizaje profundo. En este trabajo, la plataforma *Deeptools* fue exitosamente validada en diferentes propuestas y aplicaciones desarrolladas dentro del grupo de investigación *BIVL²ab*, que en principio muestran una aproximación científica a diversos problemas en ingeniería biomédica, pero que su despliegue en la plataforma permitió acercarse más a la cadena productiva de dichas soluciones, en las áreas específicas de aplicación.

Durante la ejecución del proyecto se aportó en un diseño novedoso basado en estrategias de suscripción de usuarios a un portafolio de algoritmos y soluciones planteadas. Desde allí, no solo se tiene un esquema efectivo de usabilidad sino que además se propende en la administración efectiva de los recursos y el registro de las múltiples ejecuciones de los algoritmos. Para este diseño se inició con planteamientos básicos y requerimientos brindados por los integrantes del equipo *BIVL²ab*, luego junto con las reglas de negocio se hicieron varias interacciones para lograr un acuerdo con respecto a las capacidades de las herramientas tecnológicas. Así, se lograron flexibilizar y adaptar algunos requerimientos a la infraestructura planteada, también buscar soluciones tecnológicas mas acordes a las vivencias en ciencias de la computación y aprendizaje de máquina, logrando construir procesos mas adecuados para el despliegue y funcionalidad de la herramienta.

En la solución entregada cabe destacar la implementación de una arquitectura de micro-servicios, permitiendo organizar los diferentes procesos y componentes de la solución de manera independiente. Siendo así mas idóneo su escalamiento y verificación de funcionalidad. También, este tipo de arquitectura debe estar acompañado de los contenedores *Docker* y tecnologías de orquestación como *Kubernetes* para mejorar la escalabilidad y accesibilidad del software, además evita solucionar manualmente los problemas de concurrencia.

La plataforma desarrollada fue evaluada y valorada por un grupo de estudiantes enfocados al diseño de soluciones en ciencias de la computación, así como profesionales dedicados a la investigación. De hecho se hizo una prueba estándar de usabilidad, logrando un puntaje de 72.5/100, lo cual indica que los usuarios estuvieron navegando con la herramienta de manera efectiva y su visión ante la herramienta es de satisfacción aceptable. También el *SPA (Single Page Application)* desarrollado nos brinda una muy buena experiencia visual. La plataforma alcanzó un nivel de maduración **TLR-L4**, el cual fue notificado al comité de propiedad intelectual (CPI) de la Vicerrectoría Académica, para seguir los mecanismos de registro de la solución.

La arquitectura propuesta se realizó bajo una *SPA* funcional con la librería *React JS*. Sin embargo, nos quita algunas características necesarias para un software tal como el *SEO (Search Engine Optimization)*, evitando un buen posicionamiento de la plataforma en los motores de búsqueda, se recomienda a otra tecnología para ofrecer soporte *SEO*. Teniendo en cuenta los requerimientos en los ambientes académicos y científicos. También se podría considerar hacer ambientes web con tecnologías como *Gatsby* o *Next JS* compatibles con la librería *React JS* para reutilizar gran parte del código.

En futuras versiones del aplicativo se recomienda se recomienda agregar la posibilidad de cancelación de un proceso, tanto en el despliegue de un algoritmo, como en su ejecución. Así como también se recomienda crear compatibilidad con imágenes docker publicadas en *Docker Hub*, esto evitaría la construcción de la imagen docker en el servidor, mejorando la experiencia a los usuarios desarrolladores. El uso de tecnologías como *Kubernetes* podrá mejorar la escalabilidad, accesibilidad y concurrencia del software. Del mismo modo podrá trabajar en versiones que tengan compatibilidad con servicios de computación en la nube para mejorar problemas de concurrencia, seguridad y escalabilidad.

Bibliografía

Algorithmia. *What is Algorithmia?* (accessed: 26.02.2021) (vid. pág. 27).

Anderson, C. “Docker [Software engineering]”. En: *IEEE Software* 32.3 (2015), págs. 102-c3. DOI: 10.1109/MS.2015.62 (vid. pág. 23).

Azure, Microsoft. *Cloud Computing Services: Microsoft Azure*. (accessed: 12.03.2021) (vid. pág. 26).

Carlos Abel Capeáns Hurtado, Rafael Rodríguez Puente. “Compatibility technology in of Enterprise Systems implementation”. En: (2015) (vid. pág. 16).

CHAUHAN, Vinod Kumar. “Smoke Testing”. En: *Int. J. Sci. Res. Publ* 4.1 (2014), págs. 2250-3153 (vid. pág. 43).

Cortex. Cloud infrastructure for machine learning at scale. 2022 (vid. pág. 26).

DEVIN, Fabio. *Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? / / UXpañol*. (Accessed on 04/15/2021) (vid. pág. 50).

Docker, Inc. *What is a Container?* URL: <https://www.docker.com/resources/what-container>. (accessed: 26.02.2020) (vid. pág. 23).

Fan, Pei y col. “A topology-aware method for scientific application deployment on cloud”. En: *International Journal of Web and Grid Services* 10.4 (2014), pág. 338. DOI: 10.1504/ijwgs.2014.064937 (vid. pág. 16).

Flask. Web development, one drop at a time. 2022 (vid. pág. 26).

Foundation, Cloud Native Computing. *What is a gRPC?* URL: <https://grpc.io>. (accessed: 15.03.2021) (vid. pág. 25).

Fowler, Martin y James Lewis. “Microservices a definition of this new architectural term”. En: URL: <http://martinfowler.com/articles/microservices.html> (2014), pág. 22 (vid. pág. 20).

Granchelli, G. y col. “Towards Recovering the Software Architecture of Microservice-Based Systems”. En: *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. 2017, págs. 46-53. DOI: 10.1109/ICSAW.2017.48 (vid. pág. 21).

Gropp, William y col. *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. MIT press, 1999 (vid. pág. 15).

Maquinaria Informática, Asociación de, ed. *Implementación de llamadas a procedimientos remotos*. DOI: 10.1145/800217.806609 (vid. pág. 24).

Merali, Zeeya. “Error: why scientific programming does not compute”. En: *Nature* 467.7317 (2010), págs. 775-777 (vid. pág. 18).

MLflow. 2022 (vid. pág. 25).

MLflow. An open source platform for the machine learning lifecycle. 2022 (vid. pág. 28).

Montiel A., Jorge. *De la Escala de Usabilidad de Sistemas / by Jorge Montiel Anguiano / Medium*. (Accessed on 04/15/2021) (vid. pág. 51).

Prabhu, P. y col. “A survey of the practice of computational science”. En: *SC '11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. 2011, págs. 1-12. DOI: 10.1145/2063348.2063374 (vid. pág. 18).

Rahwan, Iyad. *Argumentation in Artificial Intelligence*. Springer, 2009, <https://link.springer.com/content/p>
(vid. pág. 15).

Richardson, Chris. “Pattern: monolithic architecture”. En: *Posjećeno* 15 (2014), pág. 2016 (vid. pág. 20).

SAWANT, Abhijit A, Pranit H BARI y PM CHAWAN. “Software testing techniques and strategies”. En: *International Journal of Engineering Research and Applications (IJERA)* 2.3 (2012), págs. 980-986 (vid. pág. 48).

Smith, James E y Ravi Nair. “The architecture of virtual machines”. En: *Computer* 38.5 (2005), págs. 32-38 (vid. pág. 22).

Tensorflow. 2022 (vid. pág. 25).

Trycore. *Norma ISO 29119*. (accessed: 5.01.2022) (vid. pág. 43).

Turnbull, James. *The Docker Book: Containerization is the new virtualization*. James Turnbull, 2014 (vid. pág. 23).

Anexos

Anexo A. Plan de pruebas

A continuación, se presentan el informe del plan de pruebas.



Plan de pruebas

Juan Sebastian Marcon Caballero
Universidad Industrial de Santander
Biomedical Imaging, Vision and Learning Laboratory (BIVL2ab)

1 Introducción	2
1.1 Alcance	2
2 Contexto de las pruebas	2
2.1 Proyecto	2
2.2 Elementos de pruebas	2
2.3 Alcance de las pruebas	2
2.4 Restricciones	2
2.3 Partes interesadas	2
3 Registro de riesgos	3
4 Estrategia de Prueba	3
4.1 Subprocesos de prueba	3
4.2 Técnicas de diseño de prueba	3
4.3 Criterios de finalización	3
4.4 Criterios de suspensión y reanudación	3
4.4.1 Criterios de finalización	3
4.4.2 Criterios de reanudación	4
5 Estrategia de Prueba	4
5.1 Actividades de prueba	4

1 Introducción

1.1 Alcance

El propósito de este documento es proporcionar la información y el marco requerido para la planificación y desarrollo de las actividades del proceso de pruebas del prototipo de software.

2 Contexto de las pruebas

2.1 Proyecto

El prototipo de software cuenta con tres grandes módulos:

- Módulo de usuarios
- Módulo de algoritmos
- Módulo de ejecuciones

2.2 Elementos de pruebas

Se realizarán las pruebas a los siguientes elementos:

- Módulo de usuarios
- Módulo de algoritmos
- Módulo de ejecuciones

2.3 Alcance de las pruebas

De acuerdo a los requerimientos establecidos para el prototipo de software se decidió que no se tendrán en cuenta los factores de calidad no funcionales no establecidos como requerimiento, excluyendo pruebas de rendimiento, seguridad, usabilidad. Solo se tendrá en cuenta las pruebas de requerimientos funcionales y no funcionales.

2.4 Restricciones

Todo el plan de pruebas será desarrollado desde el entorno de desarrollo, no se contará con un entorno dedicado.

2.3 Partes interesadas

Director y Coodirector	Aprobación del plan de pruebas, Cronograma de pruebas y entregables. Realización de pruebas de aceptación.
Grupo de investigación Bivi2ab	Cronograma de pruebas y entregables. Realización de pruebas de aceptación.

3 Registro de riesgos

En la siguiente tabla se muestran los riesgos del prototipo así como se determina la severidad de cada uno de los riesgos multiplicando el impacto por la probabilidad de ocurrencia,

El impacto y la probabilidad se determinan teniendo en cuenta una escala de 1 a 5, donde 5 es el más alto y 1 lo más bajo.

#	Riesgo	Probabilidad [1-5]	Impacto [1-5]	Severidad	Mitigación
1	Retrasos en la implementación de funcionalidades establecidas en la arquitectura	3	5	15	Evaluar la experiencia de los miembros del equipo y re-planificar el cronograma establecido

4 Estrategia de Prueba

4.1 Subprocesos de prueba

Las pruebas para los módulos establecidos incluyen los siguientes subprocesos de prueba:

- Pruebas de integración
- Pruebas de sistema
- Pruebas de aceptación

4.2 Técnicas de diseño de prueba

Según los requerimientos establecidos se determinó que las técnicas que se utilizaran para el diseño de las pruebas:

- Pruebas de humo
- Análisis de valores límites
- Pruebas exploratorias
- Pruebas de negación

4.3 Criterios de finalización

Las pruebas deben alcanzar una cobertura de al menos el 80% de los requisitos establecidos en la fase de análisis, y ninguna de las fallas podrá ser de gravedad alta.

4.4 Criterios de suspensión y reanudación

4.4.1 Criterios de finalización

- Que el prototipo no cumpla con los requerimientos especificados en la fase de análisis
- Un error en una parte crítica del sistema

4.4.2 Criterios de reanudación

Llegar a un acuerdo entre las partes interesadas para reanudar las pruebas o que los defectos hayan sido solucionados

5 Estrategia de Prueba

5.1 Actividades de prueba

Las pruebas se dividirán en las siguientes actividades:

1. Definición de una estructura general de las pruebas por conjuntos de casos de uso a probar
2. Especificación detallada de los casos de prueba
3. Ciclo de ejecución de pruebas
4. Informe de suspensión de pruebas si aplica

Anexo B. Casos de prueba

A continuación, se presentan los casos de pruebas creados según la planeación de pruebas.

Identificador	Description	Precondiciones	Pasos	Resultados esperados	Resultados obtenidos	Pos condiciones	Bugs	Estado
CU00								
CP-CU00-1	Iniciar sesión	- Tener cuenta en el prototipo	1. Llene los campos vacíos 2. De click sobre el botón 'Sign In now'	Se redirecciona a el tablero del prototipo	Se redirecciono a el tablero del prototipo	- Se inicia sesión		Completada
CP-CU00-2	Iniciar sesión agregando con datos no reales		1. Llene los campos vacios con valores no reales 2. De click sobre el botón 'Sign In now'	Aparece una caja roja con el mensaje de error	Apareció una caja roja con el mensaje de error	- No se inicia sesión		Completada
CU01								
CP-CU01-1	Cerrar sesión	- Tener una sesión activa	1. De click sobre su foto de perfil en la parte superior derecha, esto desplegará un menú con 3 opciones 2. De click en la opción 'Logout'	Se redirecciona a el inicio de sesión del prototipo	Se redirecciono a el inicio de sesión del prototipo	- Se elimina la sesión activa		Completada
CU02								
CP-CU02-1	Registrar usuario		1. Llene los campos del formulario 2. De click sobre el botón 'Create account'	Se redirigirá a el tablero del prototipo	Se redirecciono a el tablero del prototipo	- Se crea una cuenta en el prototipo - Se inicia sesión		Completada
CP-CU02-2	Registrar usuario con un correo que ya existente		1. Llene los campos del formulario 2. Llene el campo 'Email address' con un correo ya existente 3. De click sobre el botón 'Create account'	Se muestra un error debajo del campo, informando lo sucedido	Se muestro un error debajo del campo, informando lo sucedido	- No se crea una cuenta en el prototipo - No se inicia sesión	B01	Completada
CU03								
CP-CU03-1	Ver perfil de usuario	- Tener una sesión activa	1. De click sobre su foto de perfil en la parte superior derecha, esto desplegará un menú con 3 opciones 2. De click en la opción 'Profile'	Se redirigirá a una pantalla con información de su cuenta	Se redirecciono a una pantalla con información de su cuenta			Completada
CU04								
CP-CU04-1	Editar perfil de usuario	- Tener una sesión activa	1. De click sobre su foto de perfil en la parte superior derecha, esto desplegará un menú con 3 opciones 2. De click en la opción 'Settings' 3. Modifique los campos que desee 4. De click el botón 'Upload profile'	Se redirigirá a una pantalla donde podrá editar la información de su cuenta	Se redirecciono a una pantalla donde podrá editar la información de su cuenta	- Se modifica la información de la cuenta del usuario		Completada
CP-CU04-2	Editar perfil de usuario con un correo que ya existente	- Tener una sesión activa	1. De click sobre su foto de perfil en la parte superior derecha, esto desplegará un menú con 3 opciones 2. De click en la opción 'Settings' 3. Modifique los campos que desee 4. Modifique el campo 'Email address' con un correo ya existente 5. De click el botón 'Upload profile'	Se muestra un error debajo del campo, informando lo sucedido	Se muestro un error debajo del campo, informando lo sucedido	- No se modifica la información de la cuenta del usuario	B02	Completada
CU05								
CP-CU05-1	Ver perfil de desarrolladores	- Tener una sesión activa	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. En la pestaña 'Description' se ve una línea de texto con el nombre del desarrollador, de click sobre este	Se redirigirá a una pantalla con la información del desarrollador	Se redirecciono a una pantalla con la información del desarrollador			Completada
CU06								
CP-CU06-1	Listar notificaciones	- Tener una sesión activa	1. De click sobre el icono 'Notifications' en la parte superior derecha, esto desplegará una sublista de todas las notificaciones activas 2. De click sobre la opción 'Show all'	Se redirigirá a una pantalla listando todas las notificaciones	Se redirecciono a una pantalla listando todas las notificaciones			Completada
CU07								
CP-CU07-1	Notificar novedades	- Tener una sesión activa - El estado de un algoritmo o ejecución cambio		El contador de el icono 'Notifications' aumenta, además si tienes las notificaciones activas deberá aparecer una notificación popup	El contador de el icono 'Notifications' aumento	- Se creó un registro de notificación	B03, B04	Completada
CU08								

CP-CU08-1	Suscribirse a un algoritmo	- Tener una sesión activa	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'Subscribe'	Desaparecerá el botón 'Subscribe' y aparecer 3 botones más, que te permitirán diferentes acciones para tu suscripción	desapareció el botón 'Suscribete' y aparecen 3 botones más.	- Se crea una suscripción a el algoritmo seleccionado		Completada
CU09								
CP-CU09-1	Cancelar la suscripción a un algoritmo	- Tener una sesión activa - Tener una suscripción	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de el algoritmo que desea desuscribirse, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'Unsubscribe'	Desaparecerán los 3 botones y aparecerá el botón 'Subscribe'	Desaparecieron los 3 botones y Apareció el botón 'Subscribe'	- Se eliminará la suscripción a el algoritmo seleccionado		Completada
CU10								
CP-CU10-1	Listar suscripciones	- Tener una sesión activa - Tener una suscripción	1. De click sobre el icono 'Subscriptions' en la parte superior derecha.	redirigirá a una pantalla con todas sus suscripciones	Se redirecciono a una pantalla con todas sus suscripciones			Completada
CU11								
CP-CU11-1	Desplegar un algoritmo	- Tener una sesión activa - Tener la imagen docker con las especificaciones - Ser un usuario de tipo desarrollador	1. De click sobre el icono 'New algorithm' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. Llene la información del protocolo de toma de datos y de click en el botón 'Next' 3. Seleccione los elementos que desea en su presentación y de click en el botón 'Next' 4. Responda las preguntas que harán acerca de su imagen docker 5. De click sobre el botón 'Send'	Se muestra una barra de carga que indicara que se están haciendo las comprobaciones necesarias para desplegar el contenedor, una vez finalizada será redirigido a una pantalla donde se muestran detalles de la construcción de su contenedor docker	Se mostraron una barra de carga que indica que se están haciendo las comprobaciones necesarias para desplegar el contenedor, una vez finalizada se redirecciono a una pantalla donde se muestran detalles de la construcción de su contenedor docker	- Se crea un registro del algoritmo desplegado - El algoritmo cambio de estado - El prototipo avisa al desarrollador al finalizar el despliegue	B05	Completada
CP-CU11-2	Desplegar un algoritmo con metada falsa	- Tener una sesión activa - Tener la imagen docker con las especificaciones - Ser un usuario de tipo desarrollador	1. De click sobre el icono 'New algorithm' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. Llene la información del protocolo de toma de datos y de click en el botón 'Next' 3. Seleccione los elementos que desea en su presentación y de click en el botón 'Next' 4. Responda las preguntas de manera errónea que harán acerca de su imagen docker 5. De click sobre el botón 'Send'	Se mostrará una mensaje de error debajo de cada campo que no corresponda	Se mostró una mensaje de error debajo de cada campo que no corresponda	- No se crea un registro del algoritmo desplegado		Completada
CU12								
CP-CU12-1	Iniciar un algoritmo	- Tener una sesión activa - Ser un usuario de tipo desarrollador - Tener un algoritmo desplegado	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el icono 'start' del algoritmo que desea iniciar	El icono del algoritmo cambiará a 'active'	El icono del algoritmo cambio a 'active'	- cambiará el estado del algoritmo a 'active'		Completada
CU13								
CP-CU13-1	Detener un algoritmo	- Tener una sesión activa - Ser un usuario de tipo desarrollador - Tener un algoritmo desplegado - Tener un algoritmo iniciado	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el icono 'stop' del algoritmo que desea iniciar	El icono del algoritmo cambiará a 'stopped'	El icono del algoritmo cambio a 'stopped'	- cambiará el estado del algoritmo a 'stopped'	B06	Completada
CU14								
CP-CU14-1	Eliminar un algoritmo	- Tener una sesión activa - Ser un usuario de tipo desarrollador - Tener un algoritmo desplegado - Tener un algoritmo iniciado	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título 'delete' del algoritmo que desea iniciar 3. aparecerá una ventana para verificar el proceso, de click sobre el botón 'Confirm'	El algoritmo desaparecerá de la pantalla	El algoritmo desapareció de la pantalla	- cambiará el estado del algoritmo a 'deleted'	B07	Completada

CU15							
CP-CU15-1	Administrar los usuario suscritos de un algoritmo	- Tener una sesión activa - Ser un usuario de tipo desarrollador - Tener un algoritmo desplegado	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantalla con los algoritmos activos 2. De click sobre el título de el algoritmo que desea desuscribirse, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. Ve a la pestaña 'Users'	Se redirigirá a una pantalla donde podrás administrar los usuario suscritos a un algoritmo	Se redirecciono a una pantalla donde se puede administrar los usuario suscritos a un algoritmo		Completada
CU16							
CP-CU16-1	Ejecutar un algoritmo	- Tener una sesión activa - Tener una suscripción	1. De click sobre el icono 'Subscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con sus suscripciones 2. Da click sobre el icono 'test' del algoritmo que desea ejecutar, esto lo redirigirá a una pantalla con el protocolo de toma de datos 3. Da click sobre el botón 'Next', le mostrará una caja donde podrás agregar los archivos necesarios para ejecutar el algoritmo 4. Da click sobre el botón 'Execute'	Se redirigirá a una pantalla donde se muestran detalles de la ejecución del algoritmo	Se redirecciono a una pantalla donde se muestran detalles de la ejecución del algoritmo	- Se crearán registros del log del algoritmo - Se crearán registros de los resultados del algoritmo - El prototipo avisará al usuario cuando haya finalizado la ejecución del algoritmo	Completada
CU17							
CP-CU17-1	Listar todas las ejecuciones de un algoritmo	- Tener una sesión activa - Tener una suscripción	1. De click sobre el icono 'Subscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'All'	Se redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo	Se redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo		Completada
CU18							
CP-CU18-1	Ver una ejecución de un algoritmo	- Tener una sesión activa - Tener una suscripción - Tener una ejecución finalizada	1. De click sobre el icono 'Subscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'All', esto lo redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo 4. De click sobre el icono 'Show test' de su ejecución de interés	Se redirigirá a una pantalla donde se muestran detalles de la ejecución del algoritmo	Se redirecciono a una pantalla donde se muestran detalles de la ejecución del algoritmo		B08 Completada
CU19							
CP-CU19-1	Eliminar una ejecución de un algoritmo	- Tener una sesión activa - Tener una suscripción - Tener una ejecución finalizada	1. De click sobre el icono 'Subscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'All', esto lo redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo 4. De click sobre el icono 'Delete test' de su ejecución de interés 5. aparecerá una ventana para verificar el proceso, de click sobre el botón 'Confirm'	La ejecución desaparecerá de la pantalla	La ejecución desapareció de la pantalla	- Se elimina la ejecución asociada al algoritmo	Completada
CU20							
CP-CU20-1	Clonar una ejecución de un algoritmo	- Tener una sesión activa - Tener una suscripción - Tener una ejecución finalizada	1. De click sobre el icono 'Subscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'All', esto lo redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo 4. De click sobre el icono 'Clone test with same data' de su ejecución de interés	CU09	Se redirecciono a la vista de ejecución de algoritmo con los mismos datos de entrada		Completada

CU21							
CP-CU21-1	Filtrar algoritmos	- Tener una sesión activa	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. Escriba dentro del campo superior.	Se filtran los algoritmos según lo escrito en el campo	Se filtraron los algoritmos según lo escrito en el campo		Completada
CU22							
CP-CU22-1	Listar algoritmos	- Tener una sesión activa	1. De click sobre el icono 'Algorithms' en la parte superior derecha	Se redirigirá a una pantallas con los algoritmos activos	Se redirecciono a una pantallas con los algoritmos activos		Completada
CU23							
CP-CU23-1	Descargar elementos multimedia	- Tener una sesión activa - Tener una suscripción - Tener una ejecución finalizada	1. De click sobre el icono 'Suscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'All', esto lo redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo 4. De click sobre el icono 'Show test' de su ejecución de interés, esto lo redireccionará a una pantalla con los detalles de la ejecución. 5. De click derecho sobre el elemento que desea descargar, esto desplegará un menu de opciones. 6. Seleccione la opción descargar.	Se descargara el elemento seleccionado	Se descargo el elemento seleccionado		Completada

Anexo C. Incidencias


A continuación, se muestran las incidencias o bugs registrados durante la ejecución de los casos de pruebas.

ID	Nombre	Description	Precondiciones	Resultados esperados	Resultados obtenidos	Pasos	Estado	Prioridad
B01	No se muestra feedback con usuario ya registrados	Al registrar un usuario ya existente no se muestra error, la pantalla no cambia en absoluto		Se muestra un error debajo del campo, informando lo sucedido	No se muestra mensaje de error	1. Llène los campos del formulario 2. Llène el campo 'Email address' con un correo ya existente 3. De click sobre el botón 'Create account'	Solucionado	Alta
B02	No se muestra feedback cuando intento cambiar el correo con uno ya existente	Al editar el usuario con un correo ya existente no se muestra error, la pantalla no cambia en absoluto	- Tener una sesión activa	Se muestra un error debajo del campo, informando lo sucedido	No se muestra mensaje de error	1. De click sobre su foto de perfil en la parte superior derecha, esto desplegará un menú con 3 opciones 2. De click en la opción 'Settings' 3. Modifique los campos que desee 4. Modifique el campo 'Email address' con un correo ya existente 5. De click el botón 'Upload profile'	Solucionado	Alta
B03	No se notifica al usuario	No se notifica al usuario al momento de que una ejecución o algoritmo cambia de estado	- Tener una sesión activa - El estado de un algoritmo o ejecución cambio	El contador de el icono 'Notifications'	El contador del icono no cambio		Solucionado	Media
B04	No aparece la notificación popup	Como usuario active las notificaciones del protipo, pero no se muestra la notificación popup	- Tener una sesión activa - El estado de un algoritmo o ejecución cambio	Aparece una notificación popup	No aparecio la notificación popup		Solucionado	Media
B05	No se muestra una barra de progreso	Al momento de desplegar un algoritmo no se muestra la barra de progreso en tiempo real	- Tener una sesión activa - Tener la imagen docker con las especificaciones - Ser un usuario de tipo desarrollador	Se muestra una barra de carga que indicara que se están haciendo las comprobaciones necesarias para desplegar el contenedor, una vez finalizada será redirigido a una pantalla donde se muestran detalles de la construcción de su contenedor docker	No se mostro una barra de progreso	1. De click sobre el icono 'New algorithm' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. Llène la información del protocolo de toma de datos y de click en el botón 'Next' 3. Seleccione los elementos que desea en su presentación y de click en el botón 'Next' 4. Responda las preguntas que harán acerca de su imagen docker 5. De click sobre el botón 'Send'	Solucionado	Alta
B06	No se detiene el algoritmo	Al intentar detener un algoritmo en uso, no cambio el icono a pausado	- Tener una sesión activa - Ser un usuario de tipo desarrollador - Tener un algoritmo desplegado - Tener un algoritmo iniciado	El icono del algoritmo cambiará a 'stopped'	No se muestra cambio grafico	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el icono 'stop' del algoritmo que desea iniciar	Solucionado	Media
B07	No se puede eliminar un algoritmo	Al intentar eliminar un algoritmo en uso, este no desaparece de la pantalla	- Tener una sesión activa - Ser un usuario de tipo desarrollador - Tener un algoritmo desplegado - Tener un algoritmo iniciado	El algoritmo desaparecerá de la pantalla	El algoritmo no desaparecerá de la pantalla	1. De click sobre el icono 'Algorithms' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título 'delete' del algoritmo que desea iniciar 3. aparecerá una ventana para verificar el proceso, de click sobre el botón 'Confirm'	Solucionado	Media
B08	No se ven los detalles de la ejecución del algoritmo	Al ver los detalles de la ejecución de un algoritmo, no se muestra información alguna, se ve una pantalla en blanco	- Tener una sesión activa - Tener una suscripción - Tener una ejecución finalizada	Se redirigirá a una pantalla donde se muestran detalles de la ejecución del algoritmo	Se ve la pantalla vacia	1. De click sobre el icono 'Suscriptions' en la parte superior derecha, esto lo redireccionará a una pantallas con los algoritmos activos 2. De click sobre el título de uno de los algoritmos de su interés, esto lo redireccionará a una pantalla con los detalles del algoritmo 3. De click sobre el botón 'All', esto lo redirigirá a una pantalla donde se muestra una lista de las ejecuciones del algoritmo 4. De click sobre el icono 'Show test' de su ejecución de interés	Solucionado	Alta

Anexo D. Cuestionario SUS

A continuación, se muestra el formulario implementado para el análisis de usabilidad.

Cuestionario de satisfaccion

 [juanmarcon1080@gmail.com](#) (no compartidos) [Cambiar de cuenta](#)



*Obligatorio

Creo que usaría el sistema Deeptools frecuentemente *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Encuentro el sistema Deeptools innecesariamente complejo *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo



Creo que el sistema Deeptools fue fácil de usar *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar el sistema Deeptools *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Las funciones del sistema Deeptools están bien integradas *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo



Creo que el sistema Deeptools es muy inconsistente *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Pienso que la mayoría de la gente aprendería a usar el sistema Deeptools rápidamente *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Considero que el sistema Deeptools es muy difícil de usar *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo



Me siento confiado al usar el sistema Deeptools *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Necesité aprender muchas cosas antes de ser capaz de usar el sistema Deeptools *

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios



Anexo E. Manuales del usuario

A continuación, se presenta el manual de usuario para el role 'usuario'.



Manual del usuario

Juan Sebastian Marcon Caballero
Universidad Industrial de Santander
Biomedical Imaging, Vision and Learning Laboratory (BIVL2ab)

Contenido

1. Conceptos	3
1.1 Algoritmos	3
1.1.1 Descripción	3
1.1.1.1 Desarrollador	3
1.1.1.2 Extensiones	3
1.1.2 Protocolo	3
1.1.3 Ejemplo	3
1.1.4 Algoritmos activos	3
1.1.5 Ejecución	4
1.1.5 Pruebas	4
1.2 Suscripciones	4
1.3 Corriendo	4
2. ¿Cómo ingreso a la plataforma?	4
3. Partes de la plataforma	6
3.1 Barra de navegación	6
3.2 Miga de pan	6
3.3 ¿Cómo hago una búsqueda rápida?	6
4. ¿Cómo administrar mi perfil de usuario?	7
4.1 ¿Cómo cierro mi sesión?	7
4.2 ¿Cómo ver mi perfil de usuario?	8
4.3 ¿Cómo modificar mi perfil de usuario?	8
5 Módulo algoritmos	10
5.1 Mis suscripciones	11
5.2 Algoritmos activos	11
5.3 ¿Cómo busco algoritmos y desarrolladores de mi interés?	11
5.3 ¿Cómo veo más detalles de un algoritmo específico?	12
5.3.1 ¿Qué información puedo ver de un algoritmo?	13
5.3.2 ¿Cómo me suscribo a un algoritmo?	13
5.3.3 ¿Cómo elimino la suscripción a un algoritmo?	13
5.4 ¿Cómo veo más detalles de un desarrollador específico?	13
6. Módulo suscripciones	14
6.1 Estados	15
6.1.1 Activo	15
6.1.2 Mantenimiento	15
6.2 ¿Cómo ver más detalles de un algoritmo específico?	15

6.3 ¿Cómo ver más detalles de un desarrollador específico?	15
6.4 ¿Cómo ver todas las pruebas que he realizado a un algoritmo suscrito?	15
6.4.1 ¿Cómo administrar las pruebas que he realizado?	16
6.4.1.1 ¿Cómo crear una prueba?	16
6.4.1.1.1 ¿Cómo crear una prueba con datos de prueba?	19
6.4.1.2 ¿Cómo ver una prueba?	23
6.4.1.3 ¿Cómo clonar una prueba?	23
6.4.1.4 ¿Cómo eliminar una prueba?	24
7. ¿Cómo puedo ver los algoritmos que están en ejecución?	25
8. ¿Cómo accedo a las notificaciones en tiempo real?	25

1. Conceptos

1.1 Algoritmos

Los algoritmos son un conjunto de instrucciones o normas definidas, estructuradas y finitas que permite solucionar una problemática, en esta plataforma se le llama así a los diferentes módulos que tienen una tarea específica.

1.1.1 Descripción

Descripción breve de la funcionalidad de un [algoritmo](#).

1.1.1.1 Desarrollador

El desarrollador es el creador de un [algoritmo](#) específico, éste es el encargado de implementar el debido mantenimiento.

1.1.1.2 Extensiones

En esta sección se muestra una lista de las extensiones permitidas para la ejecución de un [algoritmo](#).

1.1.2 Protocolo

Durante el diseño de los algoritmos los desarrolladores deben hacer muchas pruebas para garantizar que estos funcionen como se espera, durante esta etapa se encuentran diferentes anomalías de la solución que deben tenerse en cuenta. El protocolo es una descripción precisa para que la [ejecución](#) del [algoritmo](#) se lleve con éxito. **¡Debe ser seguida al pie de la letra!**

1.1.3 Ejemplo

Los ejemplos son cada una de las ejecuciones de un [algoritmo](#) realizadas por el [desarrollador](#), en esta parte podrás ver los detalles de cada una de estas ejecuciones sin la necesidad de estar [suscrito](#).

1.1.4 Algoritmos activos

Los algoritmos activos son los algoritmos que están listos para ser usados, algoritmos que no tienen ningún tipo de error o complicación durante su [ejecución](#).

1.1.5 Ejecución

La palabra ejecución se refiere a la acción de usar un algoritmo por parte de un usuario o por el [desarrollador](#). En otras palabras, si un algoritmo está en ejecución significa que ese algoritmo se está usando en ese momento.

1.1.5 Pruebas

Test o pruebas en español, corresponde a las diferentes ejecuciones de su autoría de un [algoritmo](#), en otras palabras son todos los usos que un usuario ha realizado en un [algoritmo](#).

1.2 Suscripciones

La suscripción es la relación que existe entre un [algoritmo](#) y un usuario interesado en usarlo. Esta funcionalidad está creada para llevar un registro de los usuarios interesados en un algoritmo en específico. Dado esto, si deseas usar algún algoritmo debes suscribirte a este.

Las suscripciones son representadas con el siguiente icono:



1.3 Corriendo

Corriendo es la sección donde se muestran los algoritmos que se encuentran en estado de [ejecución](#).

2. ¿Cómo ingreso a la plataforma?

Para acceder a la plataforma Deeptools se debe ingresar la siguiente dirección en el navegador [Google Chrome v86+](#):

Aparecerá una pantalla como la siguiente:



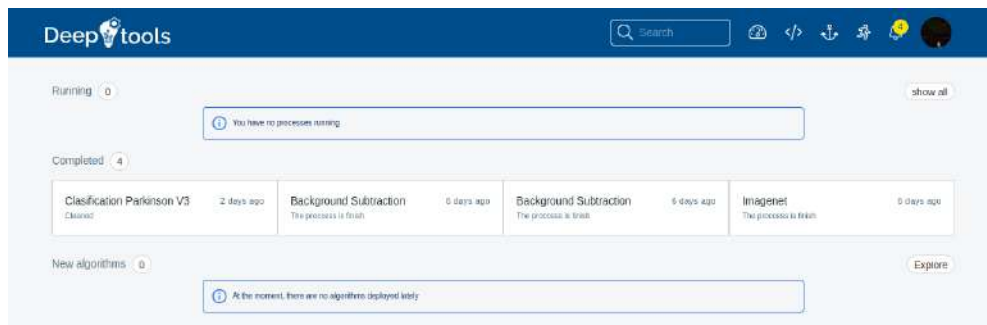
Sign In

Email address

Password

[Create account](#)

Después de ingresar su email registrado y su contraseña, presione el botón “Sign in now”. Debe aparecer una pantalla similar a la siguiente:



Nota: Para las siguientes funcionalidades, se asume que usted ya se encuentra autenticado en la plataforma Deeptools.

3. Partes de la plataforma

En esta plataforma le damos definiciones a diferentes partes representativas, con el fin de que los usuarios que se basen en esta guía entiendan cómo ubicarse durante toda la extensión del documento.

3.1 Barra de navegación

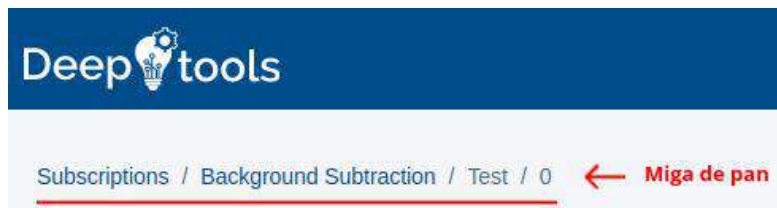
La barra de navegación está ubicada en la parte superior, cabe resaltar que esta es de color azul.



Esta barra cuenta con los botones necesarios para navegar.

3.2 Miga de pan

La [miga de pan](#) es el seguimiento de la navegación dentro de la plataforma. Esta ubicado un poco más abajo de la [barra de navegación](#).

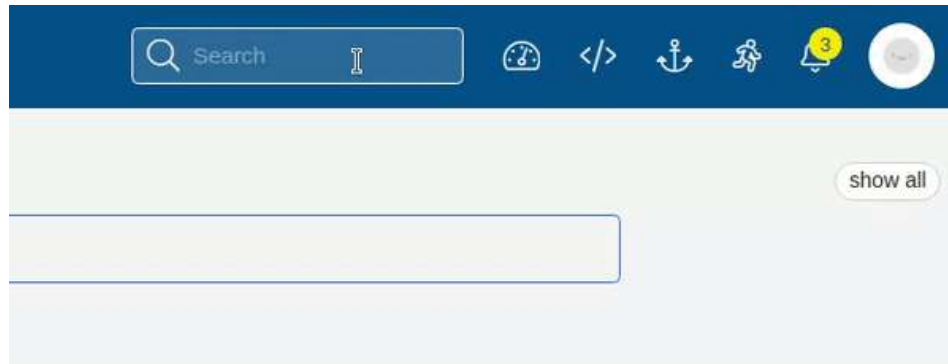


3.3 ¿Cómo hago una búsqueda rápida?

Ubicate en la parte derecha de la [barra de navegación](#) se ve la siguiente caja de texto:



En esta caja puedes buscar al igual que en la [sección de algoritmos](#), y desplegará un menú con las opciones que coincidan con la búsqueda.

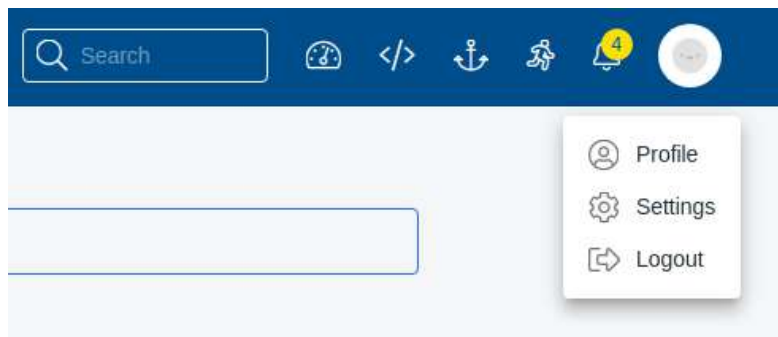


4. ¿Cómo administrar mi perfil de usuario?

Si requiere administrar su perfil de usuario, ubique el puntero en la [barra de navegación](#) hacia la derecha en el botón donde se ve su foto de perfil o en su defecto la siguiente imagen:



De click en él. Al hacerlo, se desplegará un menú con diferentes opciones.

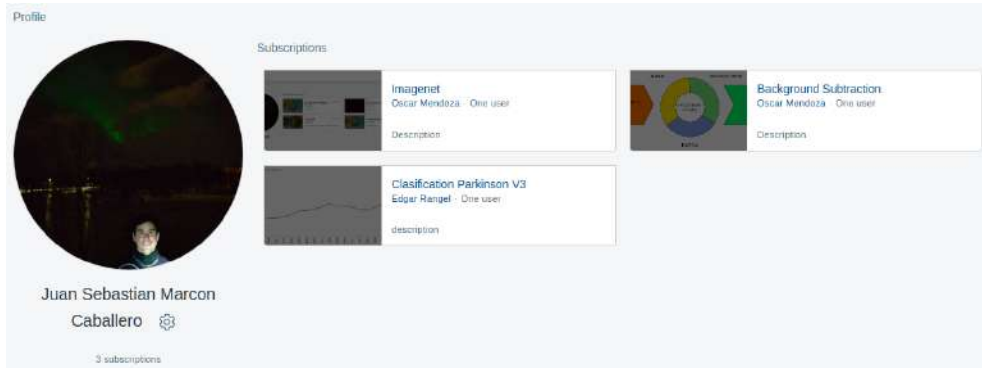


4.1 ¿Cómo cierro mi sesión?

Partiendo del [paso anterior](#). Seleccione la opción "Logout". Esto lo redireccionará a la vista ingreso.

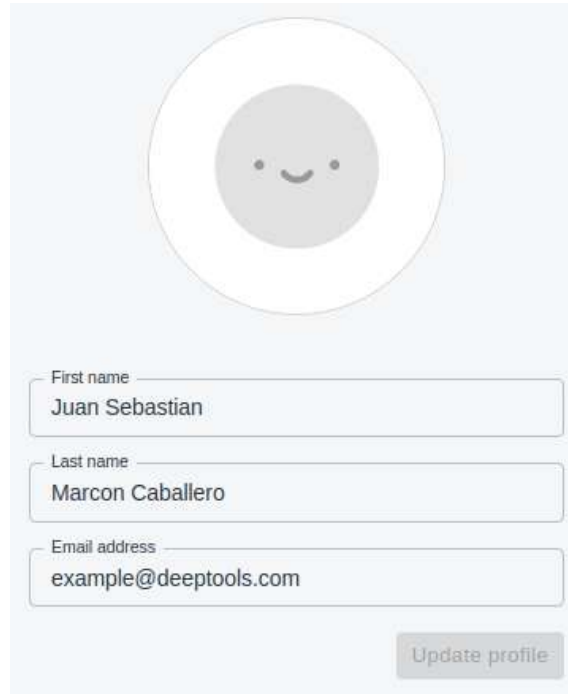
4.2 ¿Cómo ver mi perfil de usuario?

Partiendo del [ítem 4](#). Seleccione la opción “Profile”. Se mostrará una vista donde podrá visualizar los datos ingresados y sus respectivas suscripciones.



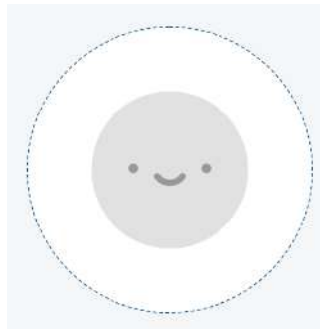
4.3 ¿Cómo modificar mi perfil de usuario?

Teniendo en cuenta el [ítem 2](#). Seleccione la opción “settings”. Se mostrará una ventana similar a la siguiente pantalla, en la cual se deben modificar los datos que desee.

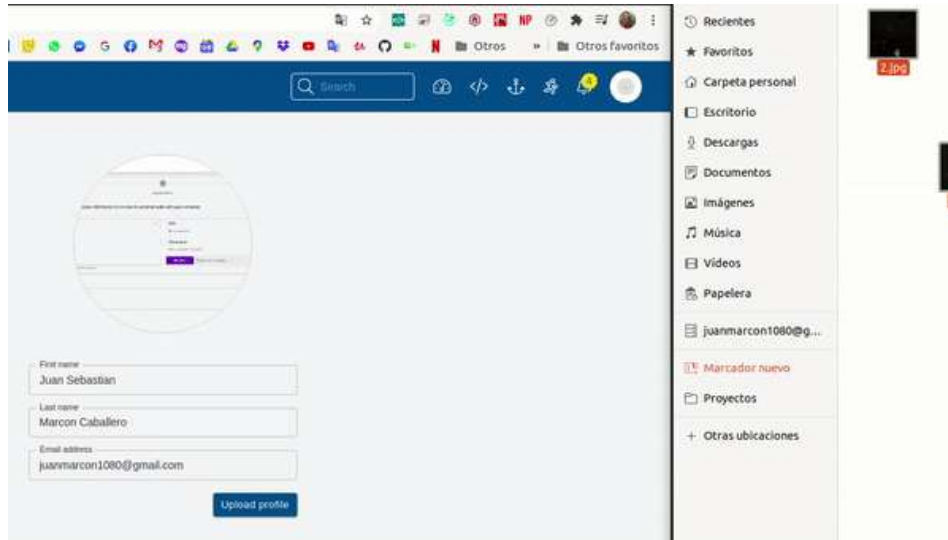


A profile update form with a light blue background. At the top is a circular profile picture placeholder containing a simple grey smiley face. Below the picture are three text input fields: 'First name' with the value 'Juan Sebastian', 'Last name' with the value 'Marcon Caballero', and 'Email address' with the value 'example@deeptools.com'. A grey 'Update profile' button is located at the bottom right of the form.

Si desea cambiar la foto de perfil, deberá poner el puntero encima de la foto actual, esto desplegará un cambio de estilo y color en los bordes indicando que este botón es una entrada de imagen.



Existen dos formas de seleccionar la imagen, la primera es haciendo click sobre la imagen, esto desplegará una administrador de archivos para que seleccione la imagen. La otra opción se basa en el arrastre. Deeptools provee la opción de arrastrar la imagen en cualquiera elemento de selección de archivos.



Una vez realizadas las modificaciones deberá dar click en el botón “Update profile” para guardar los cambios realizados. Esta acción lo llevará al [perfil de usuario](#).

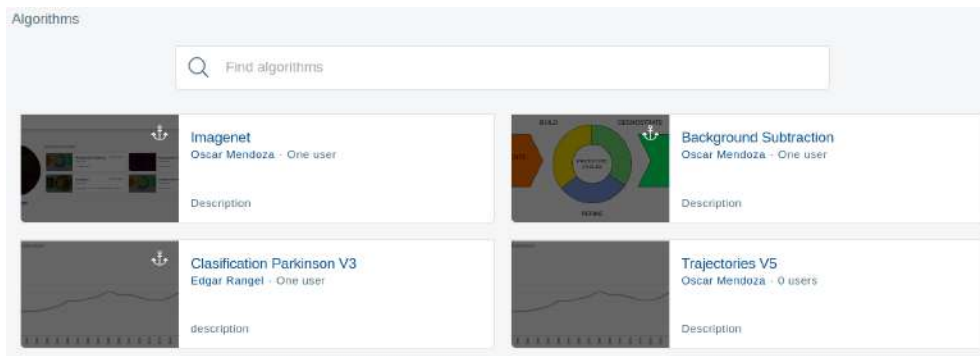
5 Módulo algoritmos

El módulo de algoritmos es una compilación de todos los [algoritmos activos](#) y tus [suscripciones](#). Además esta vista provee una caja de texto para poder filtrar entre estos algoritmos.

Para acceder a este debes ubicarte en la parte derecha de la [barra de navegación](#) y dar click en el siguiente icono:



Esto lo redirigirá hacia una vista como la siguiente:



5.1 Mis suscripciones

Mis [suscripciones](#) estarán diferenciadas con el siguiente icono:



5.2 Algoritmos activos

El indicador de suscripción es excluyente, luego todos aquellos algoritmos que **no** tengan el siguiente icono serán [algoritmos activos](#):

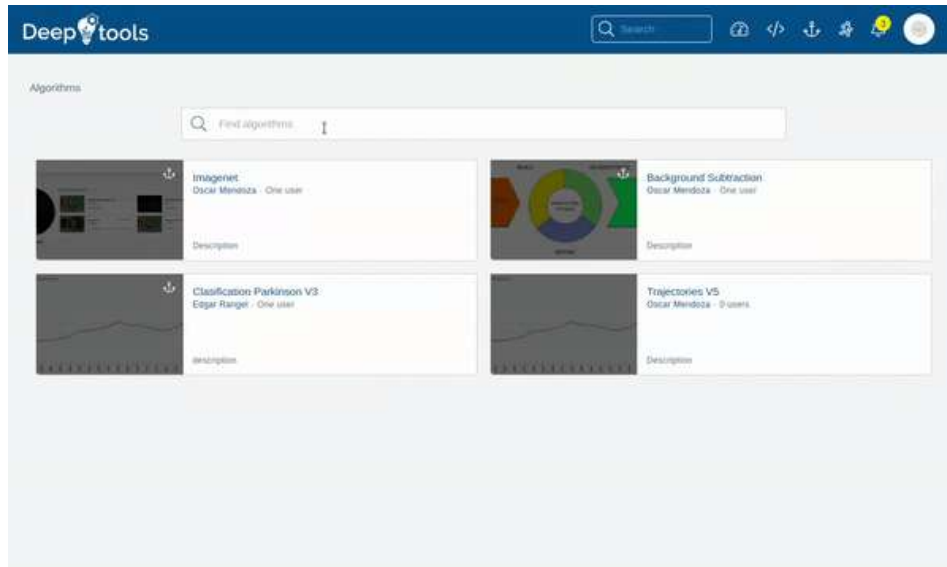


5.3 ¿Cómo busco algoritmos y desarrolladores de mi interés?

Para realizar búsquedas debes ubicarte en la caja de texto debajo de la [miga de pan](#), tal como se ve en la siguiente imagen.

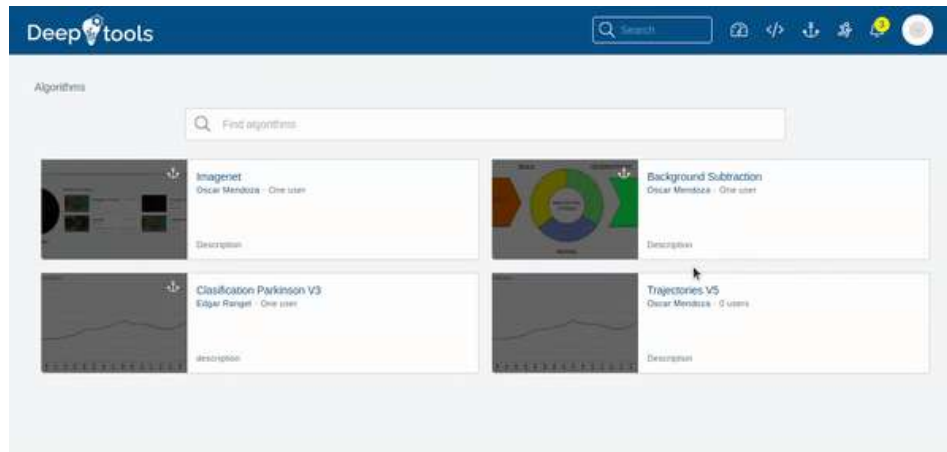


Debes escribir palabras clave para identificar los algoritmos de tu interés.



5.3 ¿Cómo veo más detalles de un algoritmo específico?

Ubica el puntero dentro del algoritmo que desees y dale click en el nombre del algoritmo.

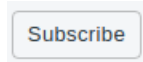


5.3.1 ¿Qué información puedo ver de un algoritmo?

Puedes ver una [descripción](#) del funcionamiento, el [creador](#), el tipo o las [extensiones](#) permitidas para los algoritmos, el [protocolo](#), una sección de [ejemplos](#), la cantidad de usuarios [suscritos](#) y una imagen representativa del [algoritmo](#).

5.3.2 ¿Cómo me suscribo a un algoritmo?

Teniendo en cuenta el [ítem anterior](#) ubíquese en la parte inferior, justo donde se ve el siguiente botón:



De click sobre este y aparecerán tres botones como los siguientes:



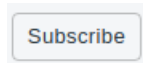
Si puedes visualizar estos tres botones significa que ya estás suscrito al algoritmo. Ahora estás lista para poder usarlo.

5.3.3 ¿Cómo elimino la suscripción a un algoritmo?

Teniendo en cuenta el [ítem 5.3](#) ubíquese en la parte inferior, justo donde se ven los siguientes botones:



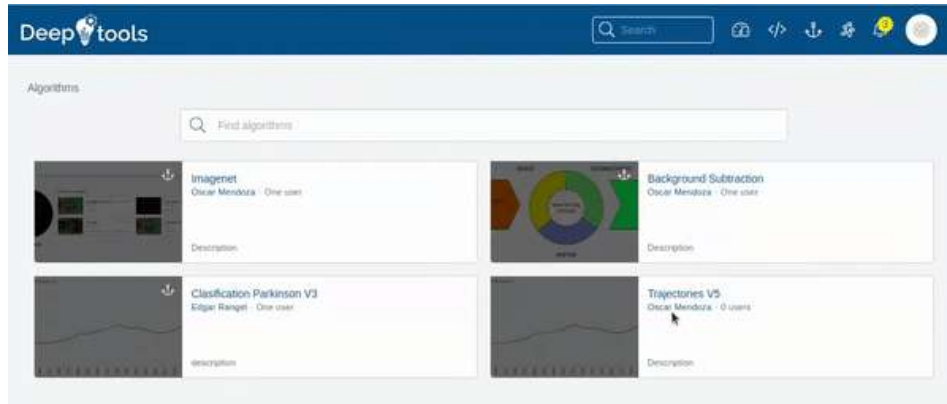
De click sobre el botón "Unsubscribe". Desaparecerán los tres botones y aparecerá el siguiente:



Si sólo puedes visualizar este botón significa que ya no estás suscrito al algoritmo.

5.4 ¿Cómo veo más detalles de un desarrollador específico?

Ubica el puntero dentro del algoritmo que desees y dale click en el nombre del desarrollador.



6. Módulo suscripciones

Para acceder al módulos de [suscripciones](#) debe ubicarse en la parte derecha de la [barra de navegación](#) y dar click sobre el botón:



Lo redirigirá hacia la siguiente vista:



En esta vista podrá visualizar las suscripciones y su estado:

 [Activo](#)

 [Mantenimiento](#)

6.1 Estados

6.1.1 Activo

Representa que un [algoritmo](#) se encuentra listo para ser usado.

6.1.2 Mantenimiento

Hubo algún inconveniente durante la [ejecución](#) de un [algoritmo](#) y el [desarrollador](#) está trabajando para solucionarlo.

6.2 ¿Cómo ver más detalles de un algoritmo específico?

Para ver más detalles de un algoritmo debe realizarse al igual que en el [módulo algoritmos](#).

6.3 ¿Cómo ver más detalles de un desarrollador específico?

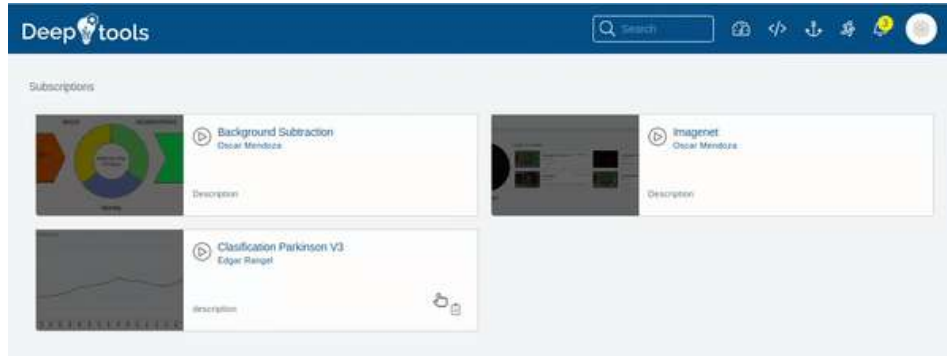
Para ver más detalles de un desarrollador debe realizarse al igual que en el [módulo algoritmos](#).

6.4 ¿Cómo ver todas las pruebas que he realizado a un algoritmo suscrito?

Para visualizar todas las [pruebas](#) de un [algoritmo](#) se debe ubicar sobre la caja, la situar el puntero sobre ella se mostrará el siguiente icono:



De click sobre este y lo redirecciona a una vista con todos las [pruebas](#) realizadas por este algoritmo.

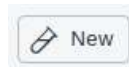


6.4.1 ¿Cómo administrar las pruebas que he realizado?

Las [pruebas](#) pueden ser creadas, visualizadas, clonadas y eliminadas, cabe aclarar que una vez eliminada la prueba no podrá ser recuperada.

6.4.1.1 ¿Cómo crear una prueba?

Teniendo en cuenta el [ítem 4.3](#) nos ubicamos en la parte inferior justo en el botón:



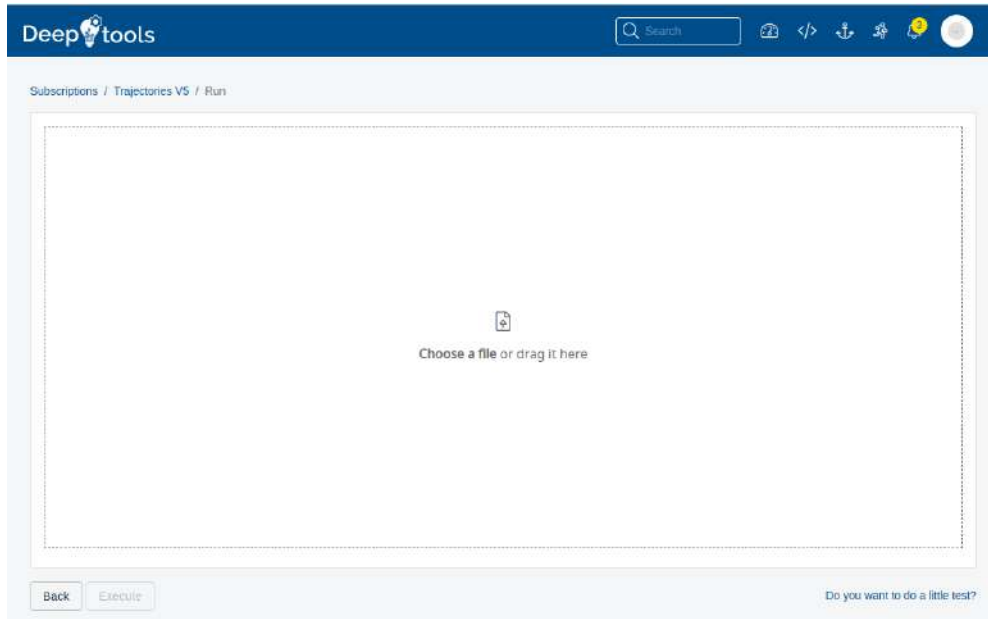
De click en el, esto lo redireccionar a la siguiente vista:



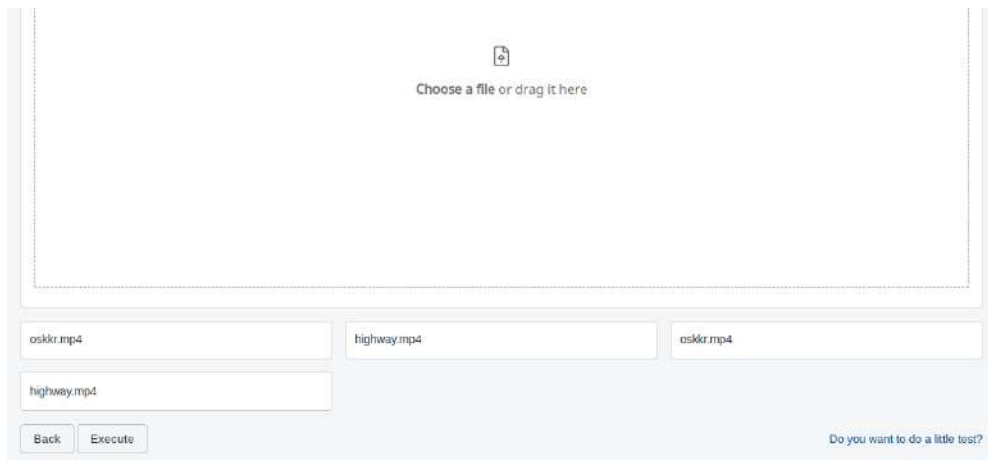
En esta vista se muestra el [protocolo](#) y en la parte inferior izquierda se ve los siguientes botones:



De click sobre el botón “Next”, esto lo redirigirá a la siguiente vista:



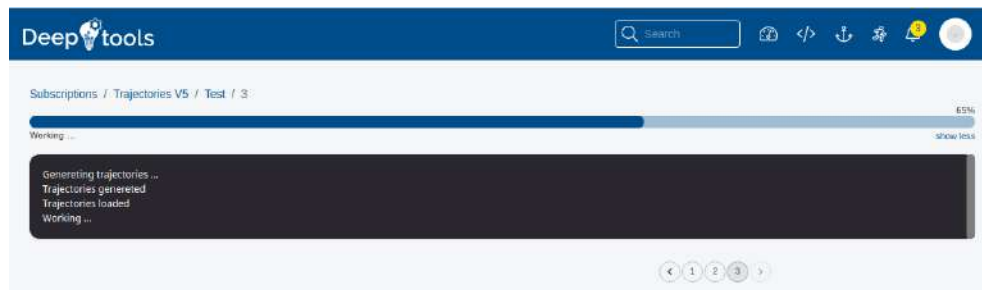
Seleccione el tipo de archivo especificado en las [extensiones](#) tal como lo hacen en el [ítem 4.2](#). Una vez hecho se de mostrará una compilación de todos los archivos subidos:



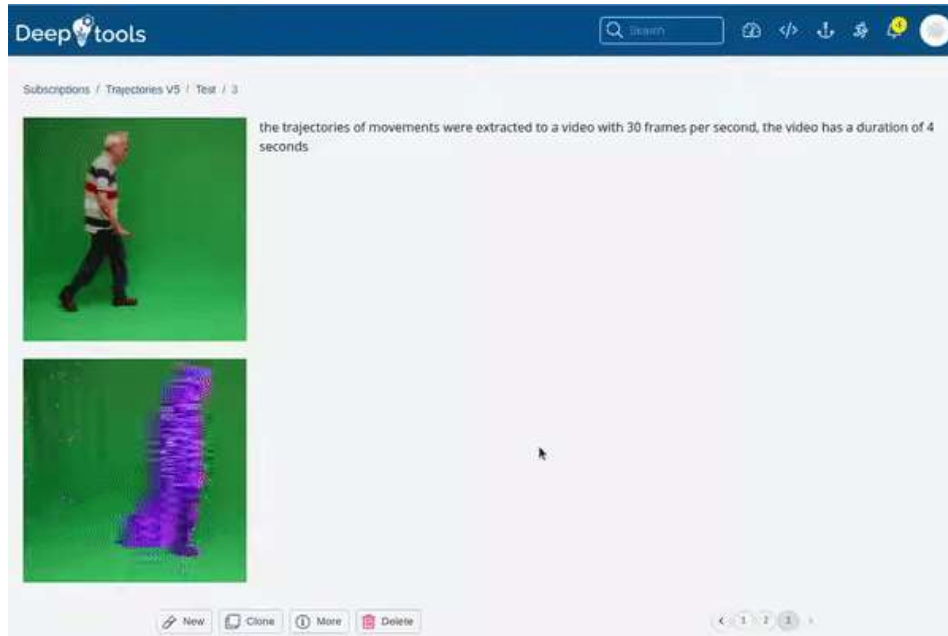
Una vez haya seleccionado los archivos especificados, dé clic sobre el botón [“Execute”](#). Esto lo redireccionará la siguiente vista:



En esta vista se visualiza un elemento importante durante el proceso de [ejecución](#) del [algoritmo](#). Debajo de la [miga de pan](#) se ve una barra de progreso que simboliza el progreso de la ejecución del algoritmo, además de esta se ve el porcentaje exacto del proceso, en la siguiente imagen de ejemplo se muestra que lleva el 65% de todo el proceso.

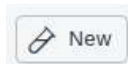


Una vez llega al 100% se podrá visualizar el resultado del [algoritmo](#).



6.4.1.1.1 ¿Cómo crear una prueba con datos de prueba?

Teniendo en cuenta el [ítem 4.3](#) nos ubicamos en la parte inferior justo en el botón:



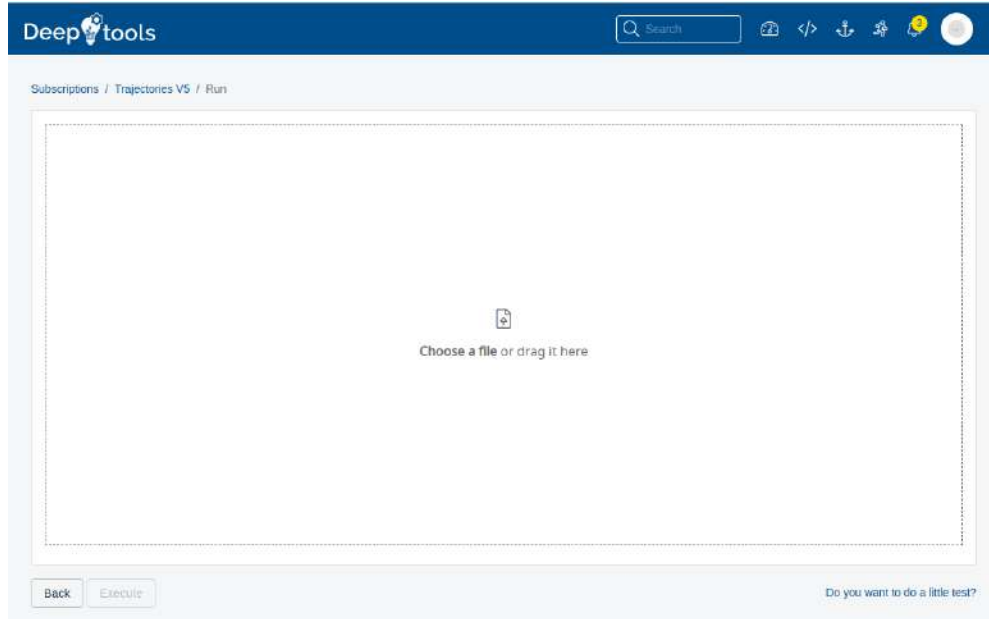
De click en el, esto lo redireccionar a la siguiente vista:



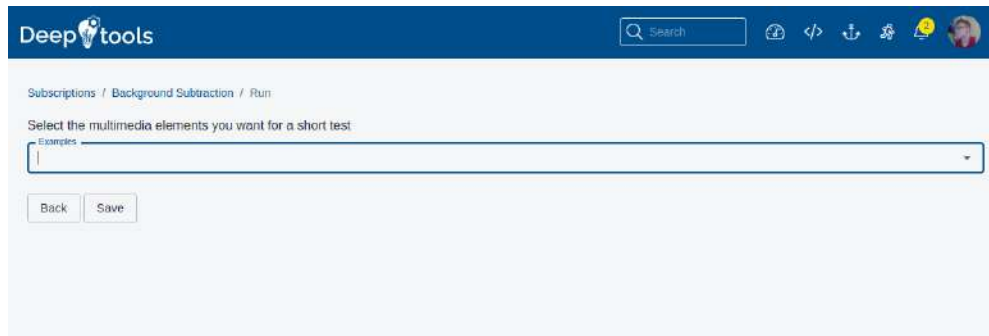
En esta vista se muestra el [protocolo](#) y en la parte inferior izquierda se ve los siguientes botones:



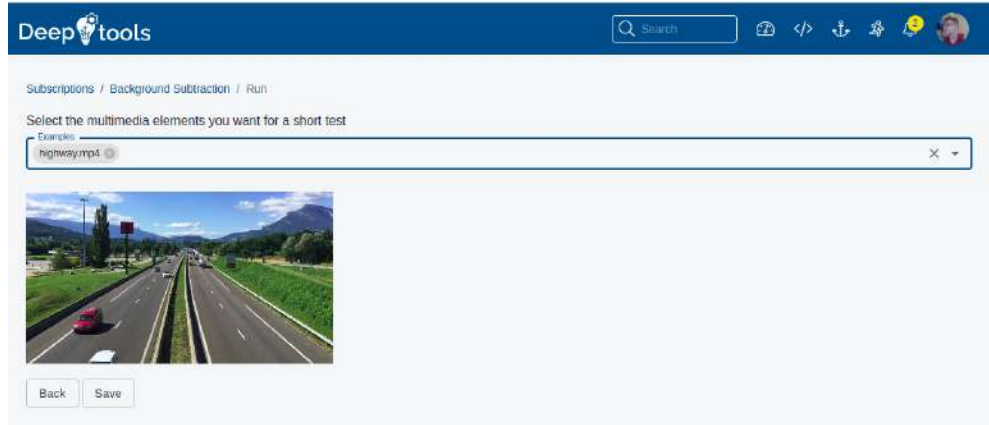
De click sobre el botón "Next", esto lo redirigirá a la siguiente vista:



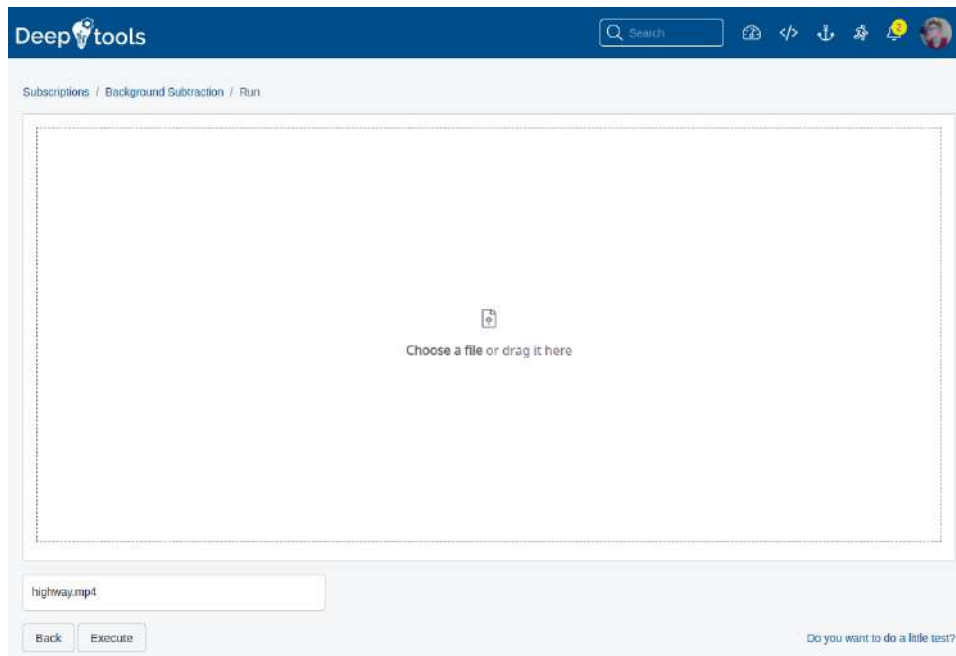
En la parte inferior derecha está el botón “Do you want to do a little test?”, de click sobre él. Esto lo redirecciona a la siguiente vista:



Seleccione los elementos que desea usar para hacer la prueba.



Haga clic sobre el botón “Save”. Esto lo llevará a la vista anterior pero tendrá los elementos cargados y listos para realizar la [prueba](#).



De click sobre el botón “Execute” y espere a que finalice la [ejecución](#) del [algoritmo](#).

6.4.1.2 ¿Cómo ver una prueba?

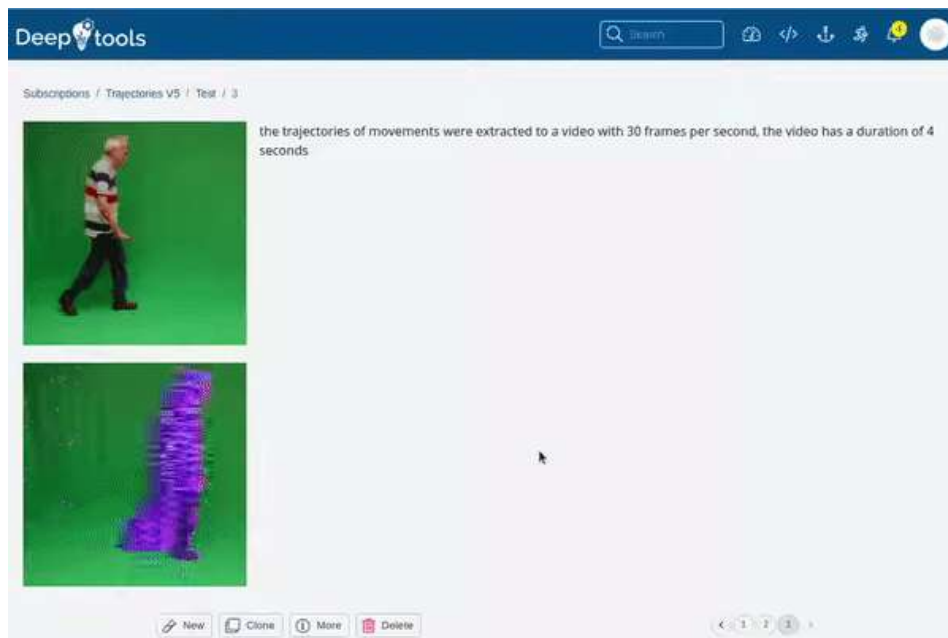
Teniendo en cuenta el [ítem 6.4](#) nos ubicamos en la prueba, una vez hecho esto aparecerán los siguiente iconos:



Le damos click al siguiente icono:



Nos redirecciona a la vista de visualización de resultados del algoritmo.



6.4.1.3 ¿Cómo clonar una prueba?

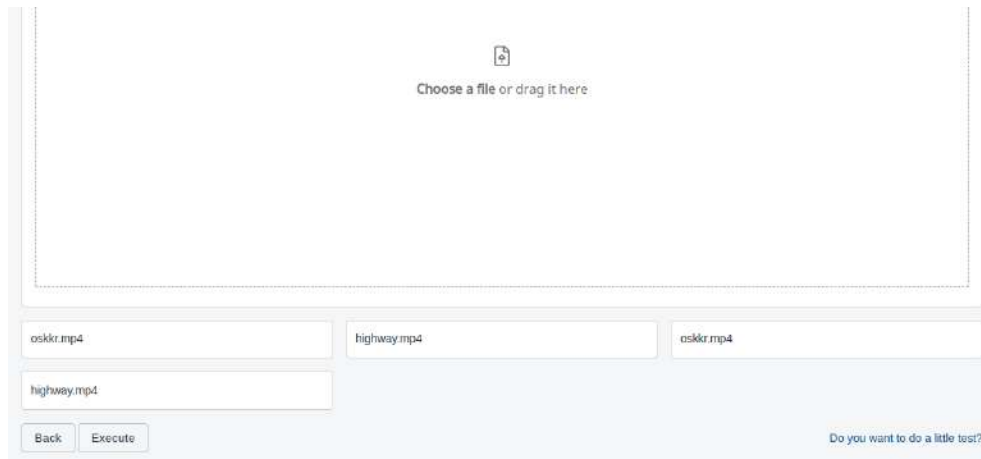
Teniendo en cuenta el [ítem 6.4](#) nos ubicamos en la prueba, una vez hecho esto aparecerán los siguiente iconos:



Le damos click al siguiente icono:



Esto nos redirecciona a la vista de [ejecución](#). Cabe aclarar que lo que se quiere hacer con esta opción es la posibilidad de crear una [prueba](#) con los mismos archivos que se usaron en la prueba seleccionada.



Para finalizar la clonación oprimimos el botón “Execute”.

6.4.1.4 ¿Cómo eliminar una prueba?

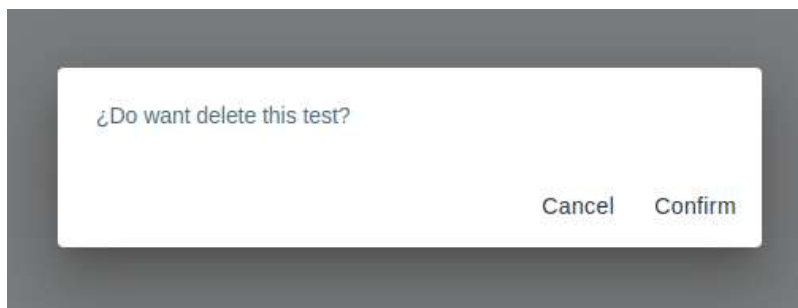
Teniendo en cuenta el [ítem 5.4](#) nos ubicamos en la prueba, una vez hecho esto aparecerán los siguiente iconos:



Le damos click al siguiente icono:



Esto nos desplegará una caja de diálogo preguntando si está seguro de eliminar la [prueba](#).



Seleccionamos la opción “Confirm”, y desaparecerá la prueba seleccionada.

7. ¿Cómo puedo ver los algoritmos que están en ejecución?

Para acceder a los [algoritmos](#) en [ejecución](#) debe ubicarse en la parte derecha de la [barra de navegación](#) justo donde se ve el siguiente ícono:



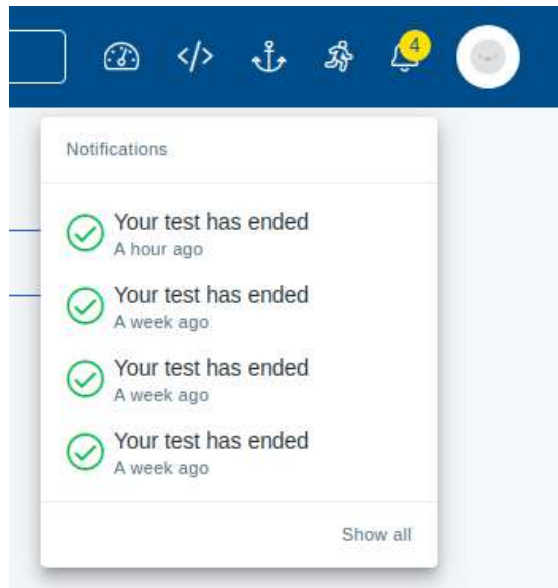
De click sobre él, esto lo redireccionará a una vista con una [compilación](#) de todos los [algoritmos](#) en [ejecución](#).

8. ¿Cómo accedo a las notificaciones en tiempo real?

Las notificaciones en tiempo real están activadas, esto se verá reflejado en un indicador numérico en la [barra de navegación](#):



Este número indica el número de notificaciones activas. Al clicar sobre este botón se desplegará un menú con las notificaciones activas.



Las notificaciones cuentan de tres partes, el título, el tipo y la descripción, si le da click a alguna de estas será redireccionado a la notificación a atender. En este ejemplo será redireccionado a la visualización de una [prueba](#).

Anexo F. Manuales del desarrollador

A continuación, se presenta el manual de usuario para el role 'desarrollador'.



Manual del desarrollador

Juan Sebastian Marcon Caballero
Universidad Industrial de Santander
Biomedical Imaging, Vision and Learning Laboratory (BIVL2ab)

Contenido

1 Conoce DeepTools	3
1.1 Conceptos básicos	3
1.1.1 Algoritmo	3
1.1.2 Prueba de ejecución	3
1.1.2.1 Prueba de ejecución finalizada	3
1.1.2.2 Prueba de ejecución iniciada	4
1.1.3 Suscripción	4
1.2 ¿Cómo empiezo a usar DeepTools?	4
1.2.1 Creación de cuenta	4
1.2.2 ¿Cómo empiezo si ya tengo una cuenta?	6
1.2.3 Seleccione un algoritmo de su interés	6
2. ¿Cómo ingreso a la plataforma?	7
3. Partes de la plataforma	8
3.1 Barra de navegación	8
3.2 Miga de pan	9
4. ¿Cómo administrar mi perfil de usuario?	9
4.1 ¿Cómo cierro mi sesión?	9
4.2 ¿Cómo ver mi perfil de desarrollador?	9
4.3 ¿Cómo modificar mi perfil de usuario?	10
5 Módulo algoritmos	12
5.1 ¿Cómo veo más detalles de un algoritmo específico?	12
5.2 ¿Qué información puedo ver de un algoritmo?	13
5.3 Estados	13
5.3.1 Activo	13
5.3.2 Inactivo	13
5.4.3 Desplegado	13
5.4 ¿Cómo administrar mis algoritmos?	14
5.4.1 ¿Cómo detener un algoritmo?	14
5.4.3 ¿Cómo eliminar un algoritmo?	14
5.5 ¿Cómo ver todas las pruebas que he realizado a un algoritmo suscrito?	14
5.5.1 ¿Cómo administrar las pruebas que he realizado?	15
5.5.1.1 ¿Cómo crear una prueba?	15
5.5.1.2 ¿Cómo ver una prueba?	18
5.5.1.3 ¿Cómo clonar una prueba?	19
5.5.1.4 ¿Cómo eliminar una prueba?	20
	1

5.6 ¿Cómo desplegar un algoritmo?	21
5.6.1 Protocolo de recolección de datos	21
5.6.2 Definición de los elementos	22
5.6.2.1 Elementos de entrada	22
5.6.2.1.1 Elementos de entrada a procesar	22
5.6.2.2 Elementos de salidas	22
5.6.2.2.1 Resultado	23
5.6.2.2.2 Grafico	23
5.6.2.2.3 Multimedia	23
5.6.3 Detalles del modelo	23
5.6.3.1 Imagen Docker	23
5.6.3.1 Direccion del directorio	24
5.6.3.2 Nombre del archivo	24
5.6.3.3 Nombre de la clase	24
5.6.3.4 Extensiones de los elementos	24
5.6.3.5 Imagen del algoritmo	24
5.6.3.6 Nombre del algoritmo	24
5.6.3.7 Descripción del algoritmo	25
5.7 ¿Cómo crear una imagen docker compatible con el software?	25
5.7.1 Directorio de trabajo	25
5.7.2 Archivo raiz	25
5.7.2.1 Argumento 'input'	25
5.7.2.2 Argumento 'output'	25
5.7.2.3 Metodo 'run'	26
5.7.2.4 Propiedades de la clase	26
5.7.2.4.1 Propiedad 'input__'	26
5.7.2.4.2 Propiedad 'output__'	26
5.7.2.4.3 Propiedad 'result__'	26
5.7.2.4.4 Propiedad 'graph__'	26
5.7.2.4.5 Propiedad 'state__'	26
5.7.2.4.6 Propiedad 'description__'	26
5.7.2.5 Recomendaciones	26
5.7.3 Creación de la imagen docker	27
7. ¿Cómo puedo ver los algoritmos que están en ejecución?	27
8. ¿Cómo accedo a las notificaciones en tiempo real?	27

1 Conoce DeepTools

Deeptools es un prototipo de aplicativo web que busca almacenar las soluciones basadas en IA y Visión por computador desarrollados por los integrantes del grupo Bivl2ab para ofrecerlos a la comunidad, de esta forma existirá un espacio amigable para que desarrolladores de estas soluciones puedan mostrar su trabajo y ser utilizado por usuarios interesados.

Deeptools se esfuerza por brindar una experiencia de usuario increíble y al mismo tiempo un amplio espectro de soluciones para los usuarios, con el fin de construir un entorno ameno para los distintos tipos de usuarios.

1.1 Conceptos básicos

En este ítem hablaremos un poco acerca de los conceptos más generales que conforman el software, conceptos indispensables para el entendimiento del mismo.

1.1.1 Algoritmo

Un algoritmo es una pieza de código o conjunto de instrucciones que represente una solución a una problemática específica, pero en este caso para nosotros es un poco más acotada, cuando hablamos de algoritmo nos referimos a una pieza de código que implemente técnicas de IA y Visión por computador, esto debido a que este sistema fue diseñado solamente para estos algoritmos.

Estas piezas de código y los usuarios son el corazón de nuestro aplicativo, siéntase con la capacidad de probar y usar todos los algoritmos disponibles, esto ayuda a que crezca nuestra comunidad.

1.1.2 Prueba de ejecución

Una prueba de ejecución es la puesta en marcha de un algoritmo por parte de un usuario, esta contiene datos de estado, datos tratados, datos generados, resultados, entre otros. Estos datos dependen del algoritmo ejecutado, dicho en otras palabras, es el registro generado por la ejecución de un algoritmo específico.

Estas pruebas tienen dos distintos estados:

1.1.2.1 Prueba de ejecución finalizada

Este estado representa una prueba ya finalizada, dado esto está lista para ser observada, ya contiene los resultados del algoritmo.

1.1.2.2 Prueba de ejecución iniciada

Este estado representa una prueba que acaba de ser creada, aún no podemos ver sus resultados, se está ejecutando el algoritmo para generar nuestros resultados.

1.1.3 Suscripción

La suscripción es la acción que realiza un usuario cuando quiere recibir una serie de servicios o productos. Para DeepTools es la relación que existe entre un usuario y un [algoritmo](#), esto quiere decir que un usuario no puede usar algoritmo si no tiene dicha suscripción, es nuestra forma de decirle a los creadores de algoritmos cuántas personas usan su algoritmo.

1.2 ¿Cómo empiezo a usar DeepTools?

Queremos que sea lo más fácil posible comenzar a usar DeepTools, por eso te guiaremos en estos primeros pasos.

1.2.1 Creación de cuenta

Si ya tiene una cuenta en DeepTools puede saltarse este item, continúe con [¿Cómo empiezo si ya tengo una cuenta?](#).

El primer paso para iniciar a usar DeepTools es la creación de una cuenta, para estos debemos dirigirnos a la siguiente [dirección](#) en el navegador [Google Chrome v86+](#):

Aparecerá una pantalla como la siguiente:

Deep tools

Create Account

First name

Last name

Role
User

Email address

Password

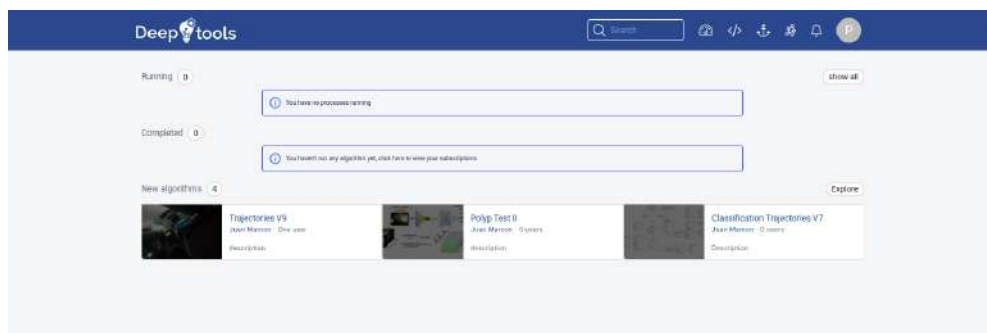
Confirm password

[Sign in](#) [Create account](#)

Agrega todos los datos solicitados en el formulario.

Nota: Es importante aclarar que esto es un paso a paso para crear una cuenta de usuario, si desea registrarse con un rol diferente solicite la documentación correspondiente.

En el campo rol, seleccione la opción por defecto *user*. Una vez diligenciado el formulario de click sobre el botón Crear cuenta ubicado en la parte inferior derecha, esto lo redireccionará a la siguiente pantalla:



Felicidades, ya tiene una cuenta en DeepTools.

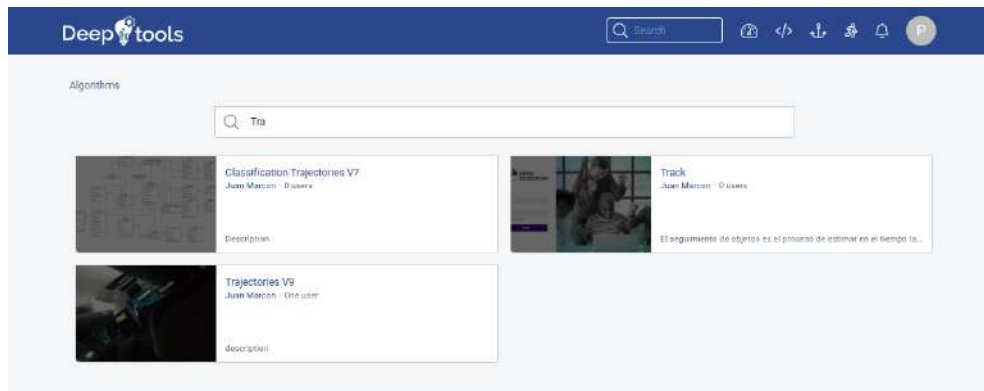
1.2.2 ¿Cómo empiezo si ya tengo una cuenta?

Si ya tiene una cuenta en DeepTools deberá iniciar su sesión, siga las instrucciones del ítem [¿Cómo ingreso a la plataforma?](#).

1.2.3 Seleccione un algoritmo de su interés

El siguiente paso para empezar a usar DeepTools es encontrar algún algoritmo de interés, esto dependerá del tamaño de nuestra comunidad al momento de hacer la búsqueda.

Existen dos maneras de hacer dicha búsqueda, la primera es mediante la [barra de búsqueda](#) y la otra es mediante la caja de búsqueda en la sección de los algoritmos, para más detalles vaya al ítem [¿Cómo busco algoritmos y desarrolladores de mi interés?](#).



Seleccione un algoritmo dando click en su título, eso lo redirigirá a una vista con los detalles del algoritmo seleccionado, si desea más información de la utilidad de esta sección consulte [¿Cómo ver más detalles de un algoritmo específico?](#).

Una vez realizado, deberá suscribirse a este para poder crear una [prueba de ejecución](#) con datos reales, o si desea una prueba de laboratorio consulte [¿Cómo crear una prueba con datos de prueba?](#)

2. ¿Cómo ingreso a la plataforma?

Para acceder a la plataforma Deeptools se debe ingresar la siguiente dirección en el navegador [Google Chrome v86+](#):

Aparecerá una pantalla como la siguiente:



Deep tools

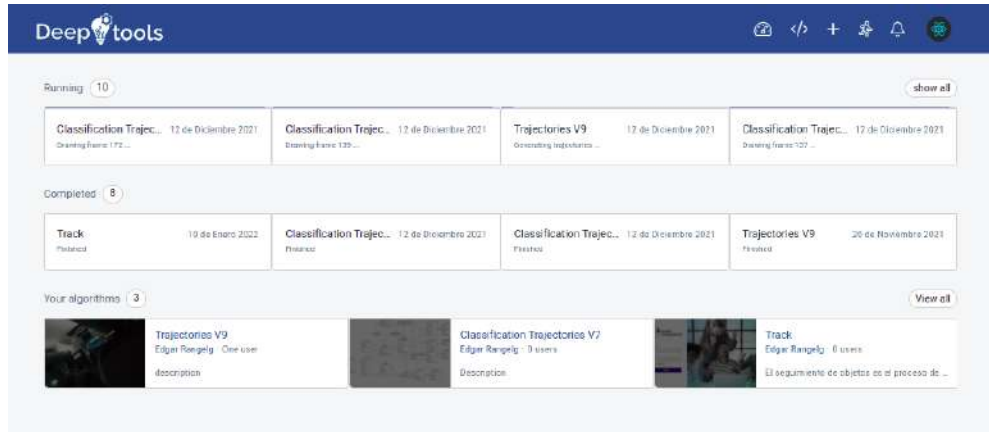
Sign In

Email address

Password

[Create account](#)

Después de ingresar su email registrado y su contraseña, presione el botón “Sign in now”. Debe aparecer una pantalla similar a la siguiente:



Nota: Para las siguientes funcionalidades, se asume que usted ya se encuentra autenticado en la plataforma Deeptools.

3. Partes de la plataforma

En esta plataforma le damos definiciones a diferentes partes representativas, con el fin de que los usuarios que se basen en esta guía entiendan cómo ubicarse durante toda la extensión del documento.

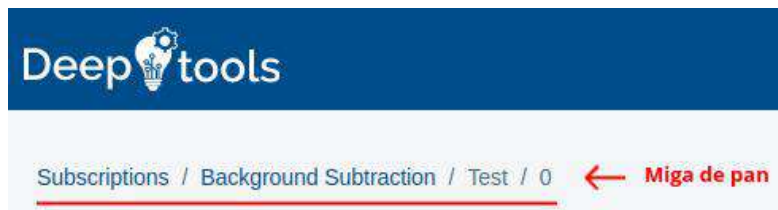
3.1 Barra de navegación

La barra de navegación está ubicada en la parte superior, cabe resaltar que esta es de color azul.



Esta barra cuenta con los botones necesarios para navegar.

3.2 Miga de pan



La [miga de pan](#) es el seguimiento de la navegación dentro de la plataforma. Está ubicado un poco más abajo de la [barra de navegación](#).

4. ¿Cómo administrar mi perfil de usuario?

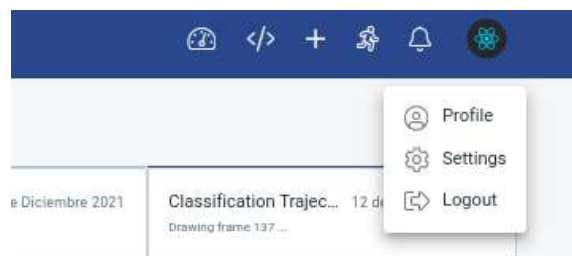
Si requiere administrar su perfil de usuario, ubique el puntero en la [barra de navegación](#) hacia la derecha en el botón donde se ve su foto de perfil o en su defecto la siguiente imagen:



De click en él. Al hacerlo, se desplegará un menú con diferentes opciones.

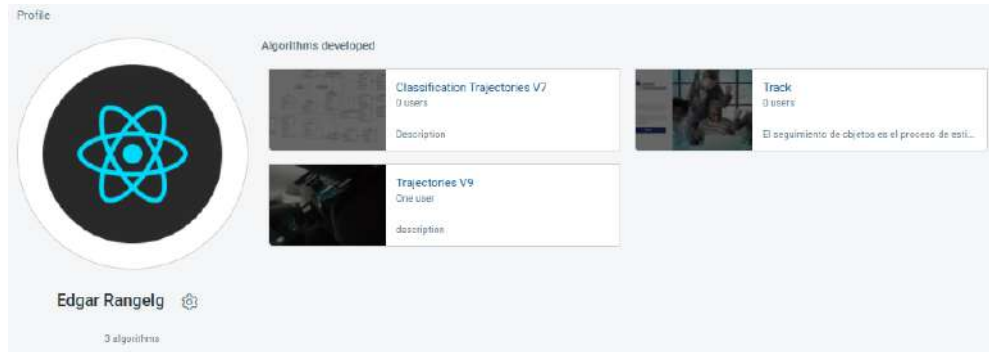
4.1 ¿Cómo cierro mi sesión?

Partiendo del [paso anterior](#). Seleccione la opción "Logout". Esto lo redireccionará a la vista ingreso.



4.2 ¿Cómo ver mi perfil de desarrollador?

Partiendo del [ítem 4](#). Seleccione la opción "Profile". Se mostrará una vista donde podrá visualizar los datos ingresados y sus respectivos algoritmos.



4.3 ¿Cómo modificar mi perfil de usuario?

Teniendo en cuenta el [item 4](#). Seleccione la opción “settings”. Se mostrará una ventana similar a la siguiente pantalla, en la cual se deben modificar los datos que desee.

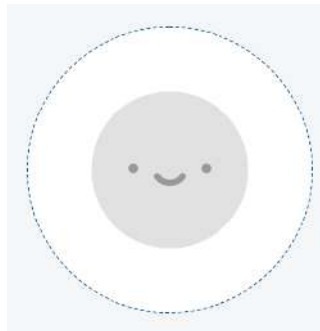
First name
Juan Sebastian

Last name
Marcon Caballero

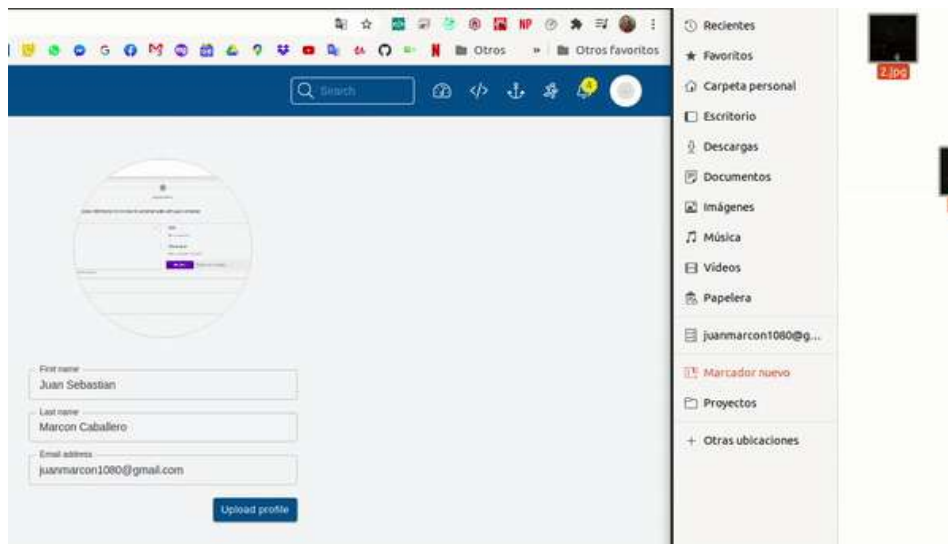
Email address
example@deeptools.com

Update profile

Si desea cambiar la foto de perfil, deberá poner el puntero encima de la foto actual, esto desplegará un cambio de estilo y color en los bordes indicando que este botón es una entrada de imagen.



Existen dos formas de seleccionar la imagen, la primera es haciendo click sobre la imagen, esto desplegará una administrador de archivos para que seleccione la imagen. La otra opción se basa en el arrastre. DeepTools provee la opción de arrastrar la imagen en cualquiera elemento de selección de archivos.



Una vez realizadas las modificaciones deberá dar click en el botón "Update profile" para guardar los cambios realizados. Esta acción lo llevará al [perfil de usuario](#).

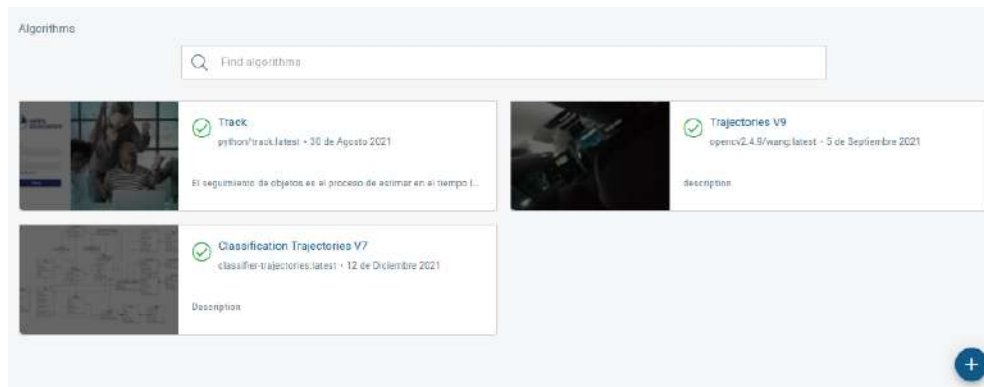
5 Módulo algoritmos

El módulo de algoritmos son todos los [algoritmos](#). Además esta vista provee una caja de texto para poder filtrar entre estos algoritmos.

Para acceder a este debes ubicarte en la parte derecha de la [barra de navegación](#) y dar click en el siguiente icono:

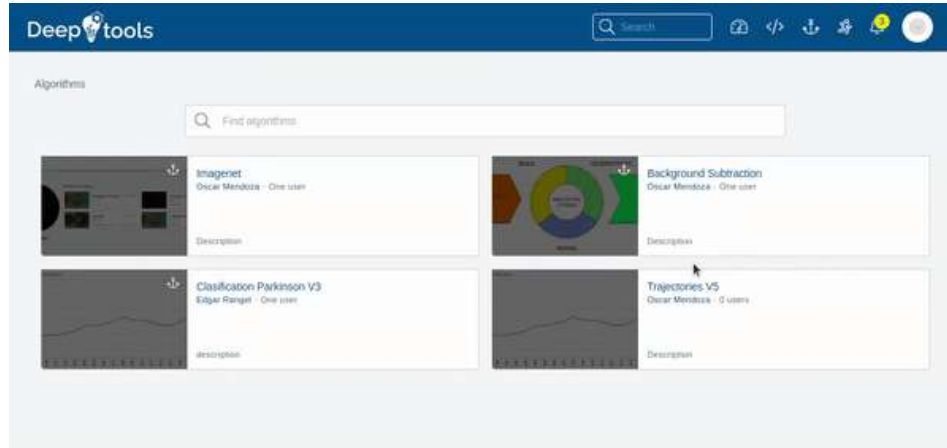


Esto lo redirigirá hacia una vista como la siguiente:



5.1 ¿Cómo veo más detalles de un algoritmo específico?

Ubica el puntero dentro del algoritmo que desees y dale click en el nombre del algoritmo.



5.2 ¿Qué información puedo ver de un algoritmo?

Puedes ver una [descripción](#) del funcionamiento, el tipo o las [extensiones](#) permitidas para los algoritmos, el [protocolo](#), una sección de [ejemplos](#), la representación visual, los metadatos del algoritmo, los usuarios [suscritos](#) y una imagen representativa del [algoritmo](#).

5.3 Estados

5.3.1 Activo

Representa que un [algoritmo](#) se encuentra listo para ser usado.

5.3.2 Inactivo

Representa que un [algoritmo](#) está pausado, por razones de mantenimiento o error.

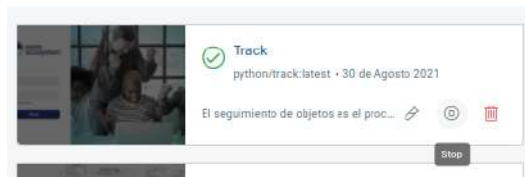
5.4.3 Desplegado

El algoritmo acaba de ser creado, pero debe agregarse una prueba para validar su funcionamiento.

5.4 ¿Cómo administrar mis algoritmos?

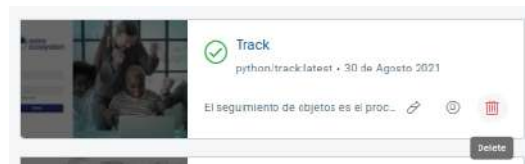
5.4.1 ¿Cómo detener un algoritmo?

Para detener un algoritmo debes tener en cuenta que se detendrá una vez todas las ejecuciones terminen, y dejará de estar disponible para todos los usuarios suscritos. Debe dar click sobre el botón 'stop' del algoritmo que desea detener.



5.4.3 ¿Cómo eliminar un algoritmo?

Para eliminar un algoritmo debes tener en cuenta que se eliminará una vez todas las ejecuciones terminen, y dejará de estar disponible para todos los usuarios. Además cabe aclarar que todas las ejecuciones creadas por los usuarios suscritos no se eliminarán. Debe dar click sobre el botón *delete* del algoritmo que desea detener.

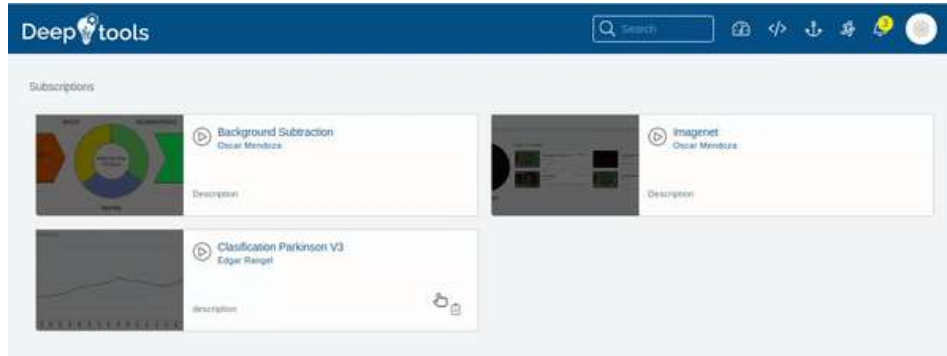


5.5 ¿Cómo ver todas las pruebas que he realizado a un algoritmo suscrito?

Para visualizar todas las [pruebas](#) de un [algoritmo](#) se debe ubicar sobre la caja, la situar el puntero sobre ella se mostrará el siguiente icono:



De click sobre este y lo redirecciona a una vista con todos las [pruebas](#) realizadas por este algoritmo.

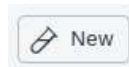


5.5.1 ¿Cómo administrar las pruebas que he realizado?

Las [pruebas](#) pueden ser creadas, visualizadas, clonadas y eliminadas, cabe aclarar que una vez eliminada la prueba no podrá ser recuperada.

5.5.1.1 ¿Cómo crear una prueba?

Teniendo en cuenta el [ítem 4.3](#) nos ubicamos en la parte inferior justo en el botón:



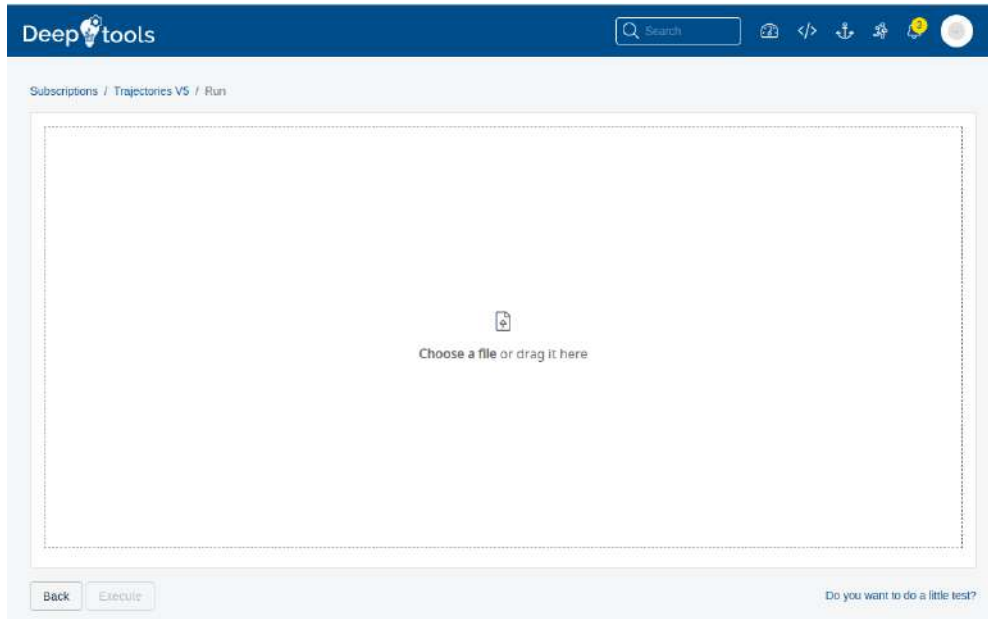
De click en el, esto lo redireccionar a la siguiente vista:



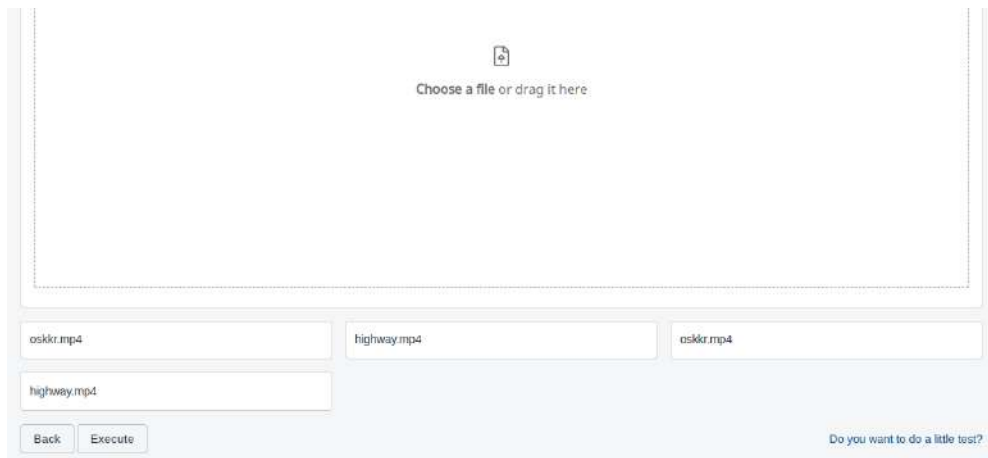
En esta vista se muestra el [protocolo](#) y en la parte inferior izquierda se ve los siguientes botones:



De click sobre el botón “Next”, esto lo redirigirá a la siguiente vista:



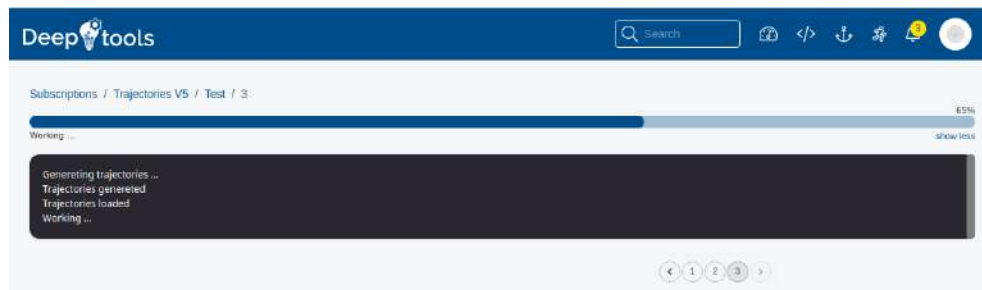
Seleccione el tipo de archivo especificado en las [extensiones](#) tal como lo hacen en el [ítem 4.2](#). Una vez hecho se de mostrará una compilación de todos los archivos subidos:



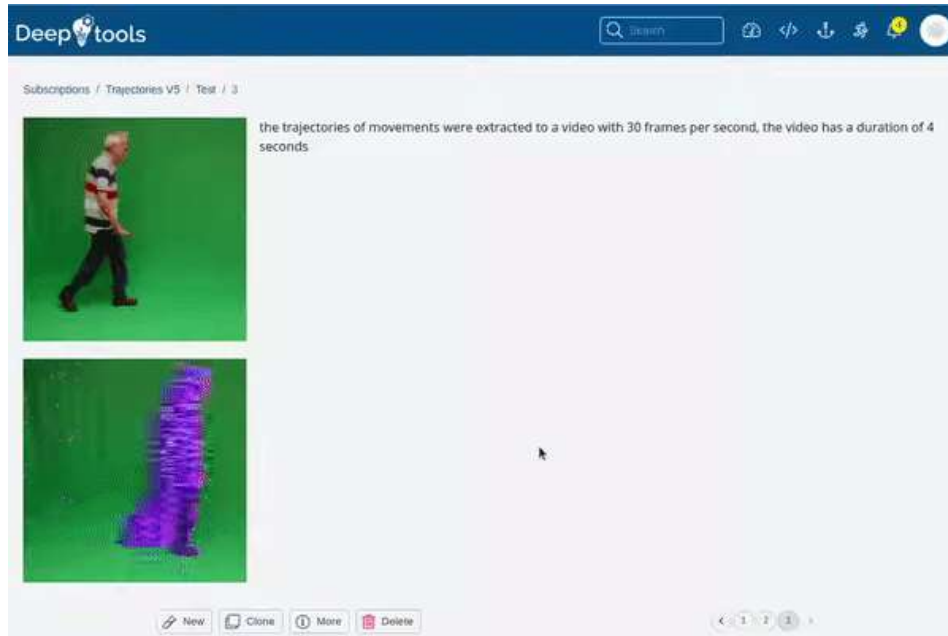
Una vez haya seleccionado los archivos especificados, dé clic sobre el botón [“Execute”](#). Esto lo redireccionará la siguiente vista:



En esta vista se muestra un elemento importante durante el proceso de [ejecución](#) del [algoritmo](#). Debajo de la [miga de pan](#) se ve una barra de progreso que simboliza el progreso de la ejecución del algoritmo, además de esta se ve el porcentaje exacto del proceso, en la siguiente imagen de ejemplo se muestra que lleva el 65% de todo el proceso.



Una vez llega al 100% se podrá visualizar el resultado del [algoritmo](#).



5.5.1.2 ¿Cómo ver una prueba?

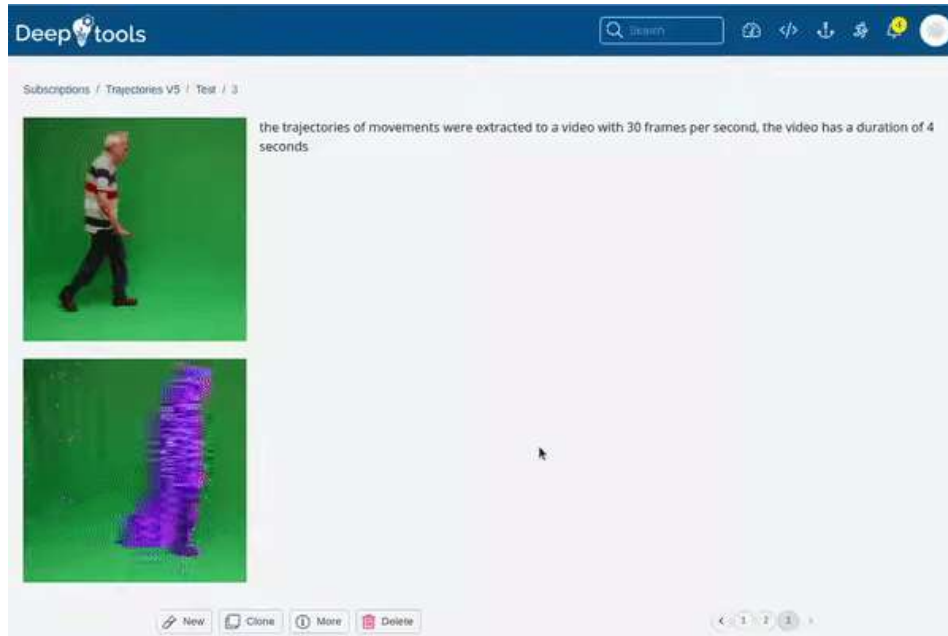
Teniendo en cuenta el [ítem 6.4](#) nos ubicamos en la prueba, una vez hecho esto aparecerán los siguiente iconos:



Le damos click al siguiente icono:



Nos redirecciona a la vista de visualización de resultados del algoritmo.



5.5.1.3 ¿Cómo clonar una prueba?

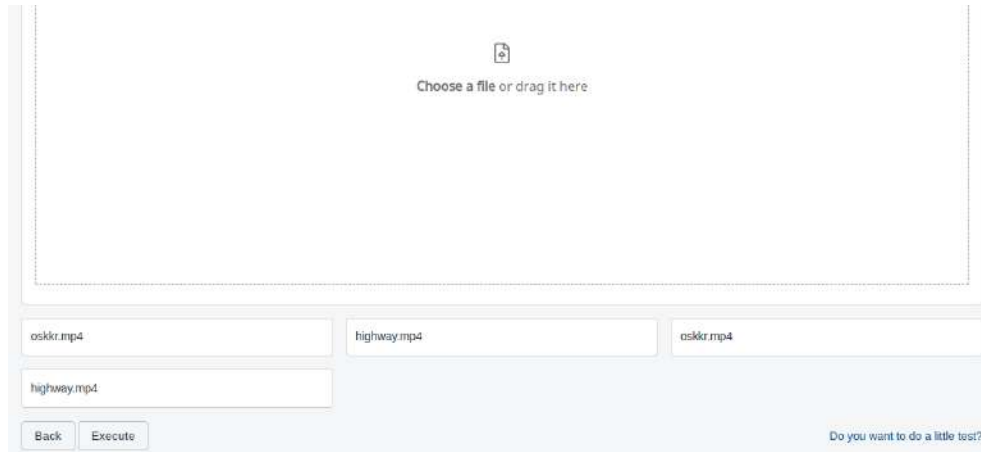
Teniendo en cuenta el [ítem 6.4](#) nos ubicamos en la prueba, una vez hecho esto aparecerán los siguiente iconos:



Le damos click al siguiente icono:



Esto nos redirecciona a la vista de [ejecución](#). Cabe aclarar que lo que se quiere hacer con esta opción es la posibilidad de crear una [prueba](#) con los mismos archivos que se usaron en la prueba seleccionada.



Para finalizar la clonación oprimimos el botón “Execute”.

5.5.1.4 ¿Cómo eliminar una prueba?

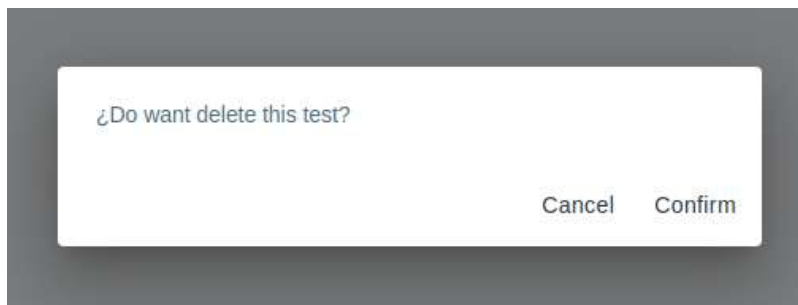
Teniendo en cuenta el [ítem 5.4](#) nos ubicamos en la prueba, una vez hecho esto aparecerán los siguiente iconos:



Le damos click al siguiente icono:



Esto nos desplegará una caja de diálogo preguntando si está seguro de eliminar la [prueba](#).



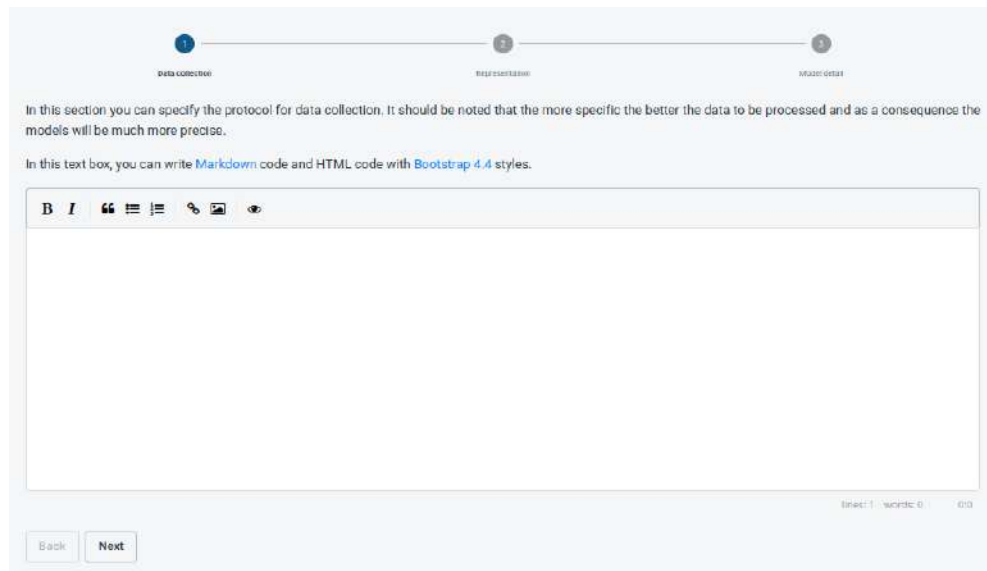
Seleccionamos la opción “Confirm”, y desaparecerá la prueba seleccionada.

5.6 ¿Cómo desplegar un algoritmo?

Para desplegar un algoritmo debe ubicarse en la parte derecha de la [barra de navegación](#) y dar click sobre el botón:



Lo redirigirá hacia la siguiente vista:



En la cual debe seguir los siguientes pasos:

5.6.1 Protocolo de recolección de datos

Durante el desarrollo de distintos algoritmos de IA se definen protocolos para la toma de datos, debido a la naturaleza de este tipo de algoritmos, si su algoritmo lo necesita agreguelo en este paso. Este campo soporta sintaxis Markdown y los estilos Bootstrap 4.4.

Después de esto haga click sobre el botón *Next*.

5.6.2 Definición de los elementos

En esta sección seleccionará los elementos necesarios para su algoritmo, de esta forma se crea la estructura de datos para la comunicación con el contenedor.

The screenshot displays a configuration interface for an algorithm. At the top, a progress bar indicates three stages: 1. Data collector, 2. Representation, and 3. Multimedia. Below this, a section titled 'Select the necessary elements for your algorithm' contains four checkboxes: 'Multimedia input' (checked), 'Descriptive response of the result' (unchecked), 'Multimedia output' (checked), and 'Graphic display of the result' (checked). The main content area features two video player thumbnails, each labeled 'Quantity: One video'. To the right is a bar chart with a peak around 35% and a label 'Prueba'. Below the chart is a text block about Lorem Ipsum. At the bottom, there are 'Back' and 'Next' buttons.

De click sobre el botón *Next*.

5.6.2.1 Elementos de entrada

Estos son los elementos proveídos por el usuario para procesar.

5.6.2.1.1 Elementos de entrada a procesar

Estos son los elementos que el usuario quiere procesar con el algoritmo, solo se permitirán imágenes y videos con los formatos: gif, jpg, png, jpeg, mp4(h264), los cuales son compatibles con el navegador especificado.

5.6.2.2 Elementos de salidas

Estos son los elementos resultantes del procesamiento de los elementos de entrada.

5.6.2.2.1 Resultado

Este elemento es el resultado del algoritmo, en este espacio puedes poner markdown, código html con estilos bootstrap 4.4.

5.6.2.2.2 Grafico

Este elemento es usado para representar estadísticas y cualquier representación generada por la librería [HIGHCHARTS](#).

5.6.2.2.3 Multimedia

En este campo se mostrarán los resultados de los elementos de entrada procesados.

5.6.3 Detalles del modelo

En este paso se pide información técnica muy específica sobre la construcción de nuestro algoritmo.

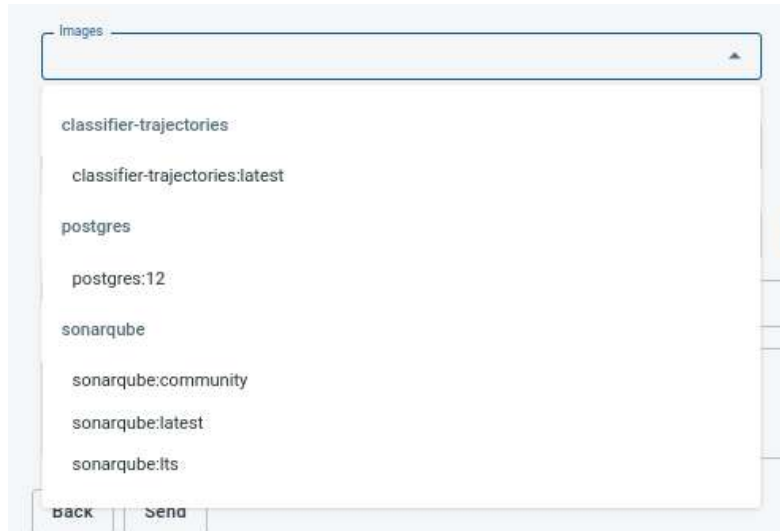
The screenshot shows a web form with a progress indicator at the top showing three steps: 1. Data extraction, 2. Representation, and 3. Model detail. The form is titled "Help us with some necessary information to be able to communicate with your container." and contains the following fields:

- Images:** A dropdown menu with the label "Images dockers" below it.
- Path:** A text input field with the label "Path to working" below it.
- Filename:** A text input field with the label "how is the file is named?" below it.
- Classname:** A text input field with the label "how is the class named?" below it.
- Video extensions:** A dropdown menu with the label "Video extensions" below it.
- Select:** A blue button.
- Select an image...:** A text input field.
- Name:** A text input field.
- Description:** A large text area.

At the bottom of the form, there are two buttons: "Back" and "Send".

5.6.3.1 Imagen Docker

En este campo debe seleccionarse la imagen docker con su algoritmo, al dar click se mostrará una lista de las imágenes docker disponibles.



5.6.3.1 Dirección del directorio

En este campo se debe especificar la dirección de la ruta de trabajo de la imagen docker, debe especificar una ruta absoluta.

5.6.3.2 Nombre del archivo

En este campo se debe especificar el nombre del archivo raíz junto con su extensión.

5.6.3.3 Nombre de la clase

En este campo se debe especificar el nombre de la clase de su archivo raíz.

5.6.3.4 Extensiones de los elementos

En este campo se detalla el tipo de archivo que pueden ingresar los usuarios suscritos al algoritmo.

5.6.3.5 Imagen del algoritmo

En este campo se selecciona una imagen que represente el algoritmo desarrollado.

5.6.3.6 Nombre del algoritmo

En este campo se ingresa una etiqueta o nombre que represente el algoritmo desarrollado.

5.6.3.7 Descripción del algoritmo

En este campo se ingresa una descripción del algoritmo desarrollado.

5.7 ¿Cómo crear una imagen docker compatible con el software?

Para la creación exitosa de una imagen docker funcional para la plataforma DeepTools debe cumplir con ciertos requisitos.

5.7.1 Directorio de trabajo

Debe seleccionar una ruta donde agregaras todos los archivos requeridos para la creación del contenedor, a esta ruta se le llamara Directorio de trabajo. Este directorio de trabajo debe tener acceso de escritura y debe ser una ruta absoluta.

5.7.2 Archivo raiz

Este archivo se encargará de la ejecución de todo el algoritmo, tenga en cuenta que debe estar ubicado en el directorio de trabajo. Este archivo debe tener una clase con ciertas especificaciones:

- El constructor de la función recibe dos argumentos (input, output).
- Debe tener un método llamado 'run', este se encargará de ejecutar el algoritmo.

5.7.2.1 Argumento 'input'

Esta es la dirección de los elementos de entrada o elementos a procesar, debe tomar esta y guardarla en la propiedad `input__`.

```
def __init__(self, input, output):  
    self.input = input
```

5.7.2.2 Argumento 'output'

Esta es la dirección de los elementos de salidas o elementos procesados, debe guardarla en la propiedad `output__`. Tenga en cuenta que esta dirección aún no existe deberá crearla el proceso de ejecución, en otras palabras debe guardar su archivo final en esa dirección y agregar a la propiedad `output__` el nombre del archivo generado.

```
def __init__(self, input, output):  
    self.output__ = output + "/output.mp4"
```

5.7.2.3 Metodo 'run'

En este método se desarrollará toda la lógica del algoritmo y una vez finalizado el método la instancia en ejecución deberá contener distintas propiedades para garantizar una óptima comunicación.

5.7.2.4 Propiedades de la clase

Para poder crear una óptima comunicación con el software deben existir ciertas propiedades ligadas a los elementos seleccionados durante el despliegue del algoritmo

5.7.2.4.1 Propiedad 'input__'

En esta propiedad debe guardar la dirección de los elementos de entrada o elementos a procesar.

5.7.2.4.2 Propiedad 'output__'

En esta propiedad debe guardar la dirección de los elementos de salida o elementos procesados.

5.7.2.4.3 Propiedad 'result__'

En esta propiedad debe guardar el contenido de la respuesta del algoritmo, por lo general texto o markdown, también soporta HTML con estilos bootstrap 4.4.

5.7.2.4.4 Propiedad 'graph__'

En esta propiedad debe guardar los elementos asociados al gráfico seleccionado en el despliegue, debe contener la estructura compatible con la librería [HIGHCHARTS](#).

5.7.2.4.5 Propiedad 'state__'

Esta propiedad se encargará de mantener el estado actual de la aplicación en una escalas del 0 al 100, donde 0 es el valor al iniciar el proceso y 100 es el valor bandera para finalizar el proceso, si por alguna razón este valor no llega a 100, el software se encargará de destruir la prueba y notificar al creador del algoritmo este fallo.

5.7.2.4.6 Propiedad 'description__'

Esta propiedad se encarga de informar al usuario que está pasando durante la ejecución del algoritmo, está ligada directamente al [estado](#), La descripción debería cambiar si el estado lo hace.

5.7.2.5 Recomendaciones

Tenga en cuenta que se estará trabajando con volúmenes compartidos entre contenedores docker y scripts python, no hay soporte para notebooks ni para el uso de librerías de interfaz de usuario.

5.7.3 Creación de la imagen docker

El aplicativo requiere imágenes docker, para crear estos debe ejecutar el comando `docker commit ${container_id} ${image_name}/${tag}` puede guiarse por la documentación de [Docker](#).

7. ¿Cómo puedo ver los algoritmos que están en ejecución?

Para acceder a los [algoritmos](#) en [ejecución](#) debe ubicarse en la parte derecha de la [barra de navegación](#) justo donde se ve el siguiente icono:



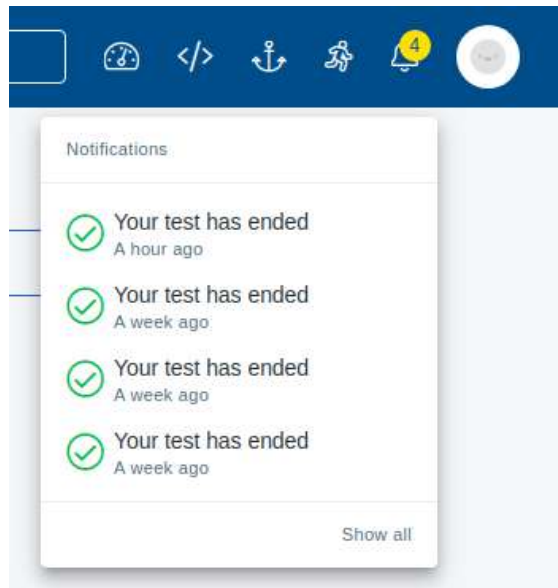
De click sobre él, esto lo redireccionará a una vista con una [compilación](#) de todos los [algoritmos](#) en [ejecución](#).

8. ¿Cómo accedo a las notificaciones en tiempo real?

Las notificaciones en tiempo real están activadas, esto se verá reflejado en un indicador numérico en la [barra de navegación](#):



Este número indica el número de notificaciones activas. Al clickear sobre este botón se desplegará un menú con las notificaciones activas.



Las notificaciones cuentan de tres partes, el título, el tipo y la descripción, si le da click a alguna de estas será redireccionado a la notificación a atender. En este ejemplo será redireccionado a la visualización de una [prueba](#).

Anexo G. Implementación del prototipo

Figura 14. CU00 - Iniciar Sesión



Deep tools

Sign In

Email address

Password

Create account Sign in now

Anexo H. Casos de uso

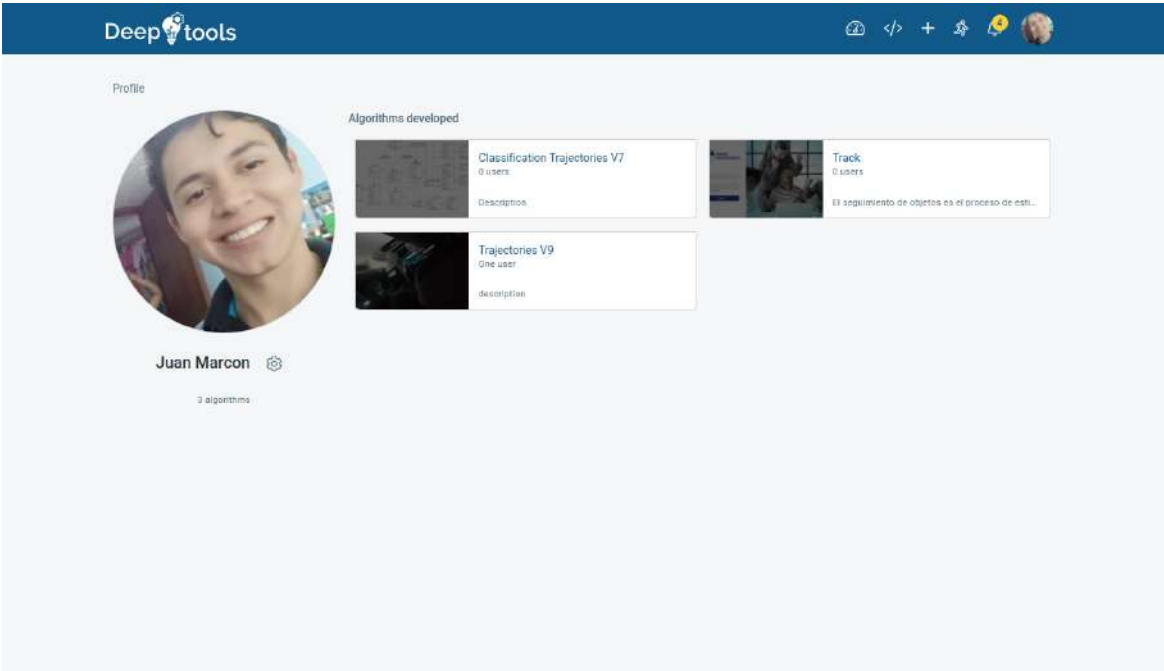
A continuación, se presenta el manual de usuario para el role 'usuario'.

Figura 15. CU02 - Registrar usuario



The image shows a registration form for 'Deep tools'. At the top is the logo 'Deep tools' with a lightbulb icon. Below it is the heading 'Create Account'. The form consists of several input fields: 'First name', 'Last name', a dropdown menu for 'Role' with 'User' selected, 'Email address', 'Password', and 'Confirm password'. At the bottom left is a 'Sign in' link, and at the bottom right is a blue 'Create account' button.

Figura 16. CU03 - Ver perfil



The image shows a user profile page for 'Juan Marcon' on the 'Deep tools' platform. The page has a dark blue header with the 'Deep tools' logo and navigation icons. The profile section includes a circular profile picture of Juan Marcon, his name 'Juan Marcon', a gear icon for settings, and a note that he has '3 algorithms'. To the right, under the heading 'Algorithms developed', there are three algorithm cards: 'Classification Trajectories V7' (0 users), 'Trajectories V9' (One user), and 'Track' (0 users). The 'Track' card has a partial description: 'El seguimiento de objetos es el proceso de esti...'. The background is a light blue gradient.

Figura 17. CU04 - Editar perfil

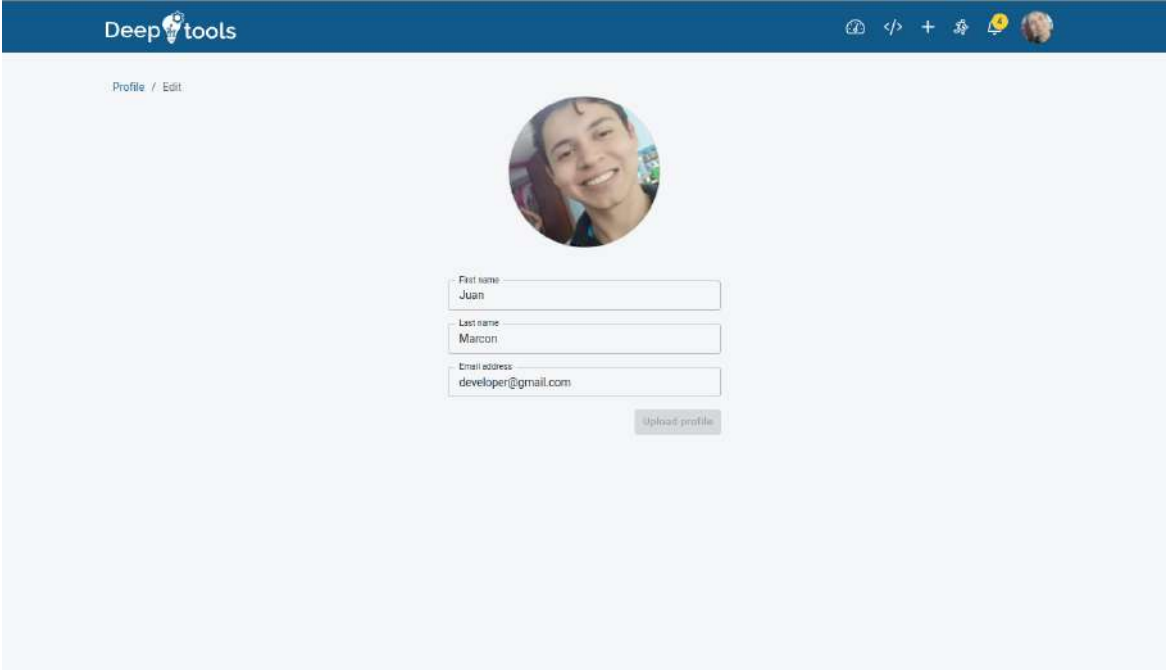


Figura 18. CU05 - Ver perfil de desarrolladores

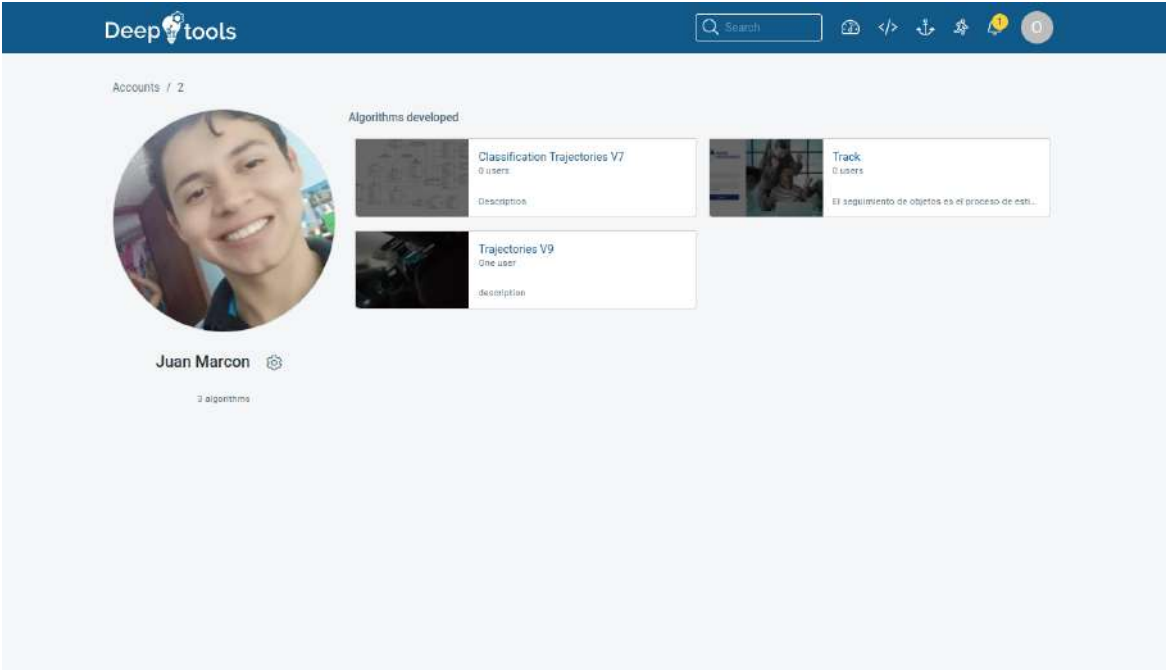


Figura 19. CU06 - Listar notificaciones

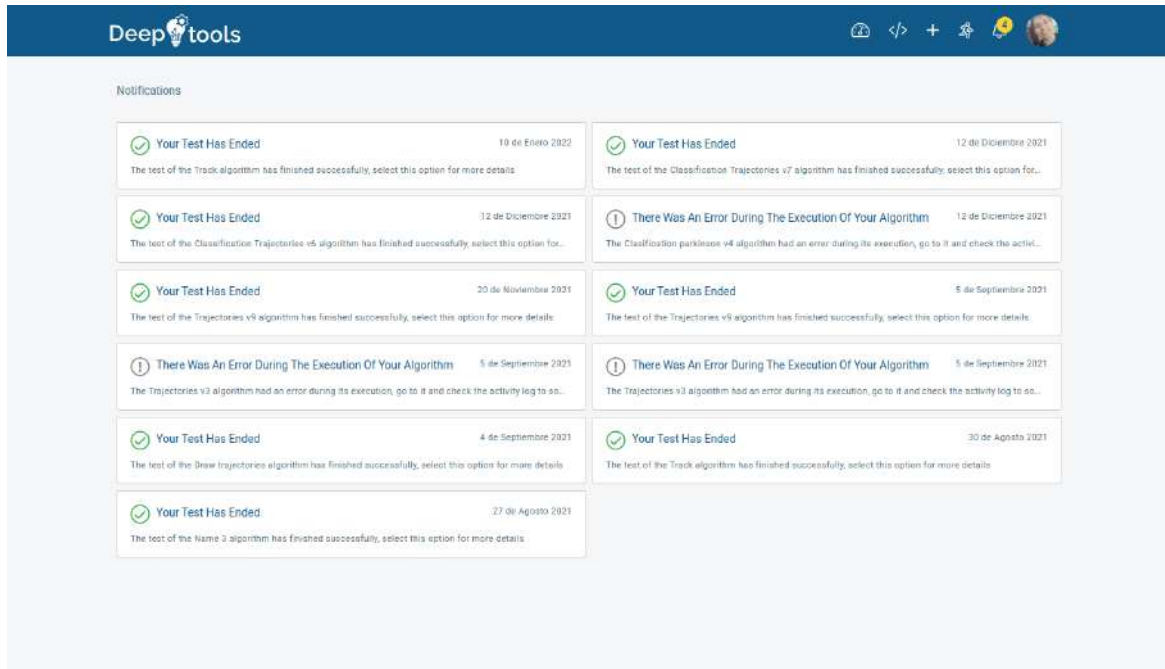


Figura 20. CU08 - Suscribirse

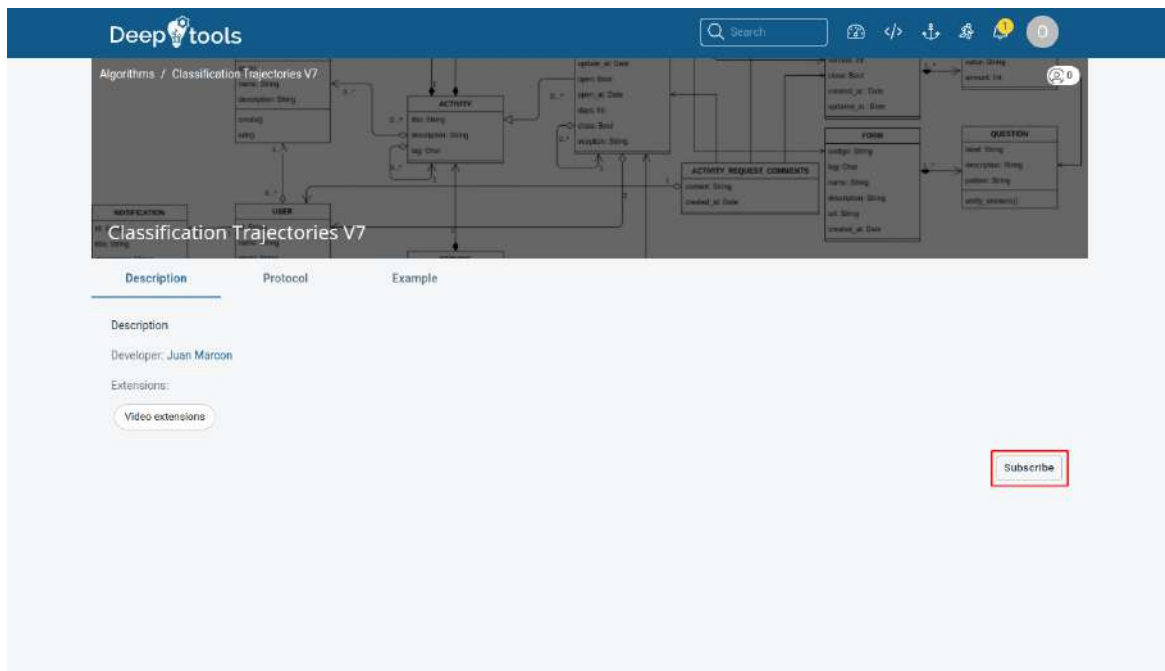


Figura 21. CU09 - Cancelar suscripción

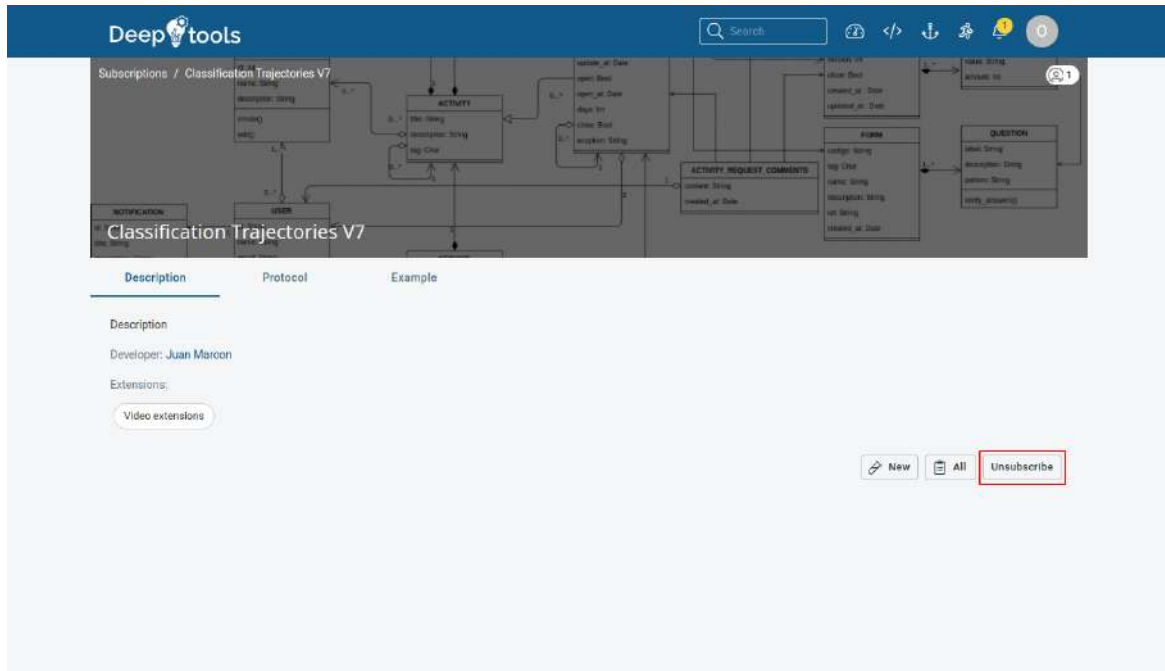


Figura 22. CU10 - Listar suscripciones

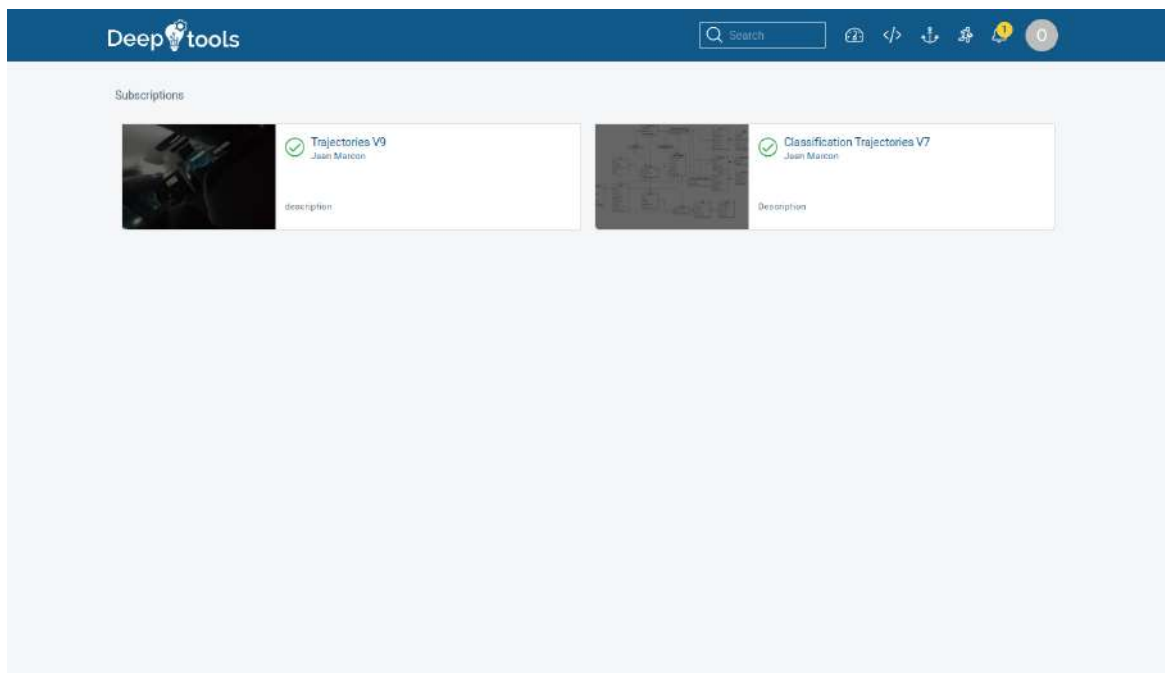


Figura 23. CU11 - Desplegar algoritmo | Paso 1

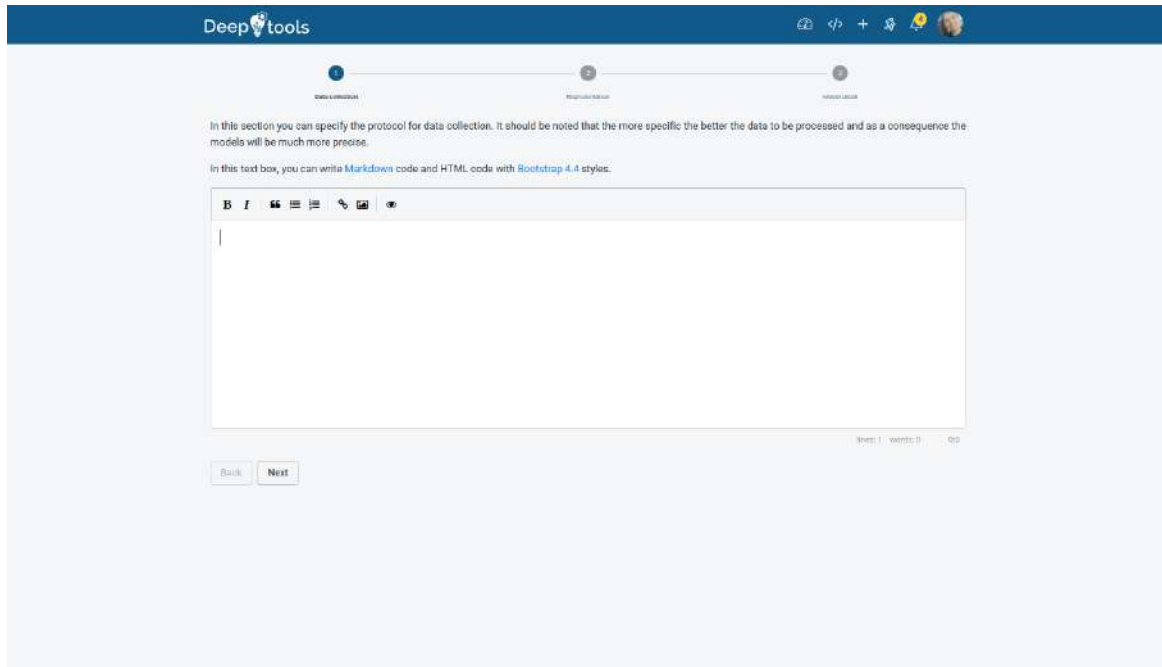


Figura 24. CU11 - Desplegar algoritmo | Paso 2

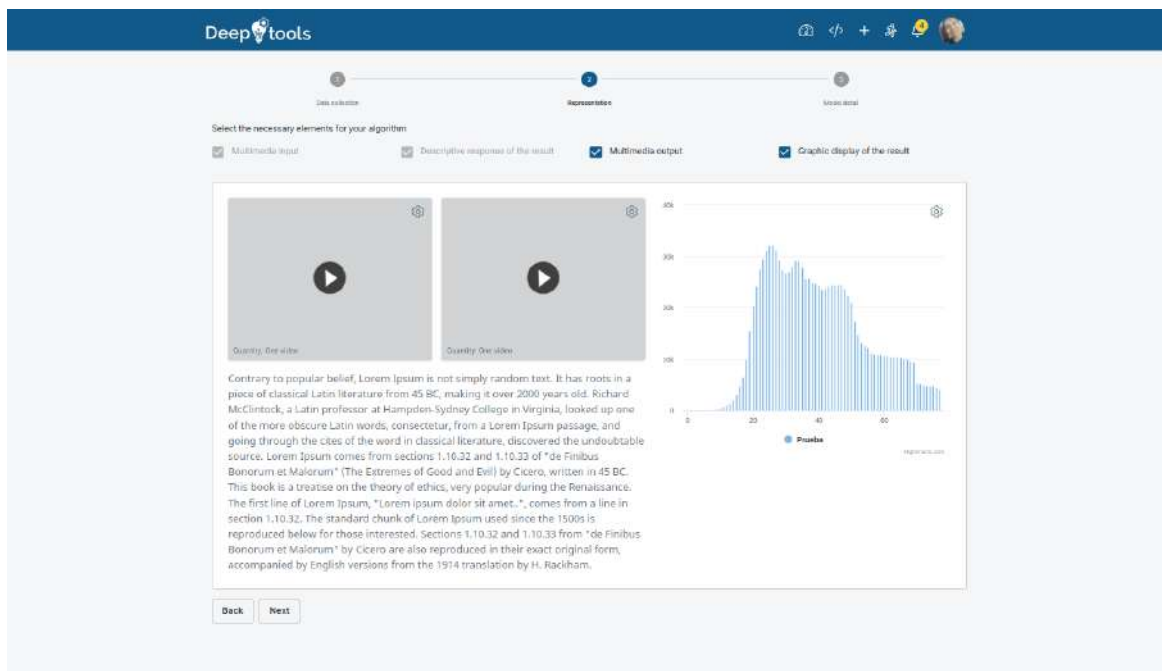


Figura 25. CU11 - Desplegar algoritmo | Paso 3

The screenshot shows the 'Microservice' step of the DeepTools deployment process. At the top, there are three numbered steps: 1. Data container, 2. Image repository, and 3. Microservice. The main heading is 'Help us with some necessary information to be able to communicate with your container.' Below this, there are several input fields: 'Images' (a dropdown menu), 'Path' (a text field), 'Filename' (a text field), 'Classname' (a text field), 'Extension' (a dropdown menu with 'Video extension' selected), 'Name' (a text field), and 'Description' (a larger text area). There are also 'Back' and 'Send' buttons at the bottom left.

Figura 26. CU11 - Desplegar algoritmo | Construyendo

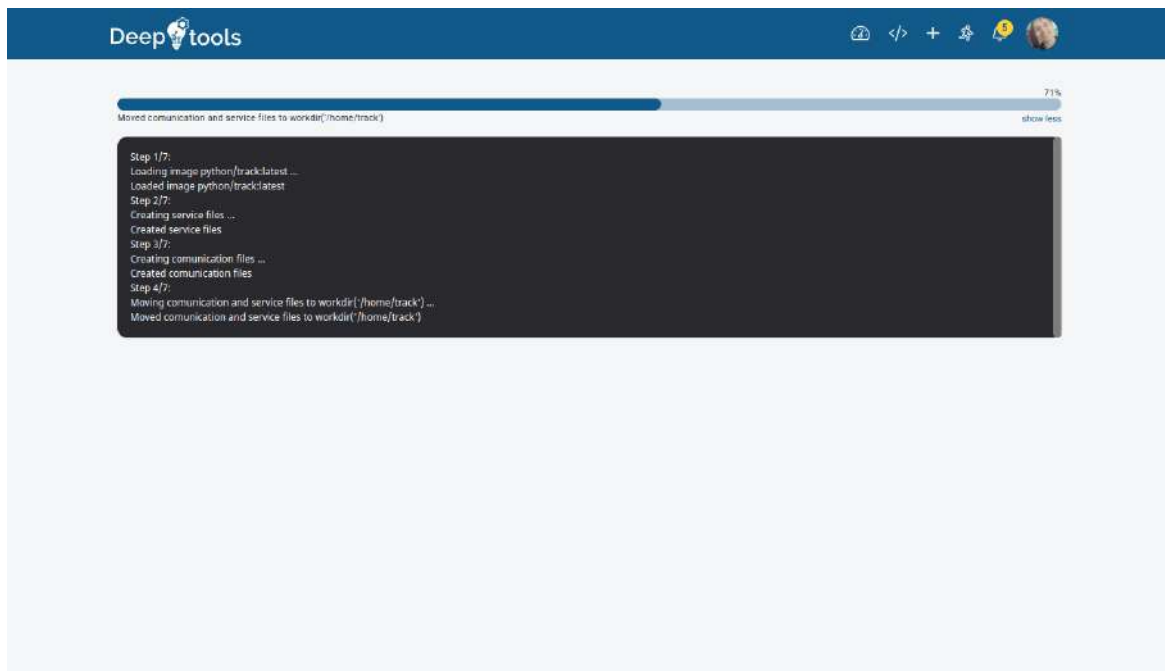


Figura 27. CU12 - Iniciar algoritmo

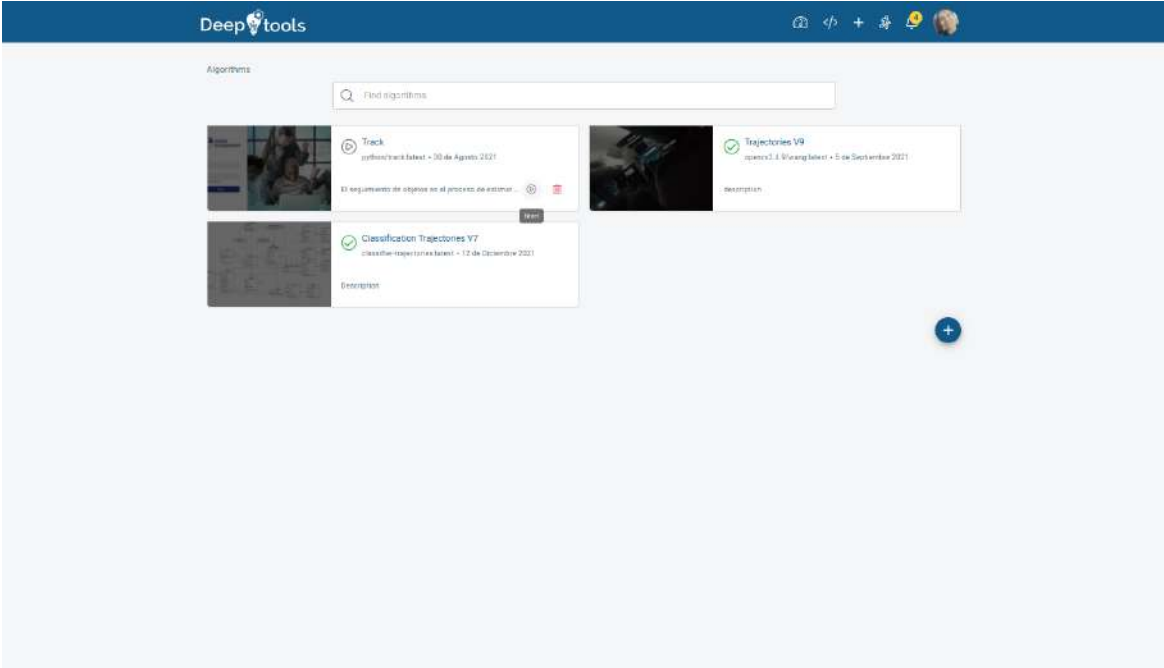


Figura 28. CU13 - Detener algoritmo

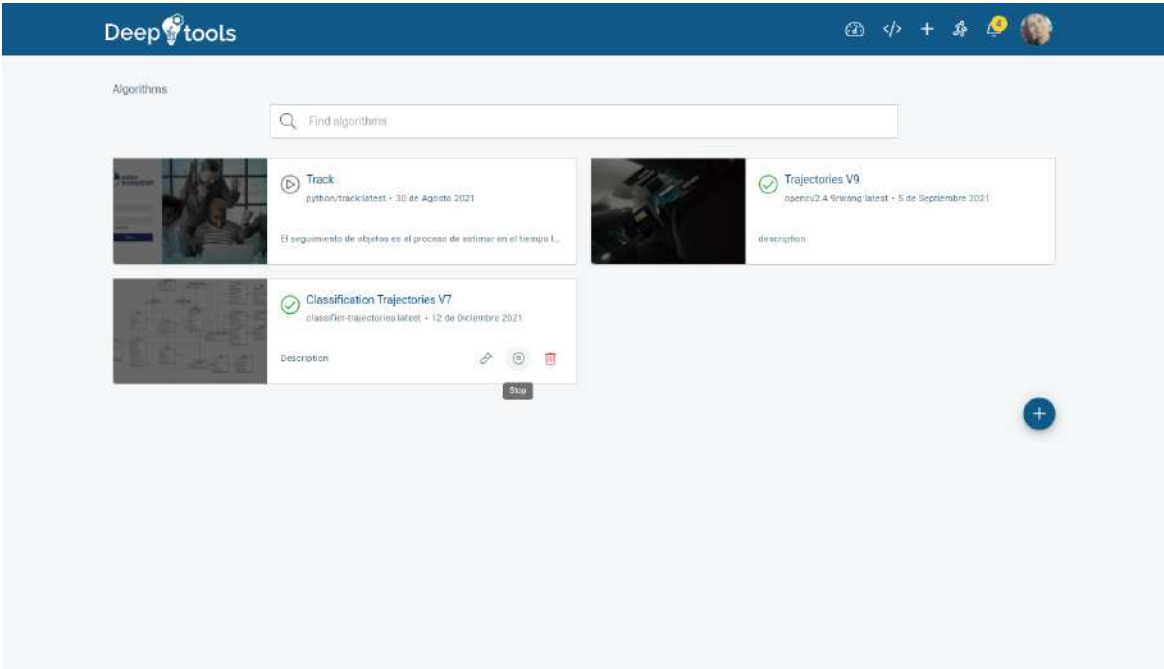


Figura 29. CU14 - Eliminar algoritmo | Selección del algoritmo

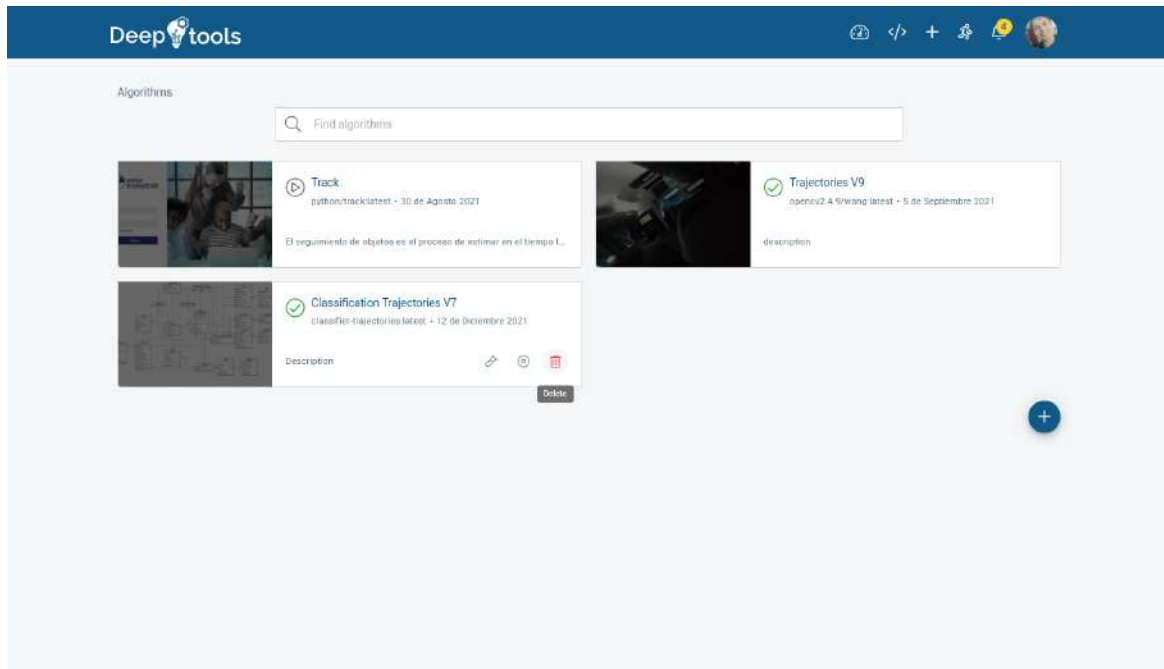


Figura 30. CU14 - Elimina algoritmo | Confirmar eliminación

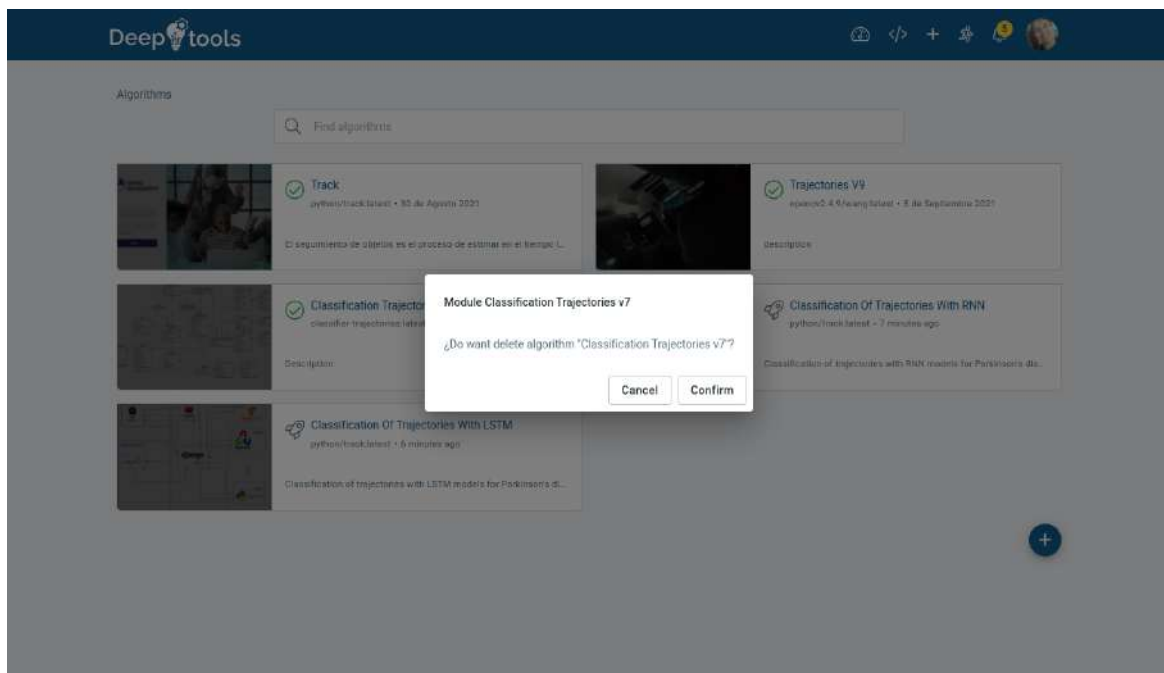


Figura 31. CU15 - Administrar usuarios

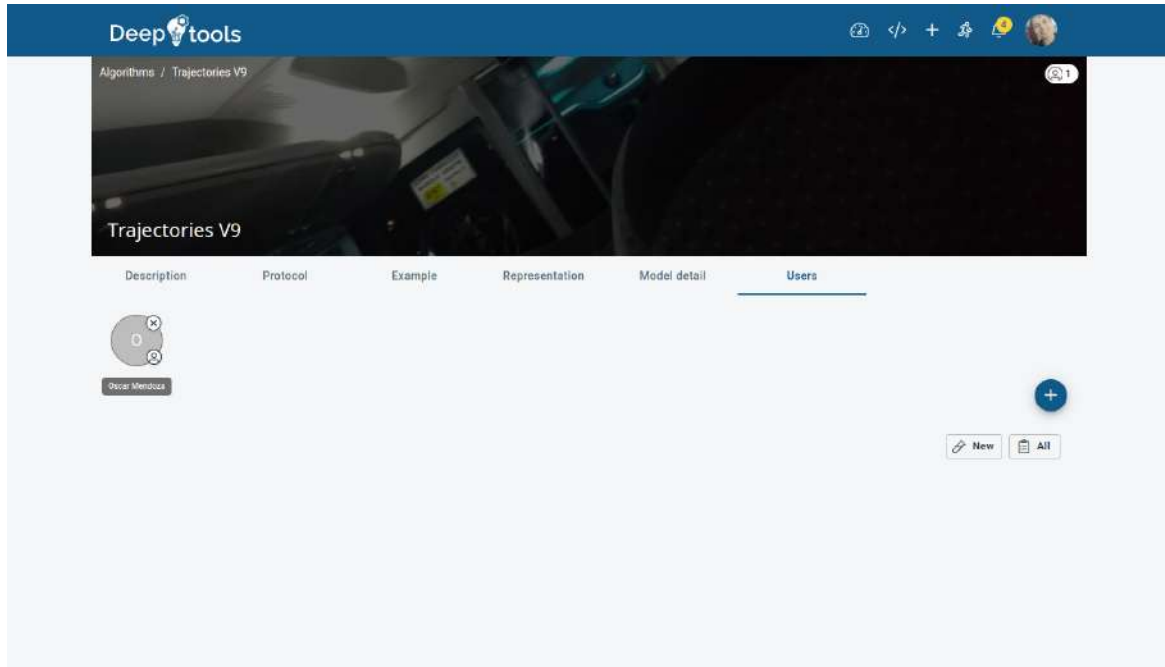


Figura 32. CU15 - Administrar usuarios | Agregar suscripción

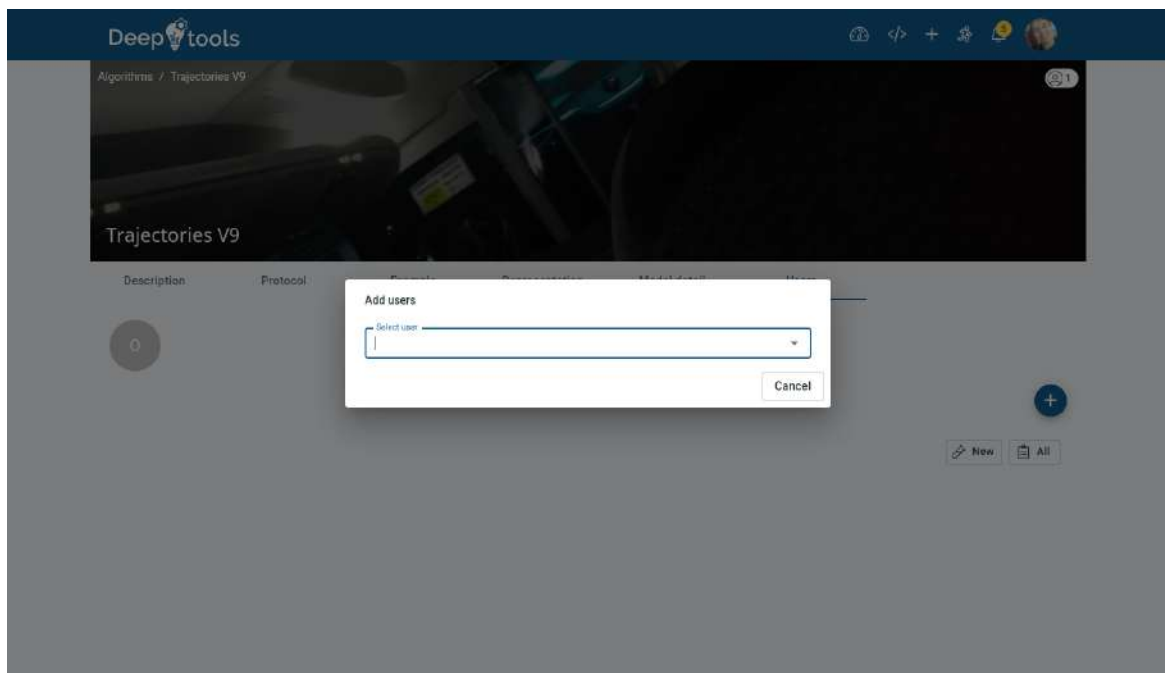


Figura 33. CU15 - Administrar usuarios | Eliminar suscripción

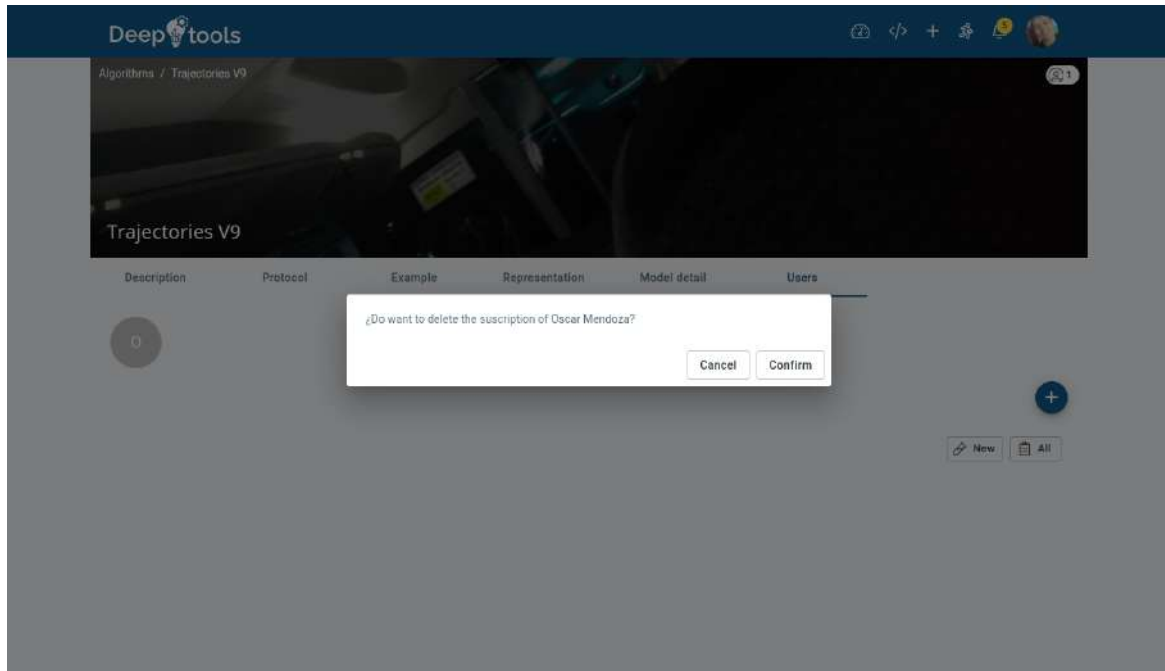


Figura 34. CU16 - Ejecutar algoritmo

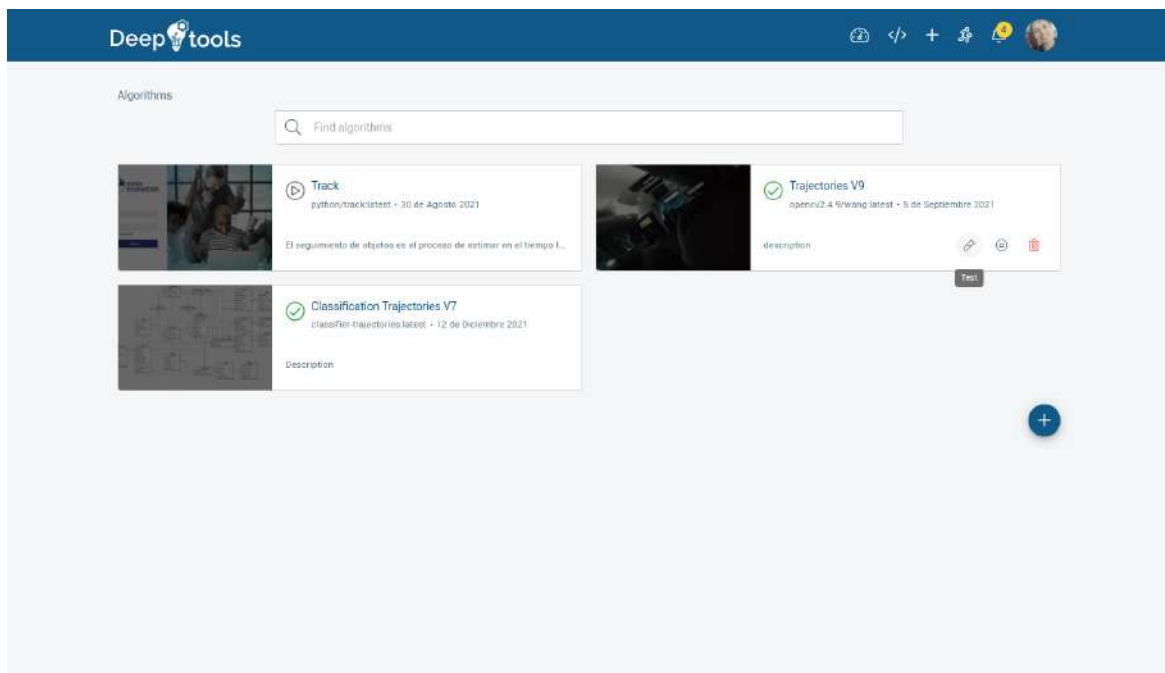


Figura 35. CU16 - Ejecutar algoritmo | Selección de archivos

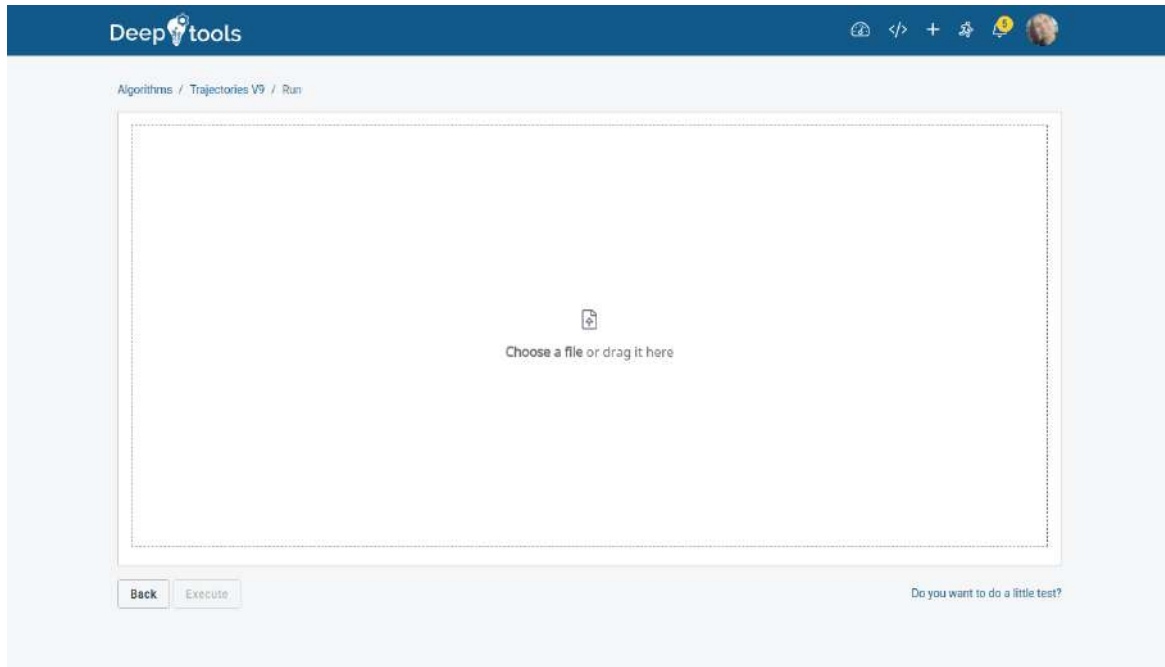


Figura 36. CU16 - Ejecutar algoritmo | Prueba con archivos probados por el desarrollador

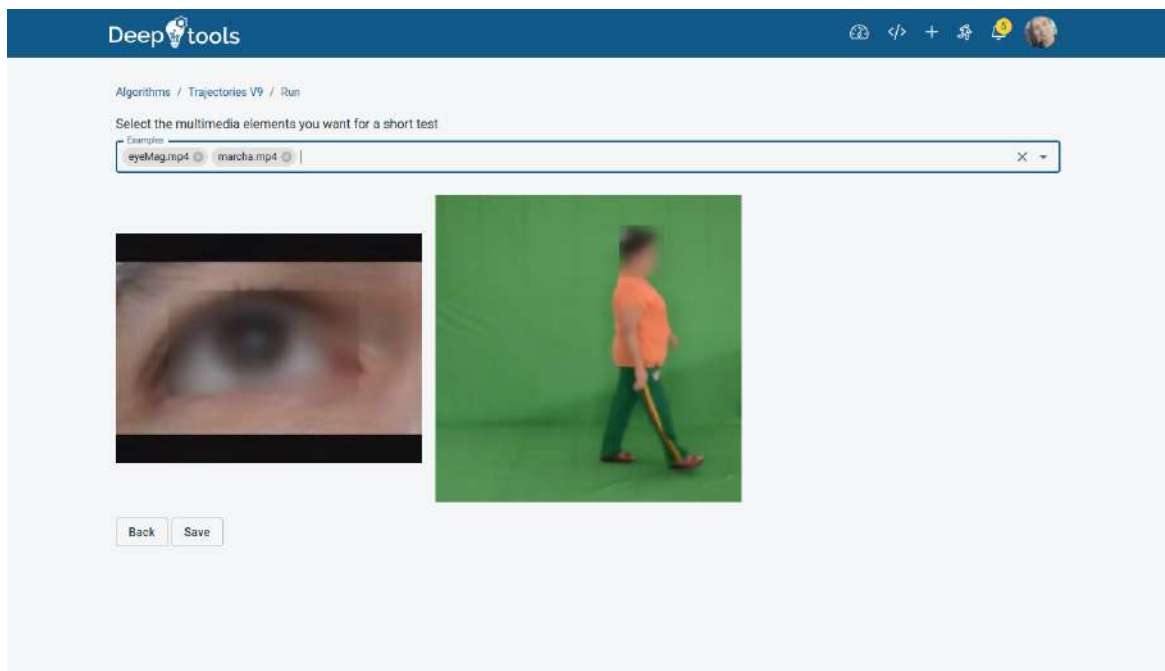


Figura 37. CU16 - Ejecutar algoritmo | Algoritmo en ejecución

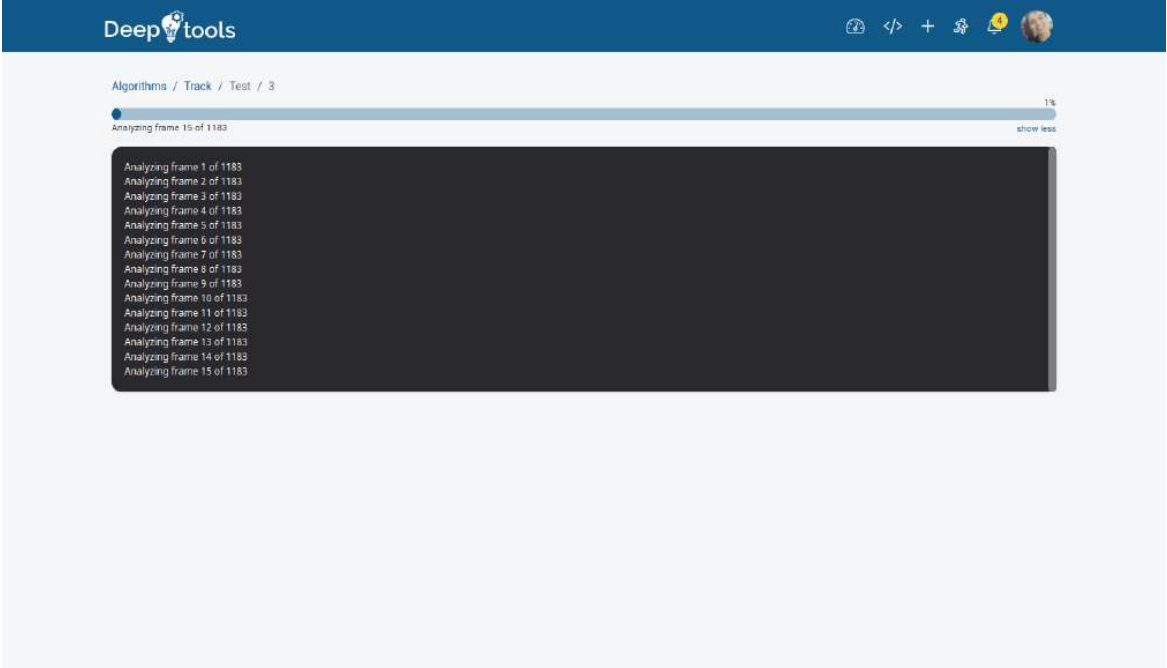


Figura 38. CU17 - Listar ejecuciones

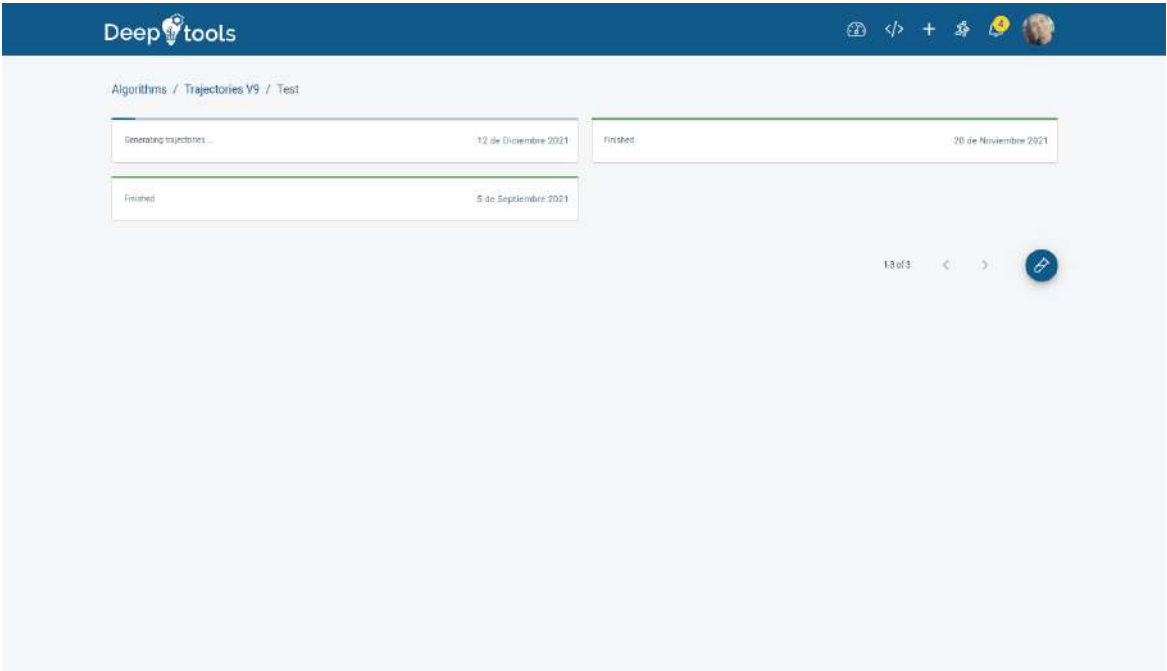


Figura 39. CU18 - Ver ejecución

The screenshot shows the DeepTools interface. At the top, there is a blue header with the 'DeepTools' logo and several utility icons. Below the header, the breadcrumb path 'Algorithms / Trajectories V9 / Test / 2' is visible. On the left side, there are two image thumbnails: the top one shows a close-up of a human eye, and the bottom one shows a heatmap of a brick wall with green and red points. To the right of these images is a table with the following data:

Name	Value
Dimensions	320x240
Frames	160
FPS	25.0
Trajectories	51211

Figura 40. CU19 - Eliminar ejecución | Selección de la ejecución

The screenshot shows the DeepTools interface with a list of execution jobs. The breadcrumb path is 'Algorithms / Trajectories V9 / Test'. There are three job cards visible:

- Job 1: 'Generating trajectories...' with a date of '12 de Diciembre 2021'.
- Job 2: 'Finished' with a date of '5 de Septiembre 2021'.
- Job 3: 'Finished' with a date of '20 de Noviembre 2021'. This job has a 'Delete test' button below it.

At the bottom right, there is a pagination control showing '13 of 3' and navigation arrows, along with a circular icon containing a pencil.

Figura 41. CU19 - Eliminar ejecución | Confirmar eliminación

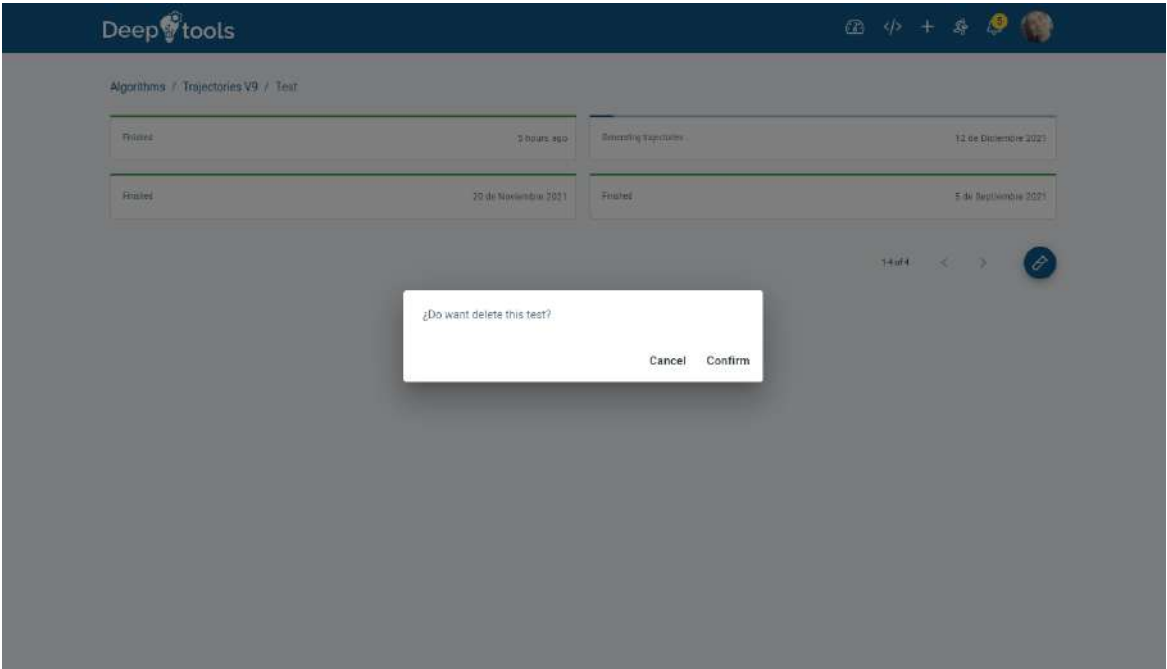


Figura 42. CU20 - Clonar ejecución | Selección de ejecución a clonar

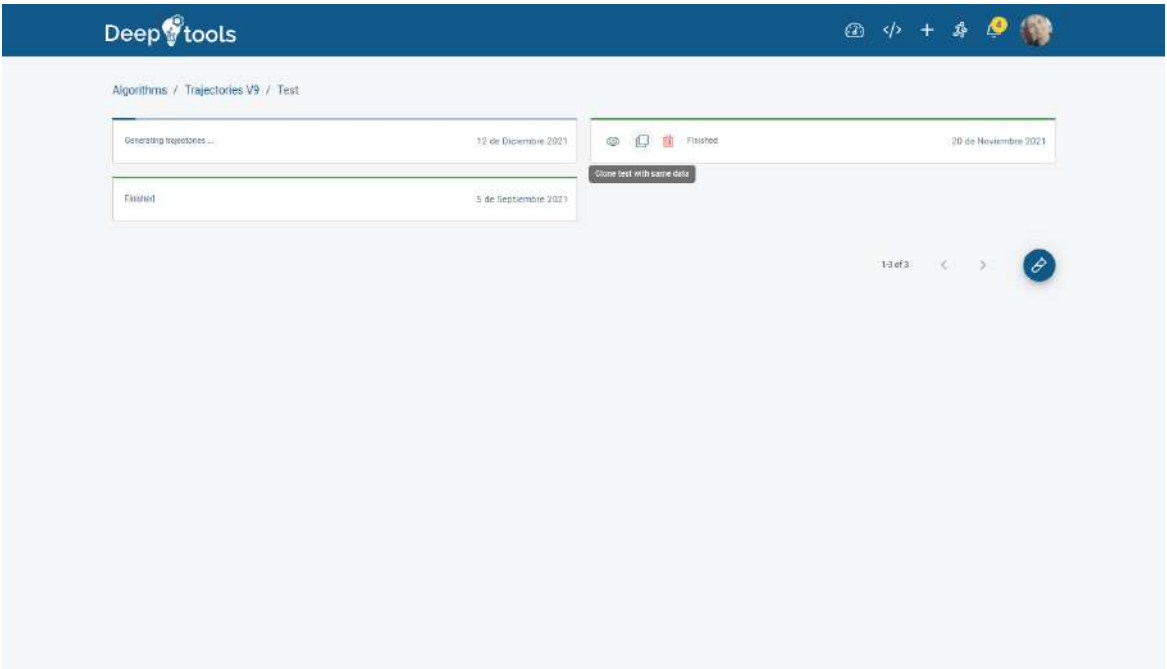


Figura 43. CU20 - Clonar ejecución | Selección de archivos

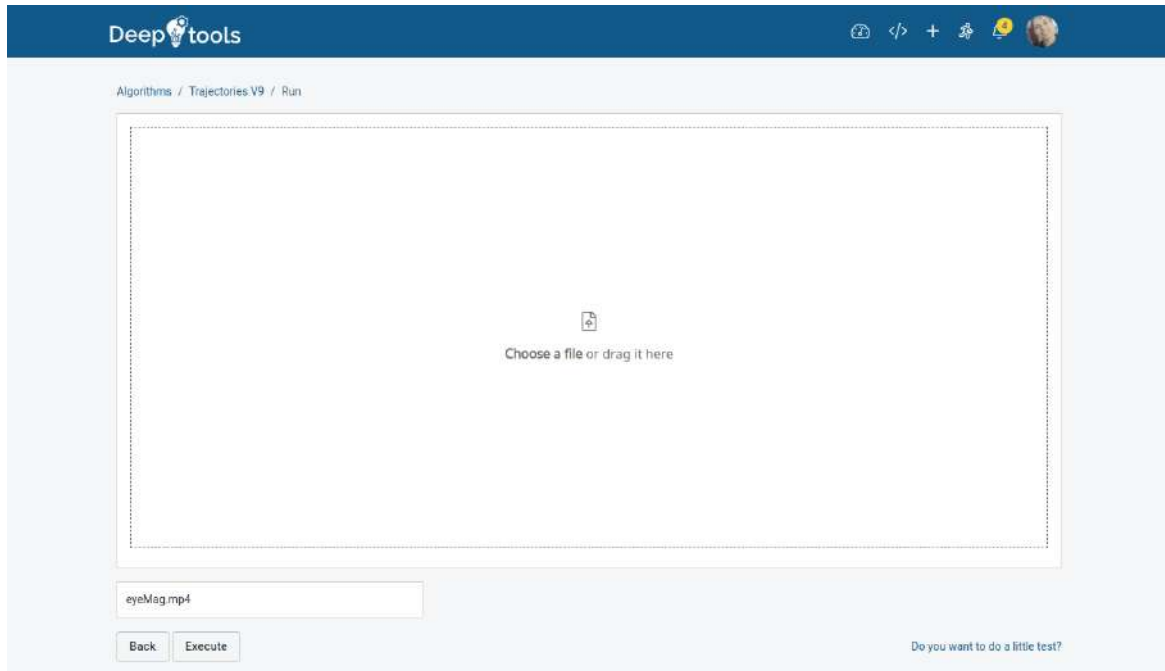


Figura 44. CU20 - Clonar ejecución | Algoritmo en ejecución de el experimento clonado

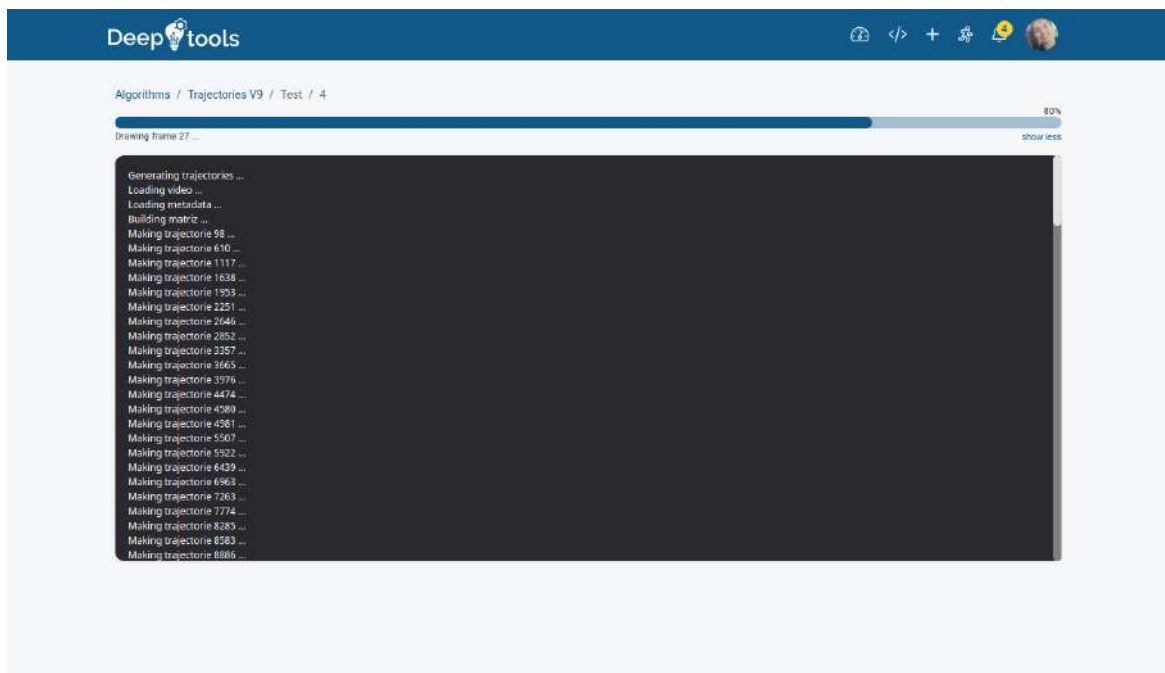


Figura 45. CU21 - Filtrar algoritmos

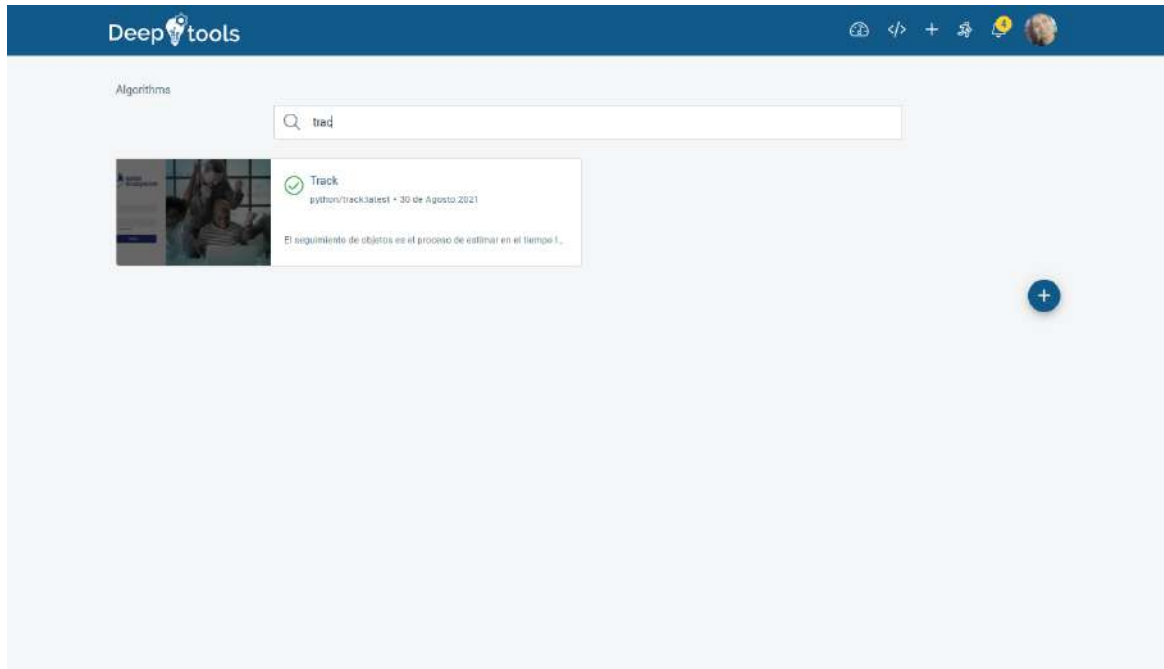


Figura 46. CU22 - Listar algoritmos

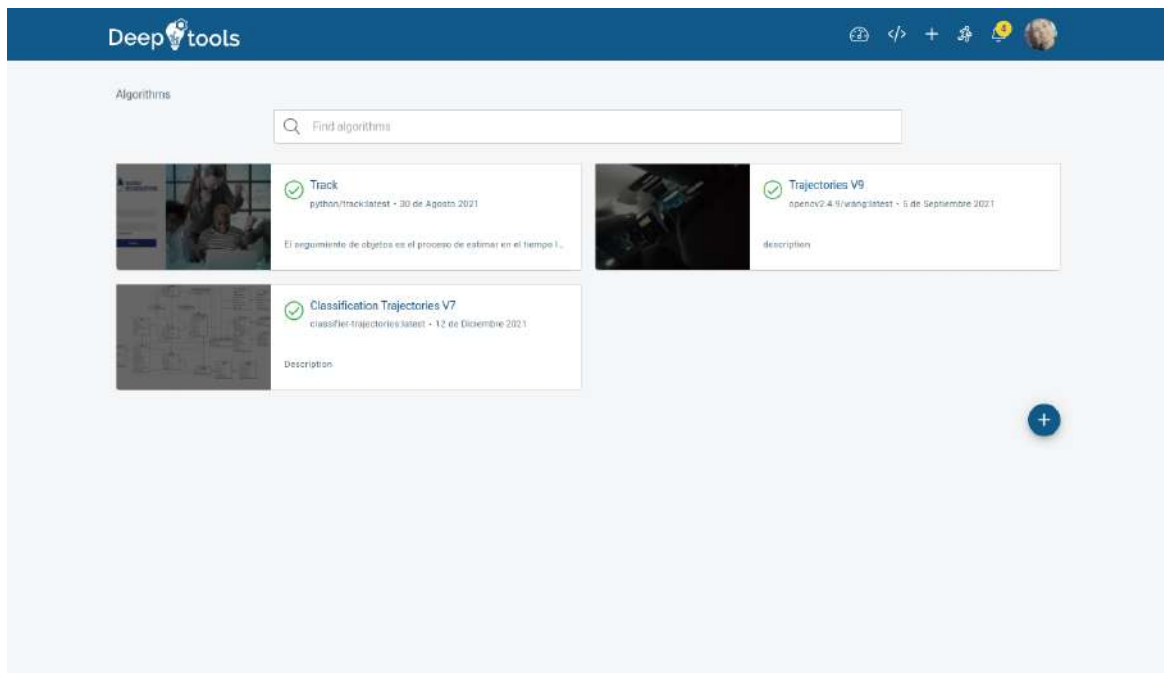


Figura 47. CU23 - Descargar elementos multimedia

The screenshot shows the DeepTools web interface. At the top, there is a blue header with the 'DeepTools' logo and several utility icons. Below the header, the breadcrumb path 'Algorithms / Trajectories V9 / Test / 1' is visible. The main content area is split into two parts. On the left, there is a video player showing a person walking on a green background. Below the video, a context menu is open, displaying options: 'Descargar' (Download), 'Velocidad de reproduc...' (Playback speed), and 'Pantalla en pantalla' (Fullscreen). On the right, a table displays the following metadata:

Name	Value
Dimensions	520x520
Frames	98
FPS	30.0
Trajectories	55094

Tabla 7. Caso de uso: CU00 - Iniciar sesión

Id	CU00
Nombre	Iniciar sesión
Actores	Usuario, Desarrollador
Descripción	El usuario podrá iniciar sesión en el software usando credenciales de acceso asignadas durante el registro.
Flujo	<ol style="list-style-type: none">1. El usuario ingresa a la página principal.2. El usuario ingresa sus credenciales y da clic en <i>Sign In now</i>.3. La pagina verifica y marca las credenciales como correctas.4. El usuario es dirigido al <i>Dashboard</i>.
Excepción	Las credenciales de ingreso no coinciden. En este caso se informara al usuario este problema.

Tabla 8. Caso de uso: CU01 - Cerrar sesión

Id	CU01
Nombre	Cerrar sesión
Actores	Usuario, Desarrollador
Descripción	El usuario podrá cerrar sesión o pasado 1 día de inactividad el software cerrará la sesión automáticamente
Flujo	<ol style="list-style-type: none">1. El usuario da click en la foto de perfil en la esquina superior derecha, esto despliega un menú con tres opciones.2. El usuario selecciona la opción <i>Logout</i>.3. El software cierra la sección.4. El usuario es dirigido a la pagina de inicio de sesión.

Tabla 9. Caso de uso: CU02 - Registrar usuario

Id	CU02
Nombre	Registrar usuario
Actores	Invitado
Descripción	El usuario podrá registrarse en el software con alguno de los dos roles disponibles (Usuario o desarrollador).
Flujo	<ol style="list-style-type: none"> 1. El usuario ingresa a la página principal. 2. El usuario da click sobre el enlace <i>Create account</i>. Esto redirige al usuario a la pagina de registro. 3. El usuario llena el formulario con su información personal. 4. El usuario da click sobre el botón <i>Create account</i>. 5. El usuario es dirigido al <i>Dashboard</i>.
Excepción 1	Si el correo electrónico ya se encuentra registrado se le informara al usuario.
Excepción 2	Si las contraseñas no coinciden no se podrá continuar con el registro. Se le informara al usuario.
Excepción 3	Si falta algún campo del formulario no se podrá continuar con el registro. Se le informara al usuario.

Tabla 10. Caso de uso: CU03 - Ver perfil

Id	CU03
Nombre	Ver perfil
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán ver la información de su cuenta, tales como Nombre, Foto de perfil y suscripciones.
Flujo	<ol style="list-style-type: none"> 1. El usuario da click en la foto de perfil en la esquina superior derecha, esto despliega un menú con tres opciones.

2. El usuario selecciona la opción *Profile*.
3. El usuario es redirigido a una pagina con la información de su cuenta.

Tabla 11. Caso de uso: CU04 - Editar perfil

Id	CU04
Nombre	Editar perfil
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán modificar la información de su cuenta, tales como Nombre, Apellido, Correo electrónico y Foto de perfil.
Flujo	<ol style="list-style-type: none"> 1. El usuario da click en la foto de perfil en la esquina superior derecha, esto despliega un menú con tres opciones. 2. El usuario selecciona la opción <i>Settings</i>. 3. El usuario es redirigido a una pagina con la un formulario para modificar la información de la cuenta. 4. El usuario da click en el botón <i>Upload profile</i>. Este es redirigido a una pagina con la información de su cuenta.
Excepción 1	Si el correo electrónico ya se encuentra registrado se le informara al usuario.
Excepción 2	Si falta algún campo del formulario no se podrá continuar con este. Se le informara al usuario.

Tabla 12. Caso de uso: CU05 - Ver perfil de desarrolladores

Id	CU05
Nombre	Ver perfil de desarrolladores
Actores	Usuario, Desarrollador

Descripción	Los usuarios podrán consultar información personal de los desarrolladores, tales como Nombre, Apellido, Foto de perfil y algoritmos activos.
Flujo	<ol style="list-style-type: none"> 1. El usuario rellena la caja de texto ubicada en la parte superior central. 2. Se despliega un menú con las coincidencias encontradas. 3. El usuario da click en una de estas. 4. El usuario es redirigido a una pagina con la información personal del desarrollador seleccionado.
Excepción 1	Si no se encuentran coincidencias no aparecerán perfiles seleccionables.
Excepción 2	Si la caja de texto no tiene una palabra con al menos 4 caracteres, no se realizara la búsqueda.

Tabla 13. Caso de uso: CU06 - Listar notificaciones

Id	CU06
Nombre	Listar notificaciones
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán listar las notificaciones del software.
Flujo	<ol style="list-style-type: none"> 1. El usuario da click en el icono de la campana ubicado en la esquina superior derecha. 2. Se despliega un menú con las ultimas 5 notificaciones activas y un botón para ver todas las notificaciones. 3. El usuario da click en el botón <i>Show all</i>. 4. El usuario es redirigido a una pagina con todas las notificaciones, organizadas por vistas y no vistas.
Excepción	Si el usuario no tiene notificaciones se mostrara un mensaje informándole que no tiene notificaciones.

Tabla 14. Caso de uso: CU07 - Notificar novedades

Id	CU07
Nombre	Notificar novedades
Actores	Usuario, Desarrollador
Descripción	El software debe notificarle a sus usuario la finalización de una ejecución, además de notificarle a los usuarios cuando un algoritmo no esta activo.
Flujo	<ol style="list-style-type: none">1. El estado de un algoritmo cambia o finaliza la ejecución de un algoritmo.2. Se notifica a los usuarios suscritos el cambio de estado o al usuario dueño que finalización de la ejecución según corresponda.

Tabla 15. Caso de uso: CU08 - Suscribirse

Id	CU08
Nombre	Suscribirse
Actores	Usuario, Desarrollador
Descripción	Todos los usuario pueden suscribirse a los algoritmos activos, al realizar esta acción permitirá a este usuario la posibilidad de ejecutar tantas veces quiera mientras este algoritmo permanezca activo. Además de esto, cada modificación en el algoritmo sera notificada a los usuarios suscritos.
Flujo	<ol style="list-style-type: none">1. El usuario lista los algoritmos.2. Selecciona un algoritmo de su interés.3. Da click en el botón 'Suscribirse'.

Tabla 16. Caso de uso: CU09 - Cancelar suscripción

Id	CU09
Nombre	Cancelar suscripción

Actores	Usuario, Desarrollador
Descripción	El software permite a los usuarios cancelar las suscripciones realizadas. Esto quitara el acceso a la ejecución de los algoritmos y evitara la envío de notificaciones relacionadas con este.
Flujo	<ol style="list-style-type: none"> 1. El usuario lista sus suscripciones (). 2. El usuario selecciona el algoritmo de interés. 3. Da click en el botón 'Cancelar suscripción'

Tabla 17. Caso de uso: CU10 - Listar suscripciones

Id	CU10
Nombre	Listar suscripciones
Actores	Usuario, Desarrollador
Descripción	El software permite a los usuarios ver una lista de todas las suscripciones. En otras palabras tiene una lista con los algoritmos que pueden usar.
Flujo	<ol style="list-style-type: none"> 1. El usuario da click en el icono que representa algoritmo ubicado en la esquina superior derecha.
Excepción	Si el usuario no tiene suscripciones se le mostrara un mensaje informándole.

Tabla 18. Caso de uso: CU11 - Desplegar algoritmo

Id	CU11
Nombre	Desplegar algoritmo
Actores	Desarrollador
Descripción	El software permite a los usuarios desarrolladores desplegar todos los algoritmos relacionados con IA que cumplan con los requisitos establecidos.

Flujo	<ol style="list-style-type: none"> 1. El usuario da click en el icono de agregar ubicado en la esquina superior derecha. 2. EL usuario debe especificar el protocolo de toma de datos, si es necesario, al finalizar esto deberá dar click en el botón 'Siguiente' para continuar con el siguiente paso. 3. El usuario debe seleccionar los elementos necesarios para el algoritmo. Al finalizar esto debe dar click en el botón 'Siguiente'. 4. El usuario debe especificar los metadatos necesarios para el despliegue, tales como imagen del docker, directorio de trabajo, archivo de comunicación, entre otros. 5. El usuario debe seleccionar los elementos necesarios para el algoritmo. Al finalizar esto debe dar click en el botón 'Siguiente'. 6. El software le mostrara en tiempo real una barra de progreso con mostrando paso a paso la contracción del algoritmo. El software le notificara cuando haya finalizado
Excepción 1	Si los metadatos necesarios para el despliegue no concuerda con la imagen docker se le informara al desarrollador.
Excepción 2	Si los procesos especificados en el archivo de comunicación falla, se detendrá el proceso y se registrara el error.

Tabla 19. Caso de uso: CU12 - Iniciar algoritmo

Id	CU12
Nombre	Iniciar algoritmo
Actores	Desarrollador

Descripción	Los algoritmos tienen distintos estados que representan la posibilidad de ser usados. Cuando se inicia un algoritmo, cambia su estado de desplegado a activo, esto significa que podrá ser usado por los usuarios suscritos.
Flujo	<ol style="list-style-type: none"> 1. El desarrollador lista sus algoritmos. 2. Da click en el botón 'Iniciar'.

Tabla 20. Caso de uso: CU13 - Detener algoritmo

Id	CU13
Nombre	Detener algoritmo
Actores	Desarrollador
Descripción	Al detener un algoritmo cambiamos su estado de activo a desplegado, cambiando la posibilidad de ser ejecutado por los usuarios suscritos y eliminando su visibilidad pública.
Flujo	<ol style="list-style-type: none"> 1. El desarrollador lista sus algoritmos. 2. Da click en el botón 'Detener'.
Excepción	Si existen algoritmos en ejecución el algoritmo no se detendrá hasta finalizar su cola de tareas. No se podrán más ejecuciones.

Tabla 21. Caso de uso: CU14 - Eliminar algoritmo

Id	CU14
Nombre	Eliminar algoritmo
Actores	Desarrollador
Descripción	La eliminación de algoritmo imposibilita el uso de este además de su visibilidad para todos los usuarios incluyendo su creador.
Flujo	<ol style="list-style-type: none"> 1. El desarrollador lista sus algoritmos.

2. Da click en el botón 'Eliminar'.
3. El software pide confirmación para esta acción.

Excepción	Si existen algoritmos en ejecución el algoritmo no se detendrá hasta finalizar su cola de tareas. Una vez finalizada se eliminara el algoritmo.
------------------	---

Tabla 22. Caso de uso: CU15 - Administrar usuarios

Id	CU15
Nombre	Administrar usuarios
Actores	Desarrollador
Descripción	Los desarrolladores podrán habilitar y deshabilitar suscripciones a usuarios.
Flujo	<ol style="list-style-type: none"> 1. El desarrollador lista sus algoritmos. 2. Selecciona el algoritmo de interés. 3. Selecciona la pestaña usuarios.

Tabla 23. Caso de uso: CU16 - Ejecutar algoritmo

Id	CU16
Nombre	Ejecutar algoritmo
Actores	Usuario, Desarrollador
Descripción	Todos los usuarios podrán ejecutar un algoritmo siempre y cuando sean dueños o este suscritos a este.
Flujo	<ol style="list-style-type: none"> 1. El usuario lista sus algoritmos. 2. Selecciona el algoritmo de interés. 3. Da click en el botón 'Ejecutar'. 4. Se le mostrar el protocolo para la toma de datos. Da click en el botón 'Siguiente'.

5. Sube los archivos necesarios para la ejecución. Da click en el botón 'Siguiente'
6. Se notifica al usuario cuando haya terminado el proceso.

Excepción 1	Si los archivos no concuerda con los especificados no podrá proceder con el proceso.
Excepción 2	Hasta no finalizar la subida de archivos no se podrá proceder con el proceso.
Excepción 3	Si falla algún proceso en la ejecución se le notificara al usuario.

Tabla 24. Caso de uso: CU17 - Listar ejecuciones

Id	CU17
Nombre	Listar ejecuciones
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán listar ejecuciones de su autoría.
Flujo	<ol style="list-style-type: none"> 1. El usuario lista sus algoritmos. 2. Selecciona el algoritmo de interés. 3. Da click en el botón 'Ver ejecuciones'.
Excepción	Si no tiene ejecuciones se le informara al usuario.

Tabla 25. Caso de uso: CU18 - Ver ejecución

Id	CU18
Nombre	Ver ejecución
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán ver los detalles de ejecuciones de su autoría.
Flujo	<ol style="list-style-type: none"> 1. El usuario lista sus algoritmos. 2. Selecciona el algoritmo de interés.

3. Da click en el botón 'Ver ejecuciones'.
4. Selecciona la ejecución de su interés.

Tabla 26. Caso de uso: CU19 - Eliminar ejecución

Id	CU19
Nombre	Eliminar ejecución
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán eliminar las ejecuciones de su autoría.
Flujo	<ol style="list-style-type: none"> 1. El usuario lista sus algoritmos. 2. Selecciona el algoritmo de interés. 3. Da click en el botón 'Ver ejecuciones'. 4. Selecciona la ejecución de su interés. 5. Da click en el botón 'Eliminar'. 6. El software pide confirmación para esta acción.

Tabla 27. Caso de uso: CU20 - Clonar ejecución

Id	CU20
Nombre	Clonar ejecución
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán crear ejecuciones similares a las ya creadas.
Flujo	<ol style="list-style-type: none"> 1. El usuario lista sus algoritmos. 2. Selecciona el algoritmo de interés. 3. Da click en el botón 'Ver ejecuciones'. 4. Selecciona la ejecución de su interés. 5. Da click en el botón 'Clonar'.

Tabla 28. Caso de uso: CU21 - Filtrar algoritmos

Id	CU21
Nombre	Filtrar algoritmos
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán buscar algoritmos mediante palabras claves, autor, entre otros.
Flujo	<ol style="list-style-type: none">1. El usuario lista los algoritmos.2. Llena la caja de texto con las palabras claves.
Excepción	Si ningún algoritmo coincide con las palabras clave se informara al usuario.

Tabla 29. Caso de uso: CU22 - Listar algoritmos

Id	CU22
Nombre	Listar algoritmos
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán listar los algoritmos activos.
Flujo	<ol style="list-style-type: none">1. El usuario da click en el botón 'Algoritmos'.

Tabla 30. Caso de uso: CU23 - Descargar elementos multimedia

Id	CU23
Nombre	Descargar elementos multimedia
Actores	Usuario, Desarrollador
Descripción	Los usuarios podrán descargar los elementos multimedia de las ejecuciones de su autoría.
Flujo	<ol style="list-style-type: none">1. El usuario lista sus algoritmos.2. Selecciona el algoritmo de interés.

3. Da click en el botón 'Ver ejecuciones'.
 4. Selecciona la ejecución de su interés.
 5. Da click en el botón 'Descargar'.
-