cedia

research
lab
by cedia

Redacción y diseño de los casos de uso de las herramientas HPC, Indata, GitHub y Mattermost

CASO DE USO 1

cedia

“Conectando ideas, transformando sociedades”

cedia

Almacena y comparte datos de investigación.

Usa computación de alto rendimiento.

Colabora en proyectos de investigación de manera efectiva.

Envía archivos grandes de forma segura.

@CediaEc

cedia.edu.ec

Mejorará la reproducibilidad y eficiencia en tu investigación

Una Plataforma que revolucionará
la Investigación Científica en Ecuador

research
lab
by cedia

Redacción y diseño de los casos
de uso de las herramientas HPC,
Indata, GitHub y Mattermost

CASO DE USO 1

CONTENIDO

CASO DE USO 1

CASO DE USO 1

Modelo de aprendizaje automático para la detección de ransomware.

1.1 INTRODUCCIÓN

El ransomware es un tipo de malware que puede presentarse como un archivo ejecutable (.exe), troyano, o a través de URLs maliciosas y técnicas como el phishing. Este software malicioso encripta los datos de la víctima y exige un rescate para su recuperación, lo que lo convierte en una seria amenaza, ya que puede extorsionar a usuarios, interrumpir operaciones y comprometer el acceso a información personal o empresarial. Para enfrentar este desafío en el ámbito de la seguridad informática, se recomienda el uso de técnicas de aprendizaje automático. Estas técnicas permiten analizar patrones de comportamiento, identificar firmas distintivas y anticipar tácticas futuras, proporcionando una solución efectiva para la detección del ransomware.

Para abordar esta problemática, se propone investigar un modelo de entrenamiento o algoritmo que maximice la efectividad en la detección de ataques de ransomware utilizando algoritmos de aprendizaje automático. El proceso incluirá la evaluación de la precisión de los modelos, la comparación entre ellos, la identificación de archivos benignos y maliciosos. Todo esto se llevará a cabo sobre un conjunto de datos específico de ataques de ransomware.

1.2

USO DE LA HERRAMIENTA HPC

HPC

Antes de iniciar la investigación, se realiza una revisión sistemática de la literatura, centrada en la selección de fuentes primarias y secundarias. El objetivo es identificar los métodos y técnicas de Machine Learning más efectivos para la detección y mitigación de ataques de ransomware, así como determinar las características esenciales para su identificación y los atributos más eficaces para este propósito. Con los datos recopilados, se puede seleccionar la herramienta HPC para acceder a Rstudio Server y generar gráficos que reflejen los resultados obtenidos en la revisión sistemática de la literatura.

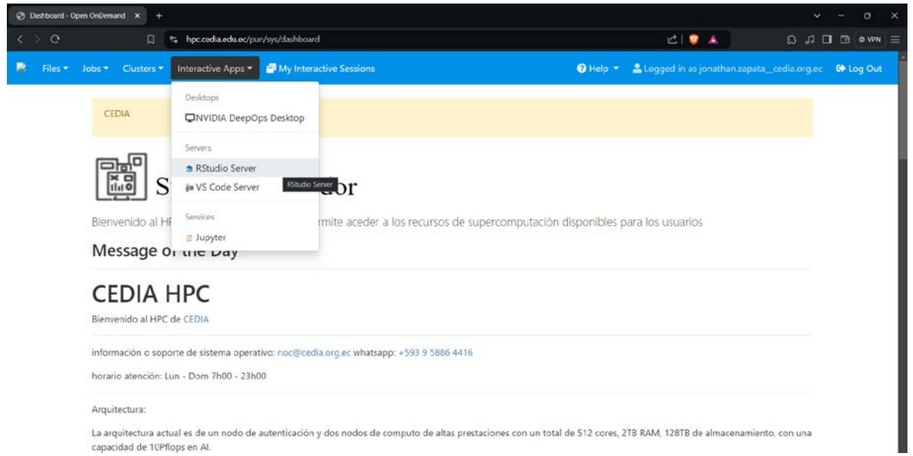


figura 1

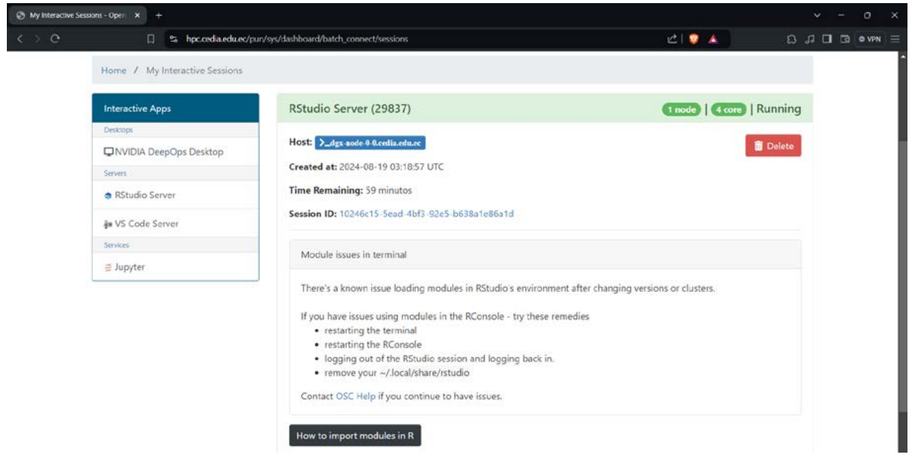


figura 2

En este caso, gracias a la revisión de la literatura, que incluyó el análisis de varios artículos de investigación en distintas base de datos de journals como World Wide Science, Scopus o Sciecne Direct, se identificaron los algoritmos de entrenamiento más utilizados y su frecuencia de uso:

- **Random Forest**: 26 veces
- **SVM**: 15 veces
- **Others**: 16 veces
- **XGBoost**: 5 veces
- **AdaBoost**: 8 veces
- **Gradient Boosting**: 7 veces
- **KNN**: 7 veces
- **Decision Tree**: 15 veces
- **Logistic Regression**: 6 veces
- **Naïve Bayes**: 7 veces
- **Neural Networks**: 2 veces
- **Long Short-term Memory**: 6 veces.

Por ende, vamos a ejecutar el un nuevo Script que nos ayude a representar y graficar de mejor manera estos datos.

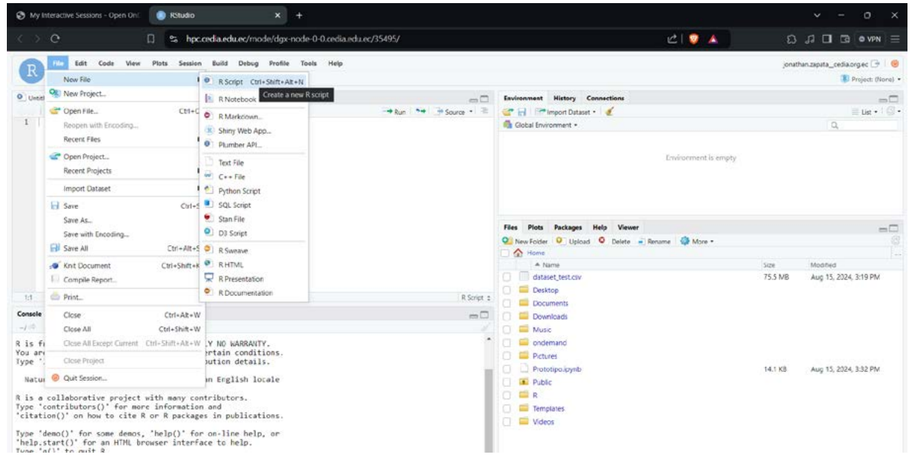


figura 3

Se ejecuto el siguiente Script en R

```
# Install and load the ggplot2 library
install.packages ("ggplot2")
library(ggplot2)

# Manual data
Algorithm <- c("Random Forest", "SVM", "Others", "XGBoost", "AdaBoost",
               "Gradient Boosting", "KNN", "Decision Tree", "Logistic Regression",
               "Naïve Bayes", "Neural Networks", "Long Short-term Memory")
values <- c(26, 15, 16, 5, 8, 7, 7, 15, 6, 7, 2, 6)

# Create a data frame
data <- data.frame(Algorithm, values)

# Calculate percentages
data$percentage <- round((data$values / sum(data$values)) * 100, 1)

# Create the scatter plot with percentages
ggplot(data, aes(x = Algorithm, y = percentage)) +
  geom_point(color = "blue", size = 3) +
  geom_text(aes(label = paste0(percentage, "%")), vjust = -0.5, size = 3.5) +
  labs(
       x = "Algorithm",
       y = "Percentage") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12), # Increased size
    axis.text.y = element_text(size = 8),
    axis.title.x = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 12, hjust = 0.5)
```

Y tenemos como resultado el siguiente grafico

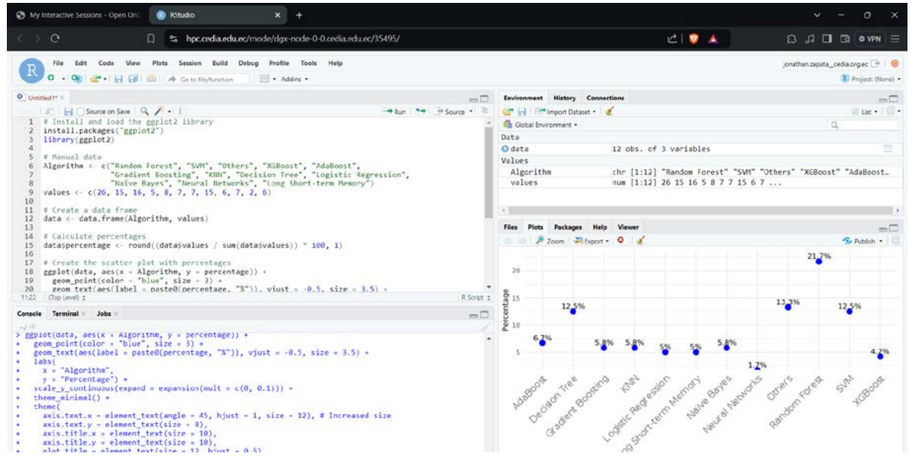


figura 4

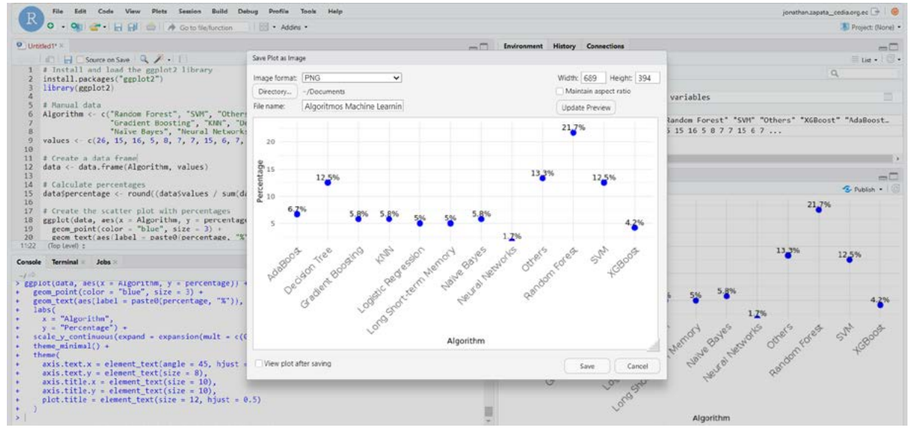


figura 5

Otro ejemplo relevante es la identificación de las técnicas o modelos con mayor frecuencia de uso en la detección de ransomware. Los resultados de la revisión sistemática fueron:

- **Dynamic Analysis**: 12
- **Static Analysis**: 18
- **Hybrid Analysis**: 1
- **Network Traffic**: 5
- **Digital DNA Sequencing**: 1
- **Supervised Learning**: 1

Se ejecutó un nuevo script para mejorar el diseño y la interpretación de los datos. Los resultados obtenidos fueron los siguientes:

```
# Instalar y cargar la librería ggplot2
install.packages("ggplot2")
library(ggplot2)

# Crear un data frame con los datos
datos <- data.frame(
  Techniques = c("Dynamic Analysis", "Static Analysis", "Hybrid Analysis", "Network
Traffic", "Digital DNA Sequencing", "Supervised Learning"),
  Quantity = c( 12, 18, 1, 5, 1, 1)
)

# Crear el gráfico de barras con colores Springer y labels en blanco
ggplot(datos, aes(x = Techniques, y = Quantity, fill = Techniques)) +
  geom_bar(stat = "identity") +

  scale_fill_manual(values = c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442",
"#0072B2", "#D55E00")) + # Colores Springer
  scale_y_continuous(limits = c(0, max(datos$Quantity) + 5)) + # Aumentar la escala del
eje Y
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),  # Eliminar el texto del eje X
    axis.title.x = element_blank(), # Eliminar la etiqueta del eje X
    axis.text.y = element_text(size = 12), # Aumentar el tamaño del texto del eje Y
    axis.title.y = element_blank(), # Eliminar la etiqueta del eje Y
    plot.title = element_text(size = 14, hjust = 0.5)  # Centrar y aumentar el tamaño del
título del gráfico
  ) +
  labs()
```

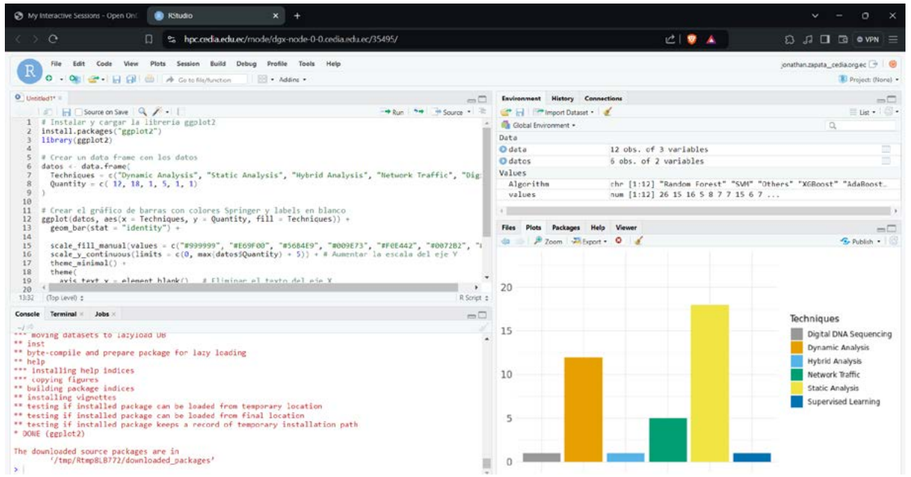


figura 6

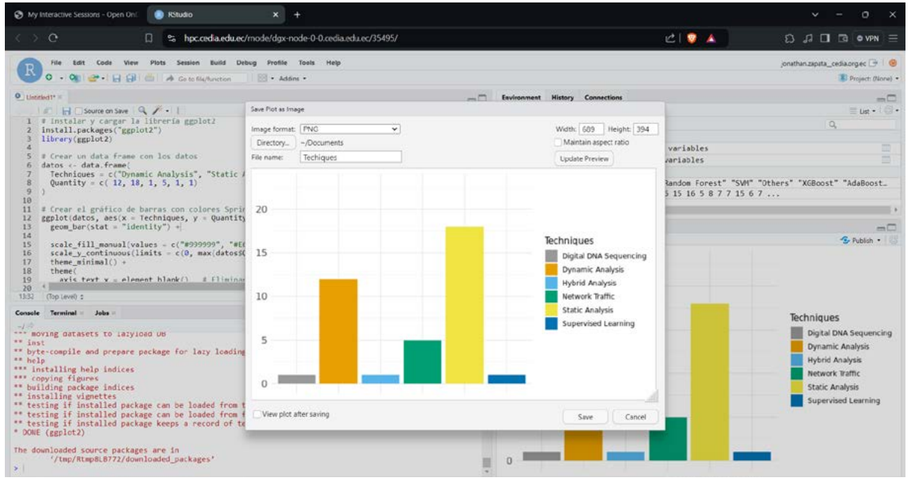


figura 7

Con los resultados obtenidos de la revisión de la literatura, ahora contamos con una comprensión clara de los procesos, técnicas y algoritmos más adecuados para la detección de ransomware.

A continuación, procederemos con la codificación y el entrenamiento de los algoritmos. Para ello, es crucial emplear herramientas adecuadas, como **HPC (High Performance Computing)**, y utilizar los servicios de **Jupyter**. Dado que Python es esencial en el desarrollo de algoritmos y procesos de machine learning, su uso será fundamental en este proyecto.

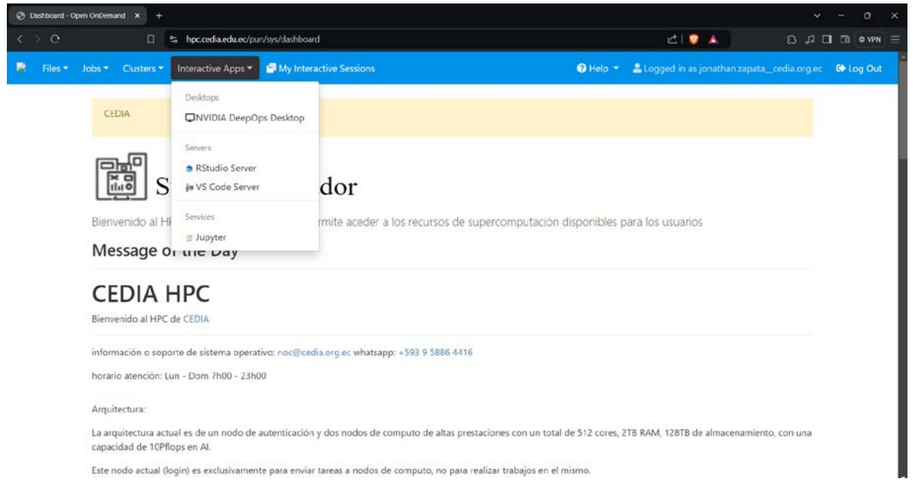


figura 8

Configuramos las opciones al momento de crear Jupyter en este caso se va a procesar mas de 200.000 datos por ese motivo se realiza las configuraciones

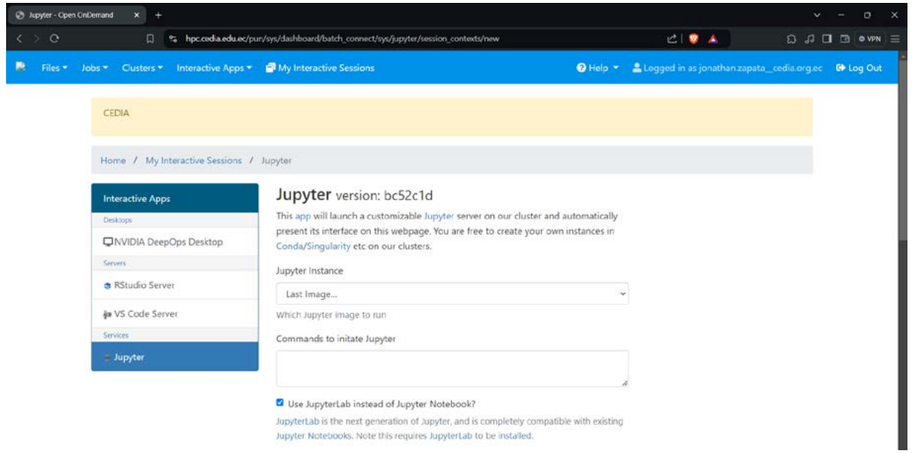


figura 9

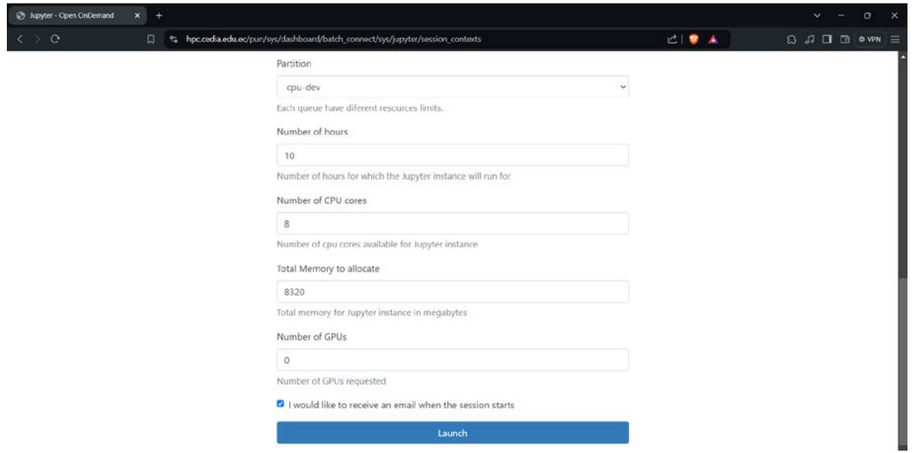


figura 10

1.3

USO DE LA HERRAMIENTA INDATA

Dataverse

Ya completado el proceso, vamos a realizar el ciclo de vida de Machine learning, que consiste en adquisición, preprocesamiento de datos, entrenamiento del modelo y evaluación:

Adquisición de datos

En este caso, utilizaremos un dataset que contiene características de archivos PE (Portable Executable) para el entrenamiento del modelo. El dataset incluye aproximadamente 52 características.

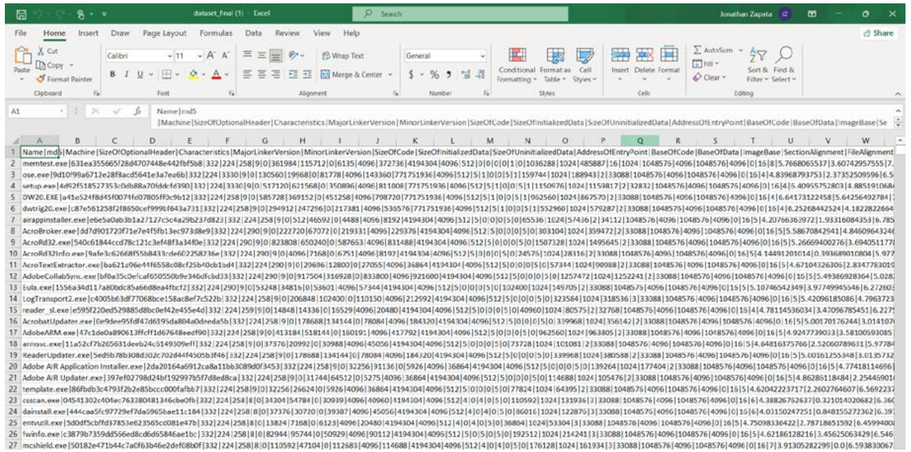


figura 11

Incluimos los resultados dentro del dataset en la aplicación InData para que otros investigadores puedan acceder a ellos de modo que puedan validar y comprobar la eficiencia el modelo de entrenamiento de aprendizaje automático.

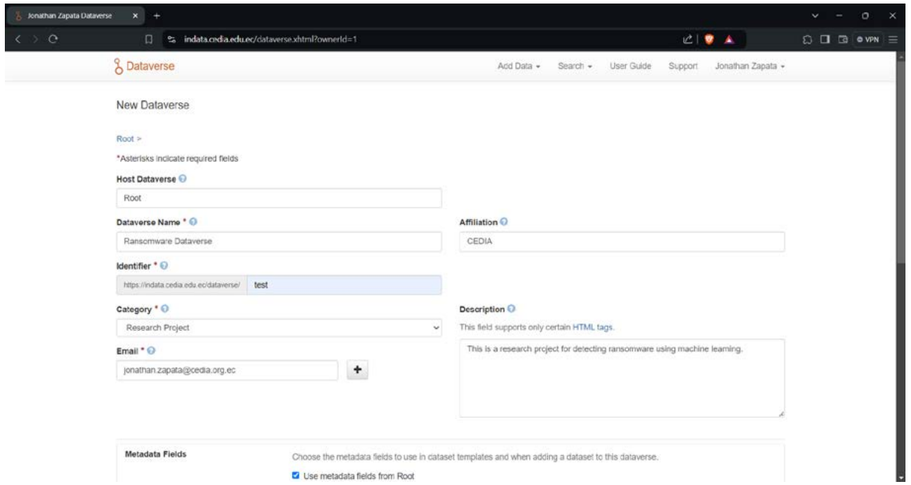


figura 12

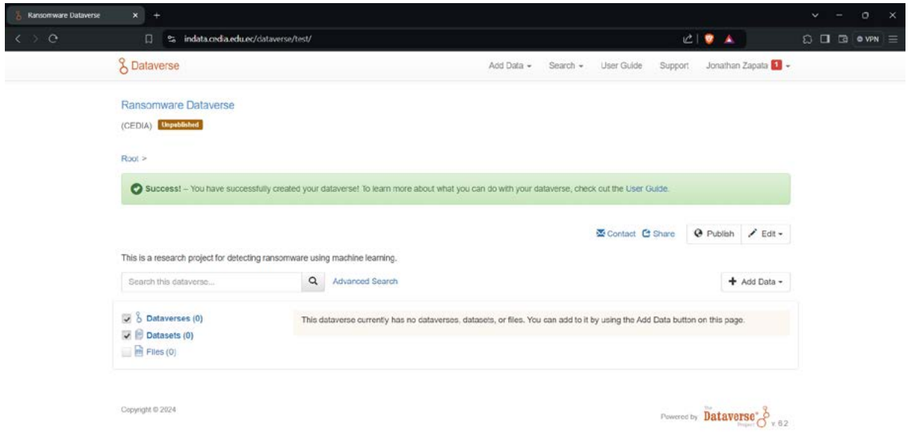


figura 13

Editamos los accesos, para que mas investigadores puedan acceder a nuestro dataset

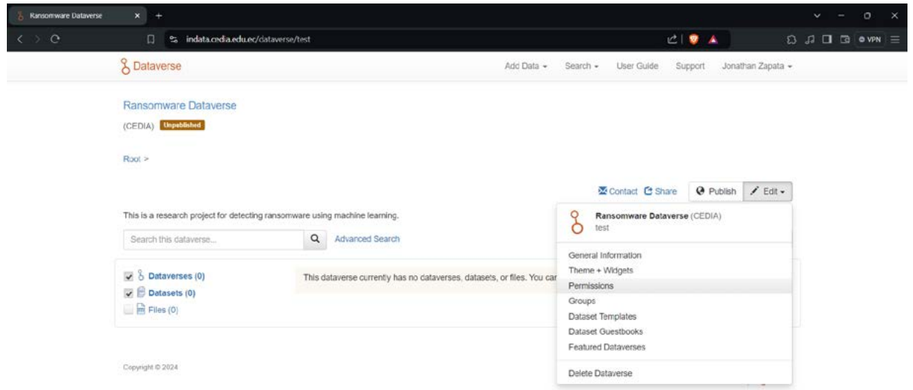


figura 14

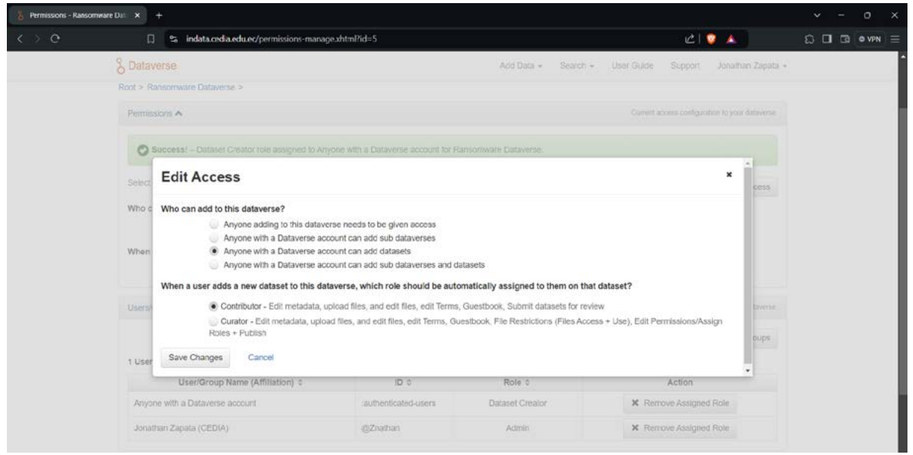


figura 15

Y finalmente publicamos nuestro repositorio de datos en dataverse

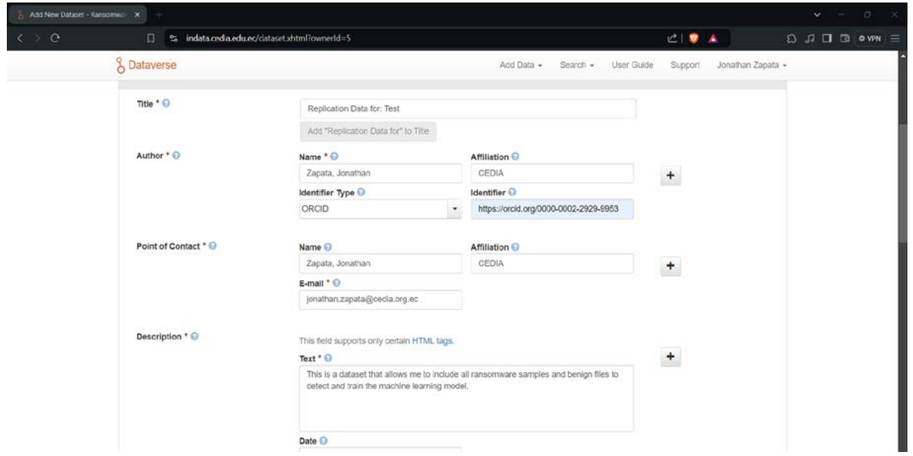


figura 16

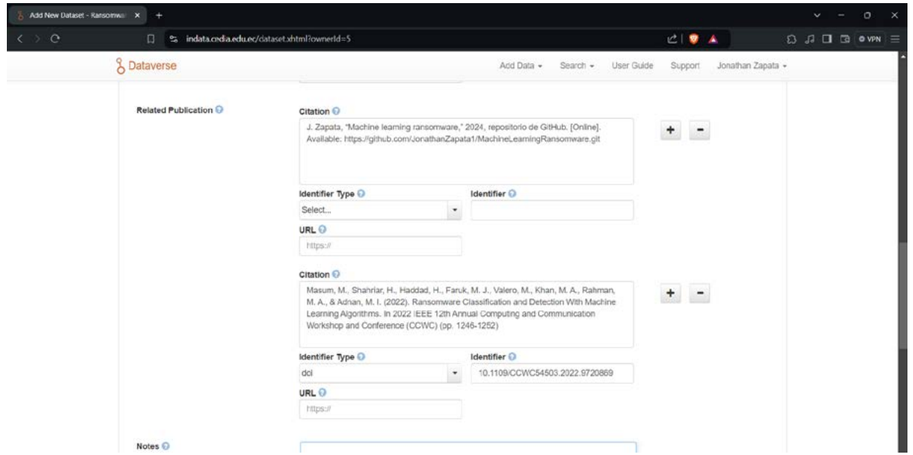


figura 17

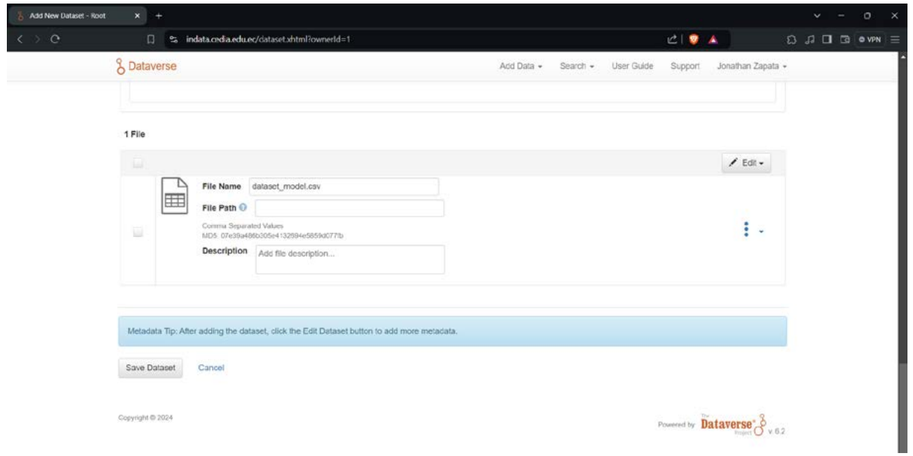


figura 18

Y finalmente creado el dataset dentro de nuestro repositorio o queda listo para publicar el Dataverse.

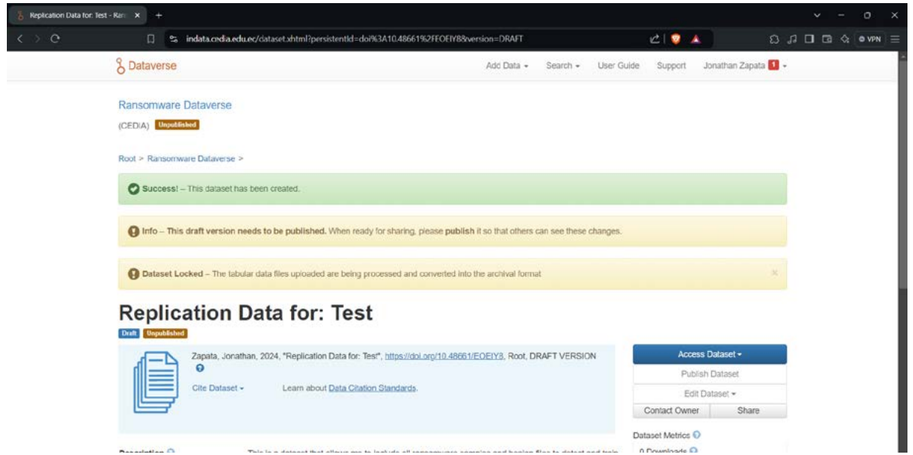


figura 19

Para que nuestro Dataverse puedan verlo otros investigadores que buscan mediante un repositorio de Indata CEDIA lo publicamos para que sea publico y de libre acceso

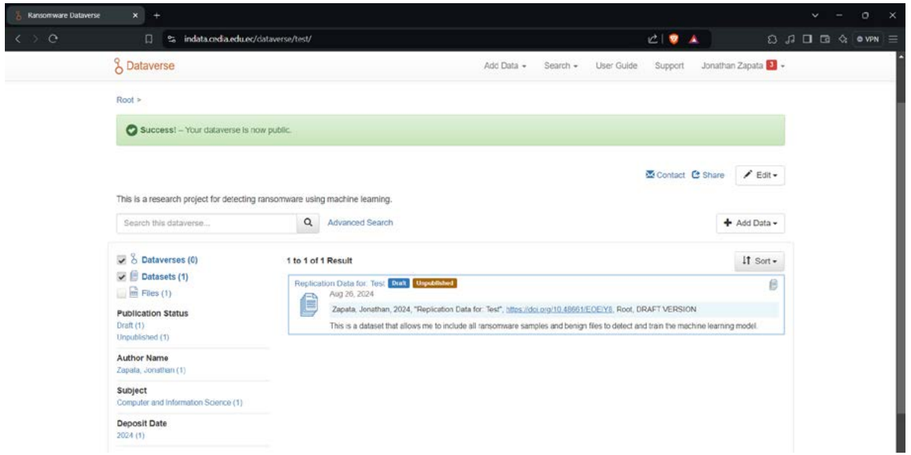


figura 20

Una vez que el archivo ha sido subido, podemos publicar nuestro dataset y compartirlo a través de diversos canales de comunicación, como Facebook, Twitter y LinkedIn.

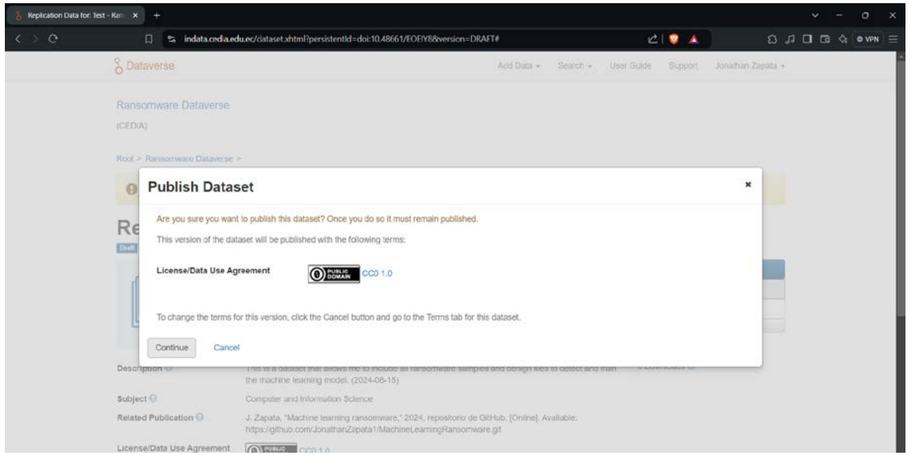


figura 21

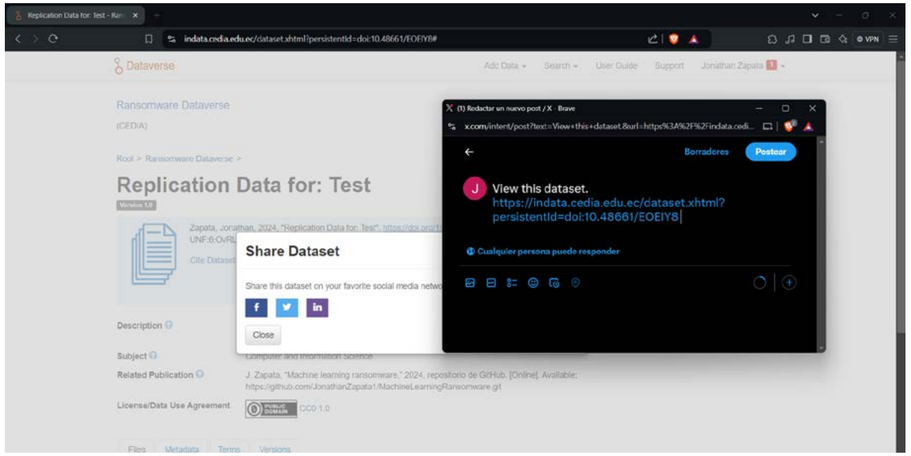


figura 22

Del mismo tenemos la opción de editar, eliminar y agregar el dataset del mismo modo los metadatos que acompañan a este:

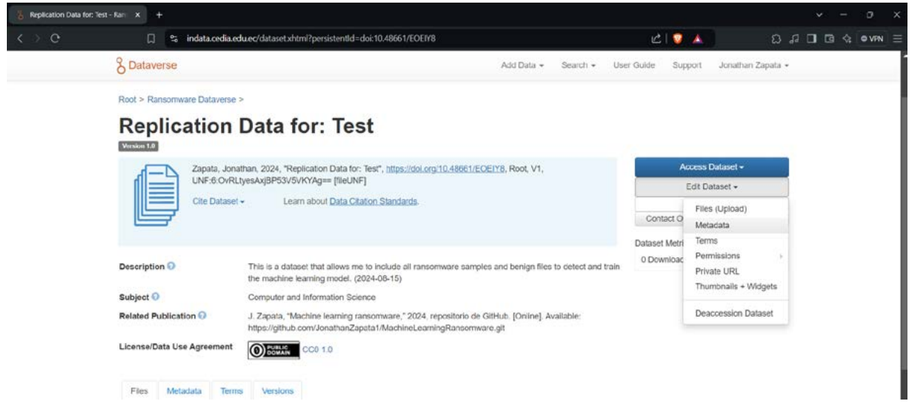


figura 23

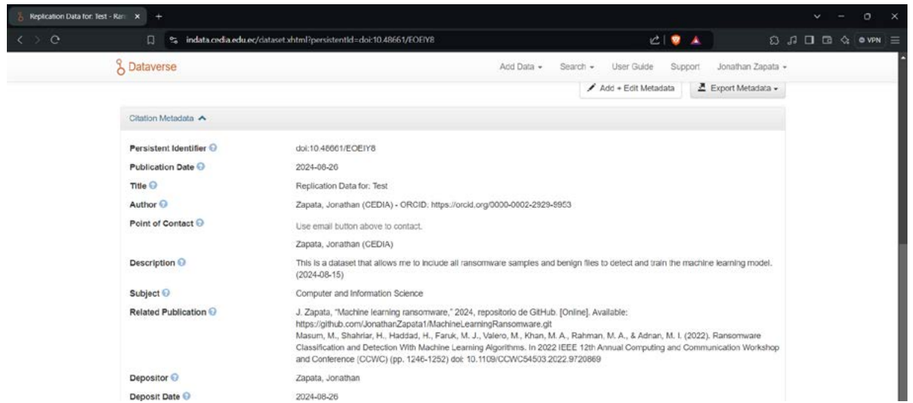


figura 24

Finalmente, nuestro Dataverse es público y accesible dentro de InDATA CEDIA, sin necesidad de realizar un login.

figura 25

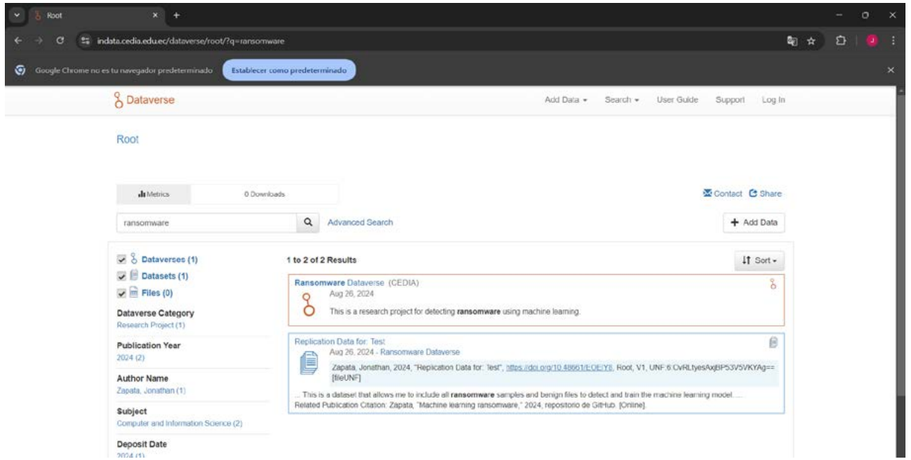


figura 26

Y descargar el Dataset

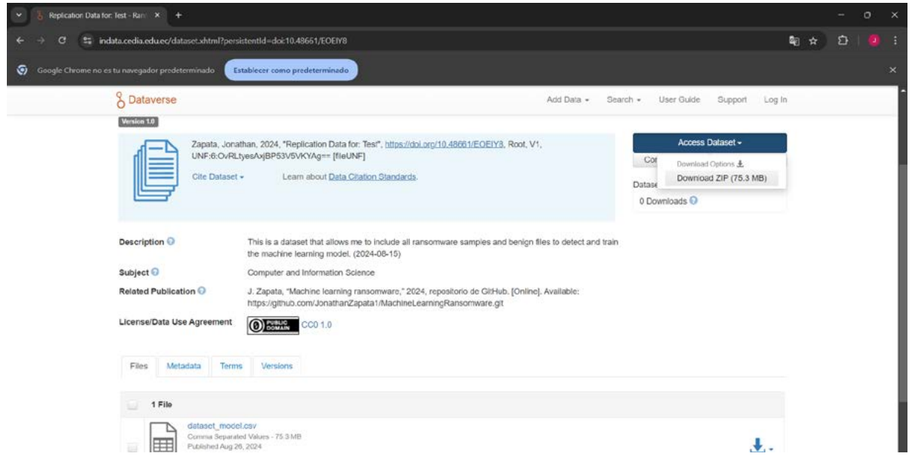


figura 27

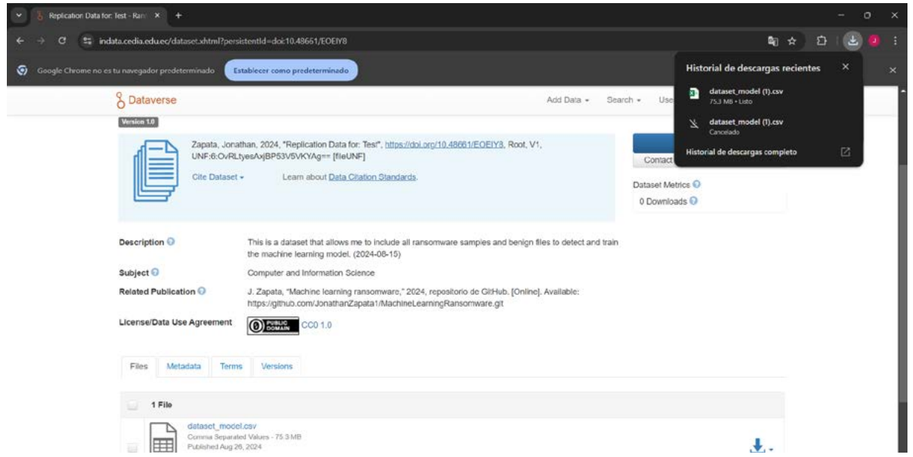


figura 28

Preprocesamiento

Durante el preprocesamiento, aplicamos técnicas de selección de modelos y utilizamos herramientas en Python, junto con librerías especializadas, para limpiar los archivos del dataset o CSV. Esto incluye la eliminación de duplicados y el manejo de celdas vacías. Utilizamos **Jupyter de HPC** para realizar estas tareas y subimos el archivo CSV al entorno.

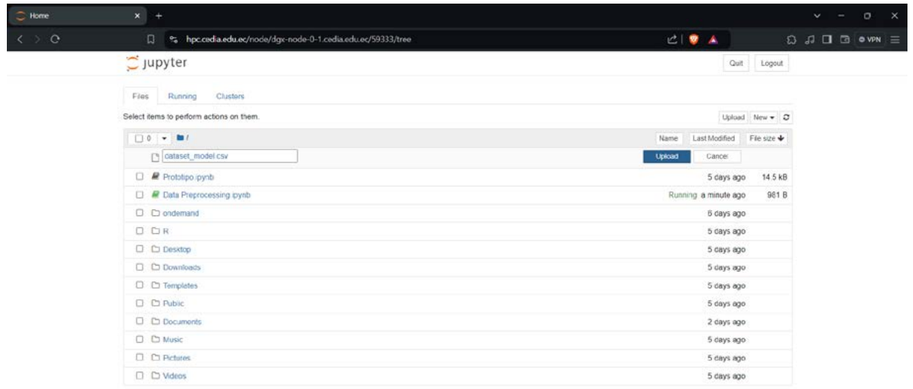


figura 29

Y ejecutamos el siguiente algoritmo.

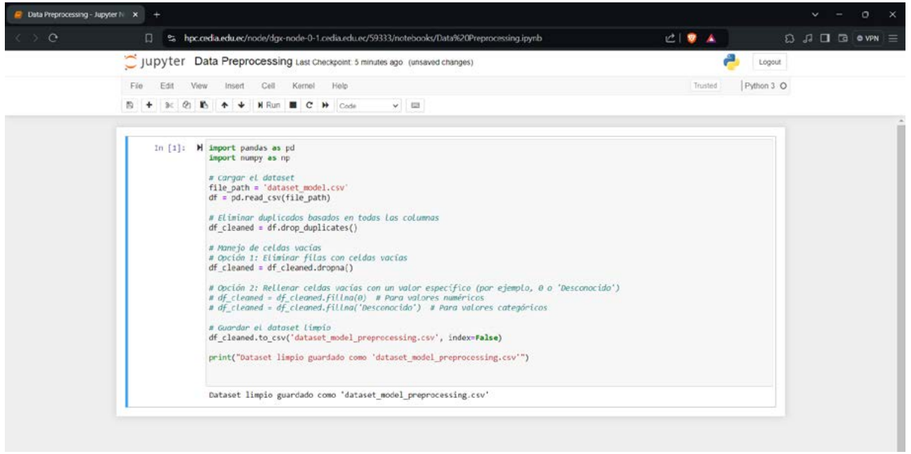


figura 30

NOTA: Aunque el tiempo de ejecución del clúster en Jupyter finalice, tanto los datasets como los archivos .py relacionados con las ejecuciones en Python se guardarán en tu carpeta personal.

Entrenamiento del modelo

Antes de continuar con el entrenamiento del modelo es la ejecución e instalación de librerías que se van a usar en ese te caso scikit-learn, matplotlib.

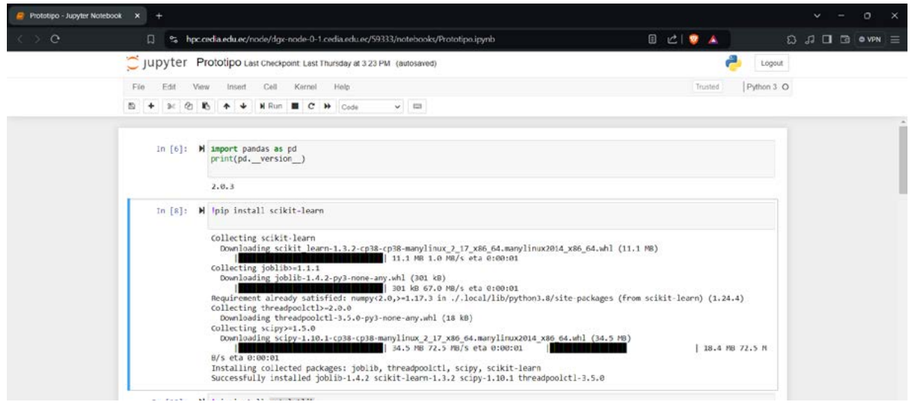


figura 31

Después, comienza la codificación y la creación del modelo a entrenar, empezando con el algoritmo Random Forest

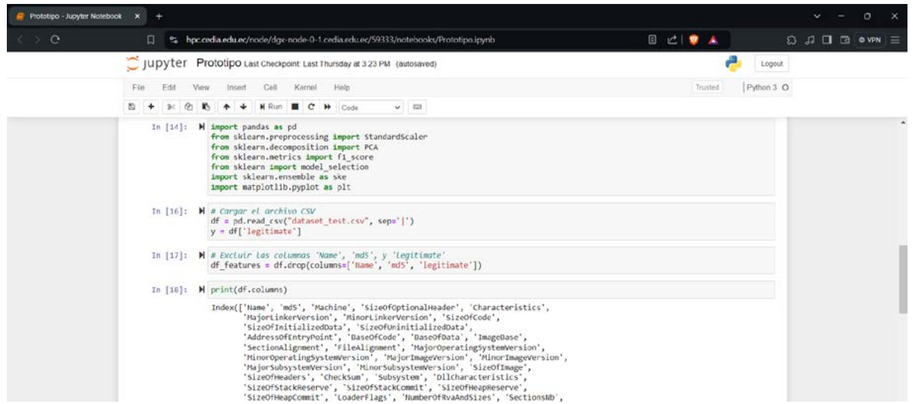


figura 32

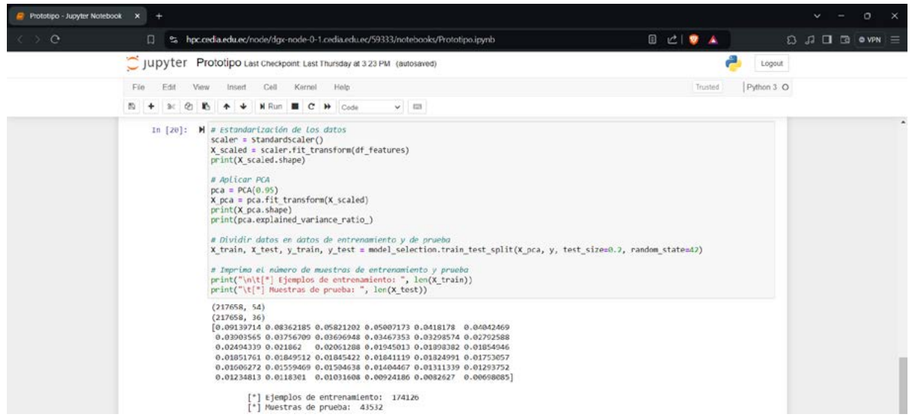


figura 33

El conjunto de datos está en un 80% para entrenamiento y un 20% para prueba. Esta división permitió entrenar el modelo y evaluar qué variables podrían ajustarse o qué métodos adicionales podrían explorarse para mejorar la calidad del modelo en el proceso de aprendizaje automático.

Y seguidamente los algoritmos restantes:

- Decision Tree

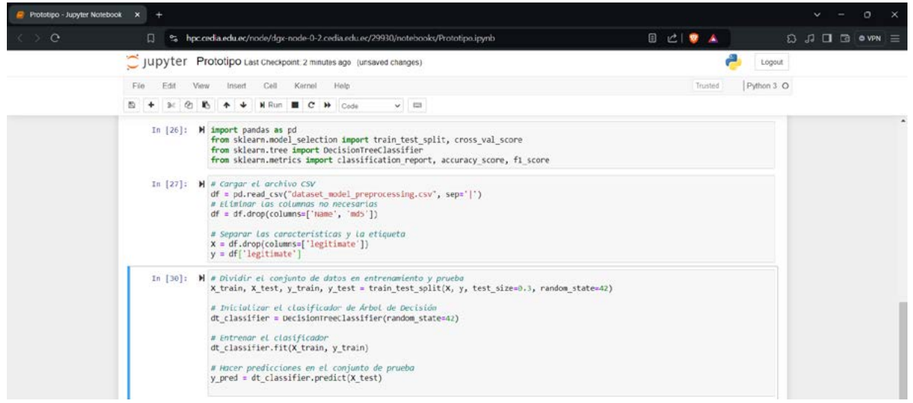


figura 34

- AdaBoost

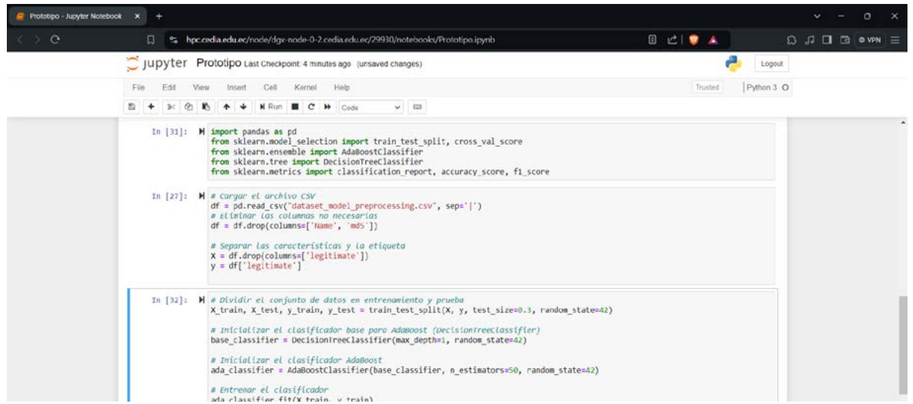


figura 35

- K-Nearest Neighbors (KNN)

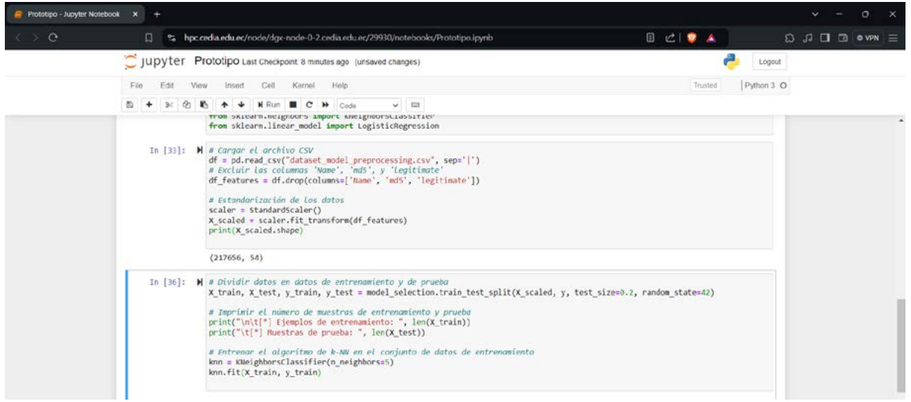


figura 36

- Logistic Regression

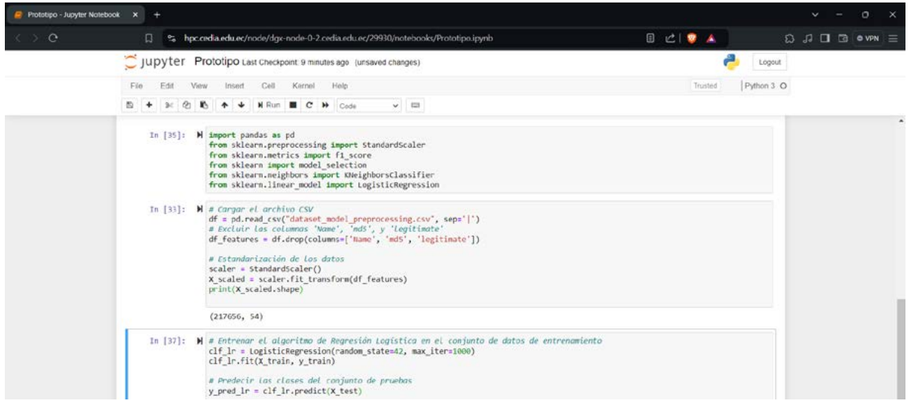


figura 37

- Naïve Bayes

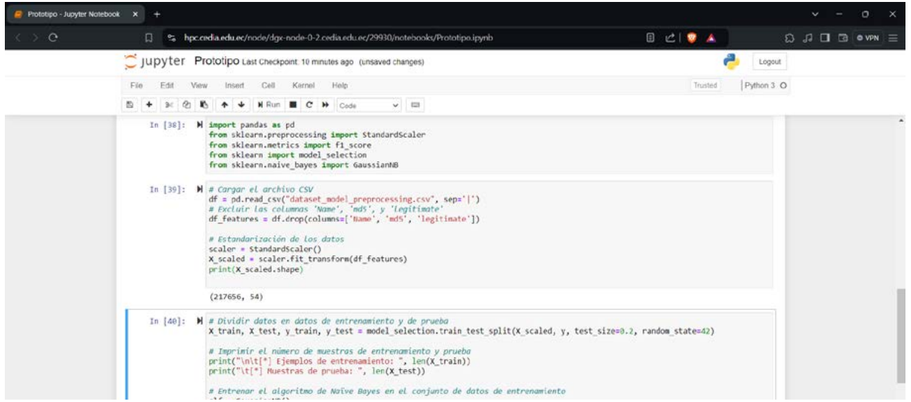


figura 38

Evaluación

Finalmente, al verificar y evaluar cada uno de los algoritmos, llegamos a los modelos de calificación utilizando métricas como la puntuación F1, la exactitud y la validación cruzada.

- Random Forest

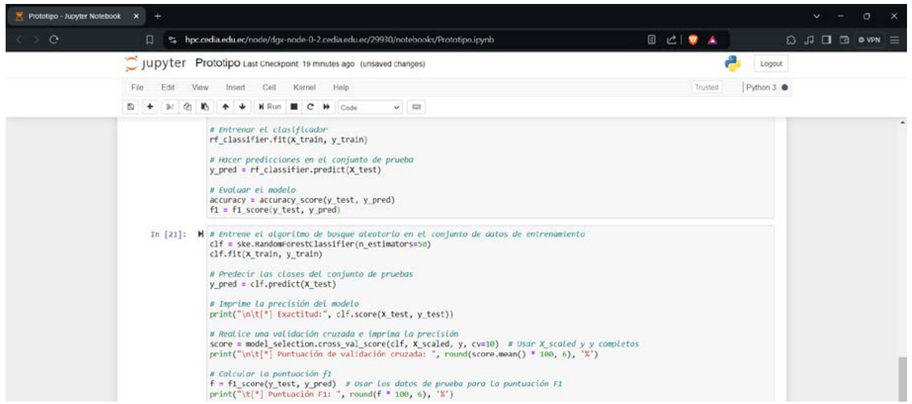


figura 39

- Decision Tree

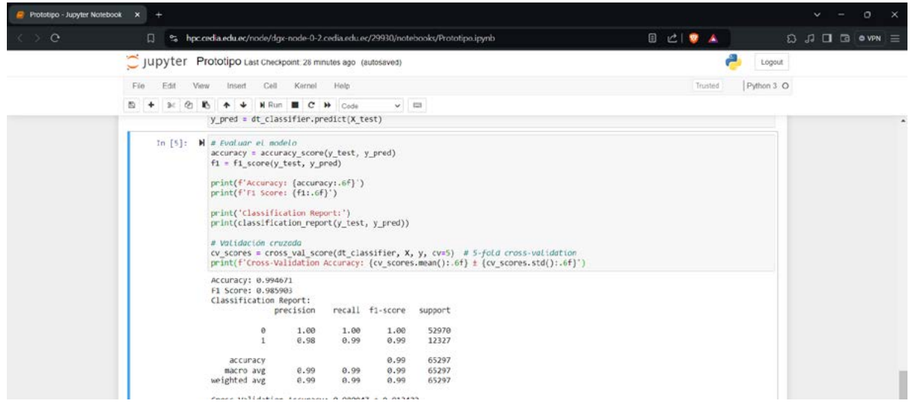


figura 40

- AdaBoost

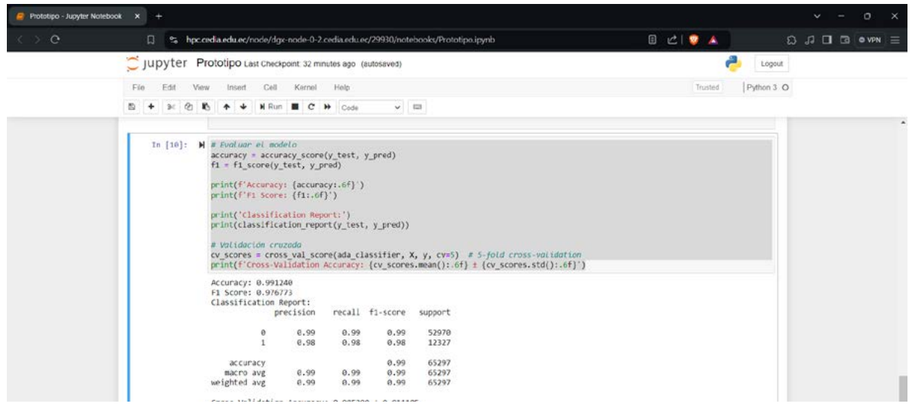


figura 41

- K-Nearest Neighbors (KNN)

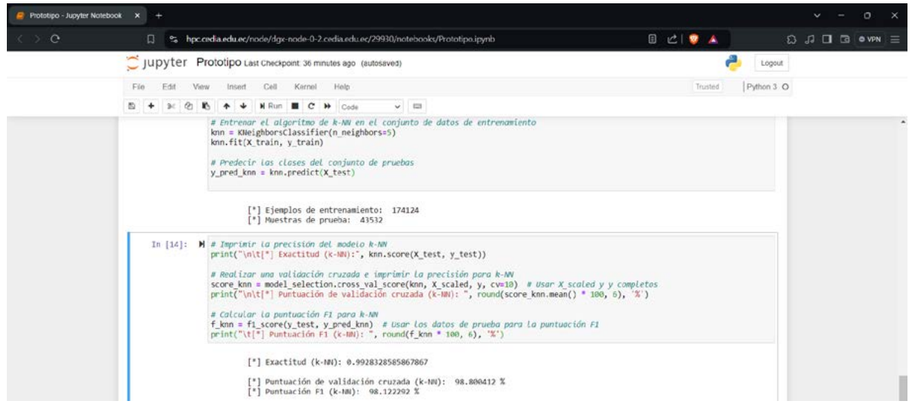


figura 42

- Logistic Regression

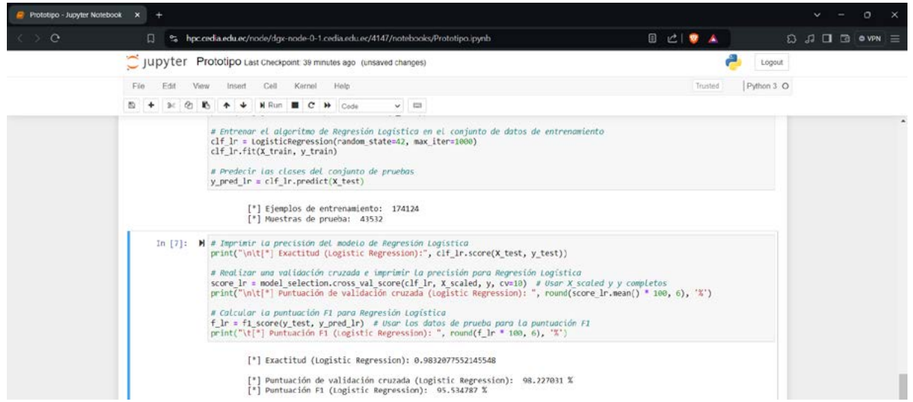


figura 43

- Naïve Bayes

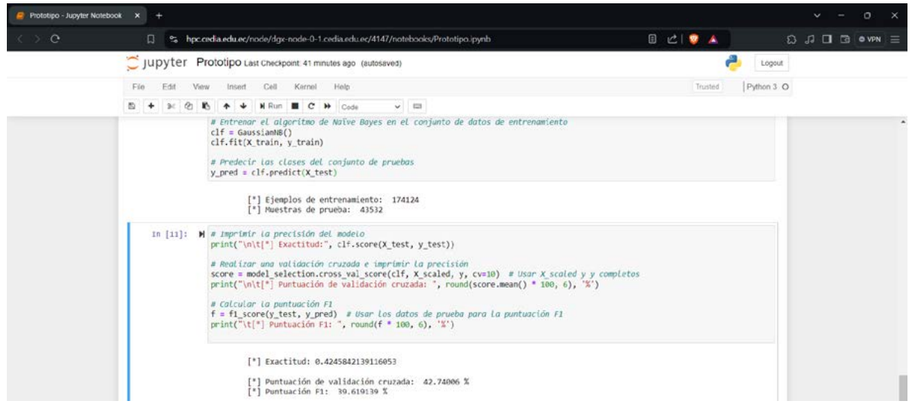


figura 44

De manera similar, podemos graficar estos datos para validar los modelos de evaluación mediante la puntuación F1, la validación cruzada y la precisión, utilizando diagramas de confusión. Esto permitirá visualizar de forma efectiva el rendimiento de los modelos y facilitará la interpretación de los resultados obtenidos.

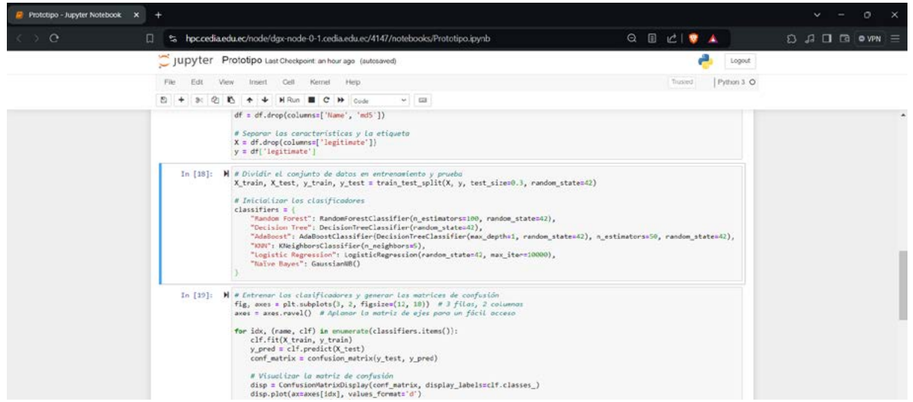


figura 45

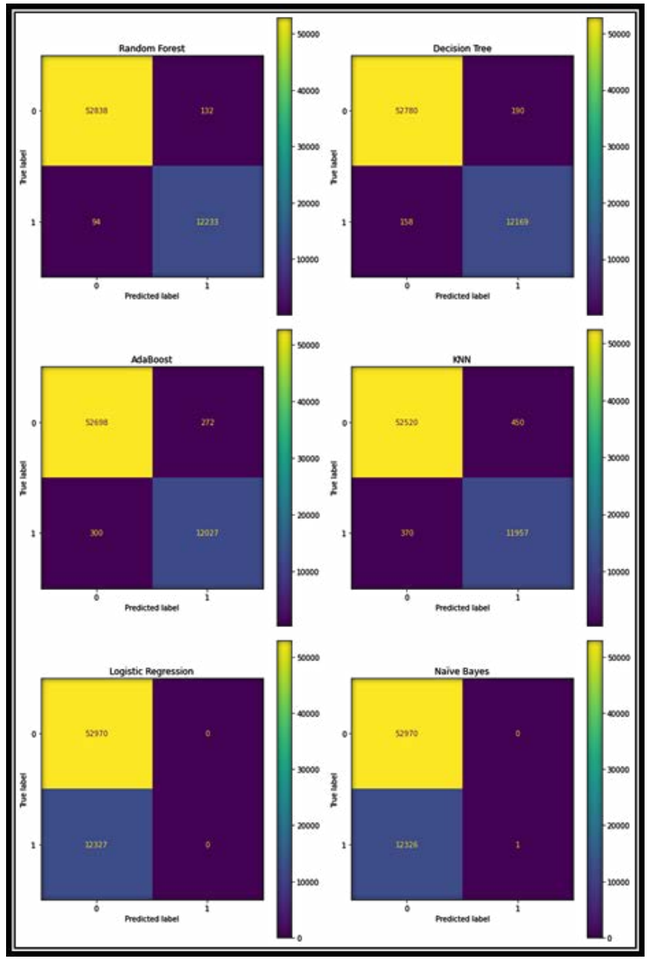


figura 46

Con esto, completamos la investigación sobre el uso de machine learning para la detección de ransomware. Aunque el proceso puede ser arduo, la utilización de supercomputadoras HPC puede optimizar significativamente las tareas de optimización y reducir los tiempos de espera durante la compilación del código. Esto resulta especialmente útil en investigaciones que requieren procesar y preprocesar grandes cantidades de datos, ya sea para predicciones o para la visualización de datos.

1.4

USO DE LA HERRAMIENTA GITLAB

GitLab

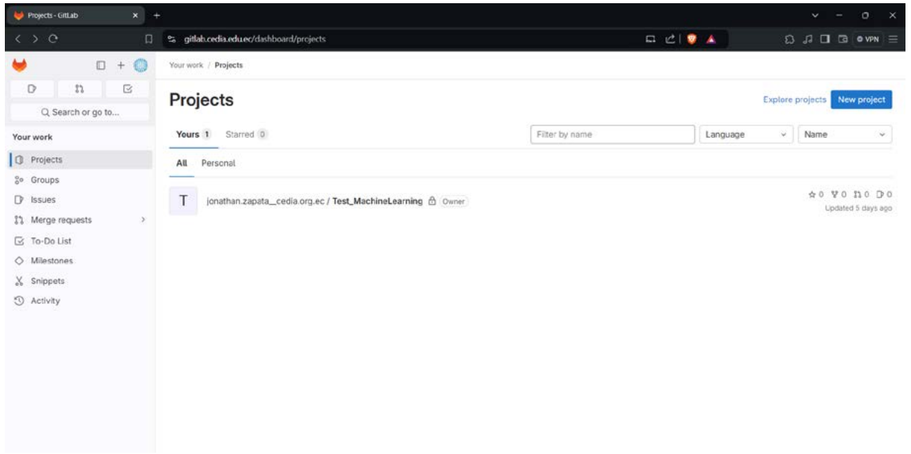


figura 47

Además, al ejecutar estos scripts de desarrollo en Python, es fundamental capturarlos o guardarlos adecuadamente para facilitar la colaboración en equipo y permitir el mantenimiento o las mejoras continuas del código. Por ello, se utiliza la plataforma GitLab para almacenar todo en un único repositorio, asegurando así una gestión eficiente y un acceso centralizado a los scripts.

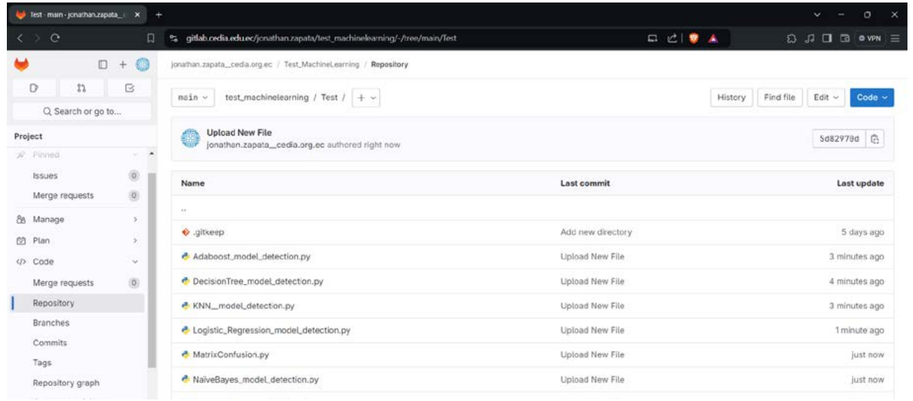


figura 48

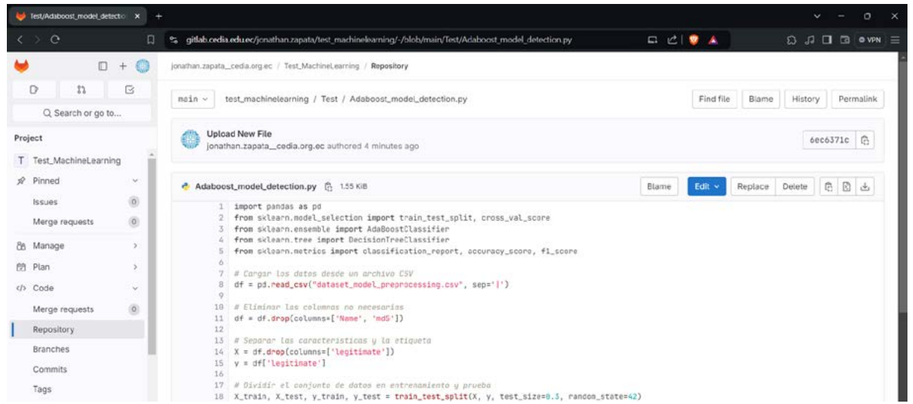


figura 49

Para maximizar el uso de **GitLab**, convertimos la investigación en una prueba de concepto, desarrollando una aplicación web. Además, establecimos una conexión directa con GitLab para gestionar el proyecto.

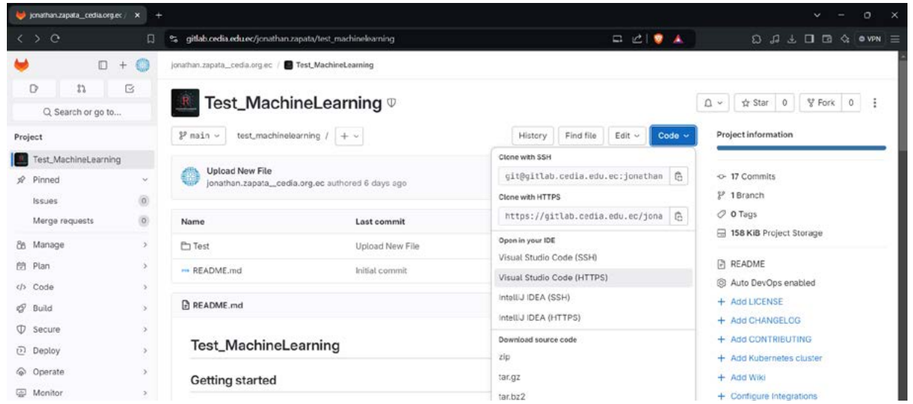


figura 50

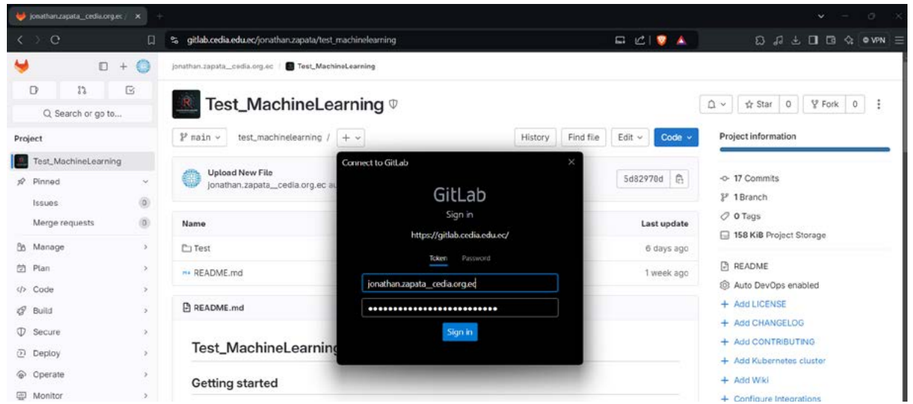


figura 51

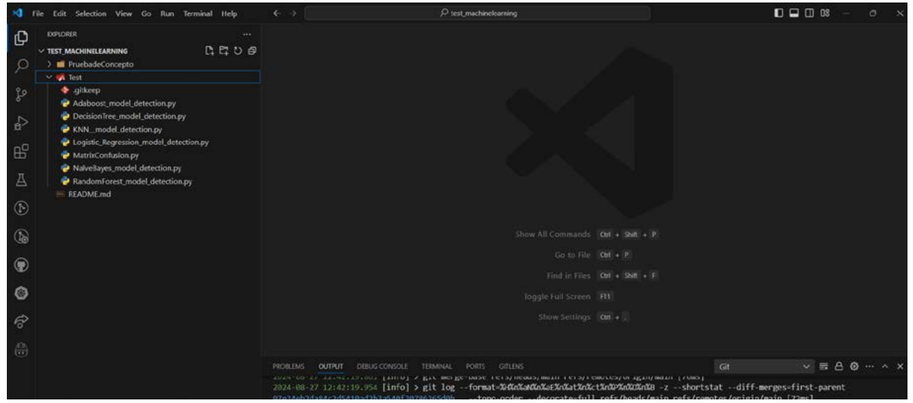


figura 52

Y creamos un pequeño prototipo usando python y Visual Studio CODE

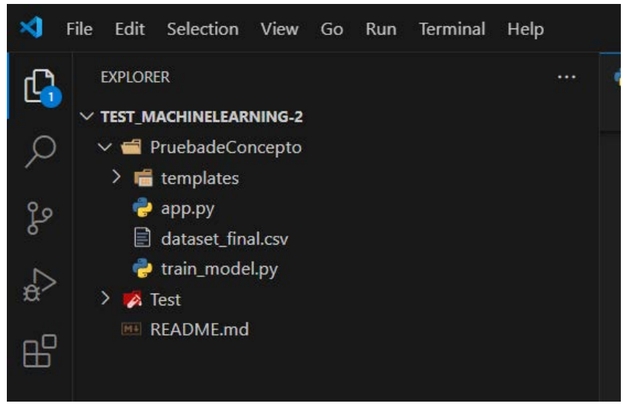


figura 53

Para crear los clasificadores o modelos de entrenamiento de machine learning y generar archivos `.pkl` que almacenan dichos modelos, utilizamos **Jupyter**.

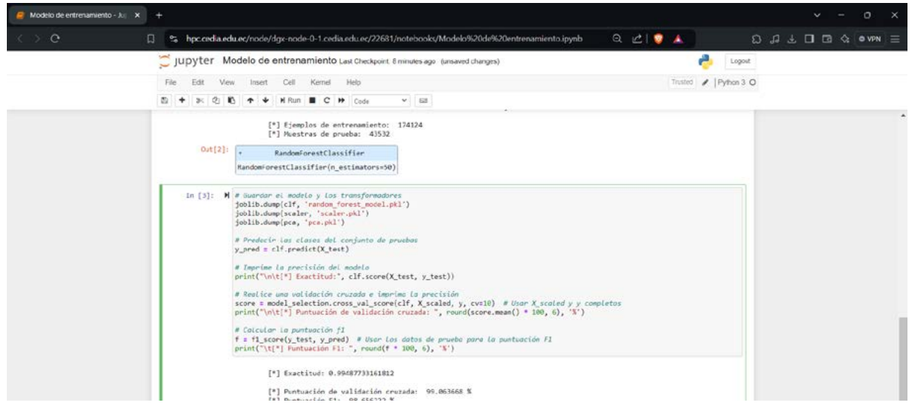


figura 54

figura 55

Y finalmente se lo puede descargar e implementar al prototipo

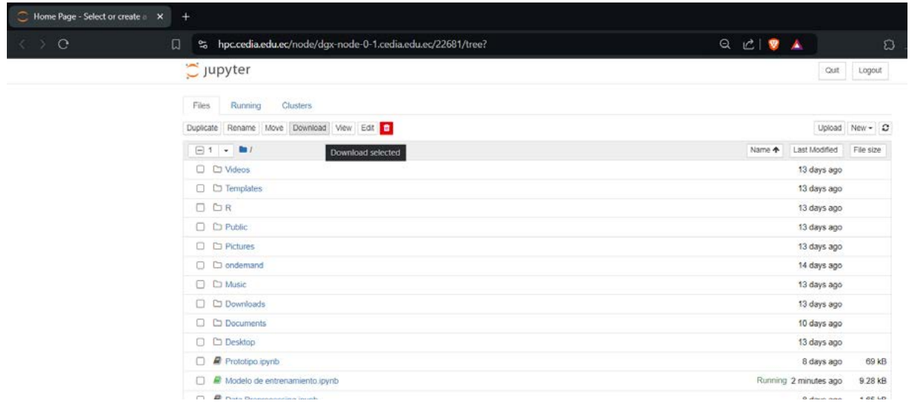


figura 56

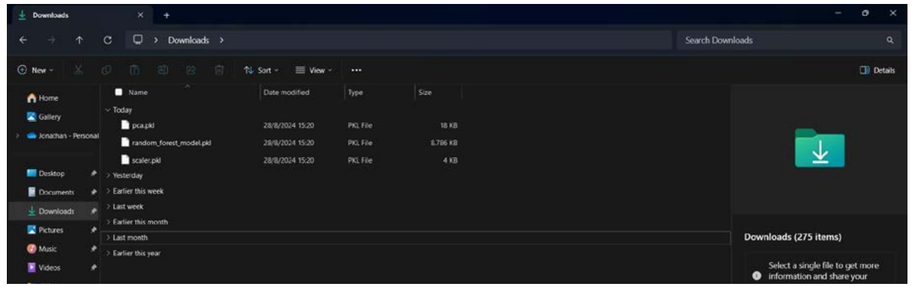


figura 57

Para maximizar el uso de **GitLab**, convertimos la investigación en una prueba de concepto, desarrollando una aplicación web. Además, establecimos una conexión directa con GitLab para gestionar el proyecto.

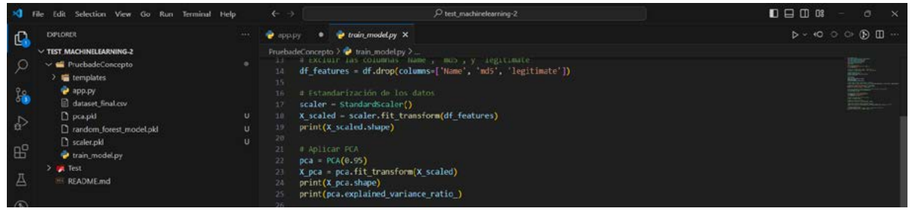


figura 58

Y ejecutamos el prototipo:

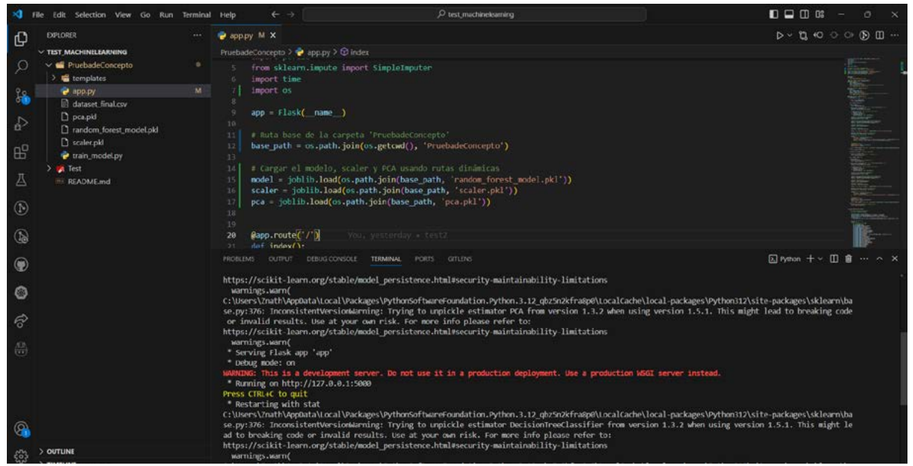


figura 59

figura 60

Para implementar la codificación en la plataforma Git, es fundamental considerar la colaboración y el trabajo en equipo, permitiendo que todos los integrantes del proyecto puedan contribuir y aportar de manera efectiva a la investigación.

Vamos a la **invitación de colaboradores**.

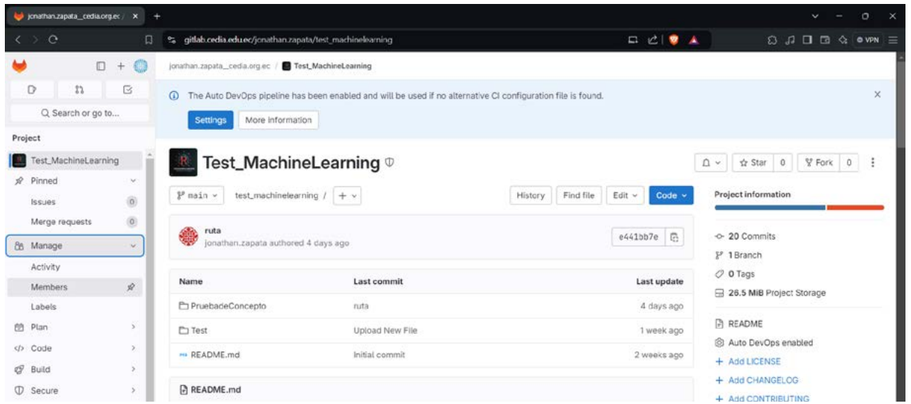


figura 60.1

Nos dirijimos a Manage >Members y finalmente Invite Members . A continuación, invita a todos los colaboradores que participarán en el proyecto.

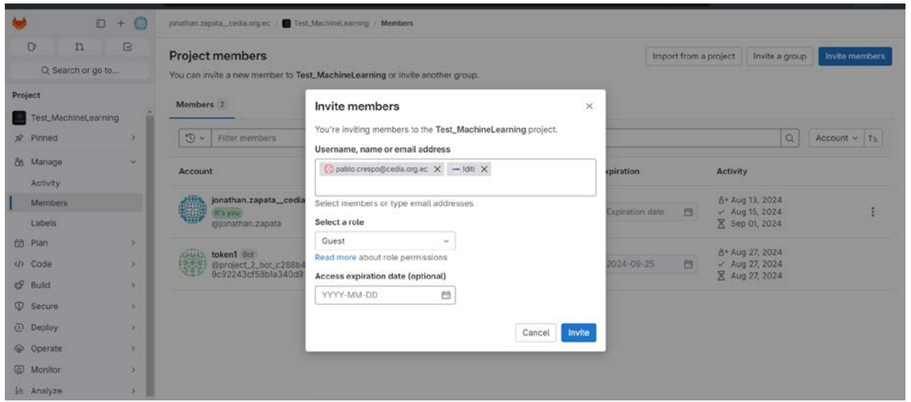


figura 60.2

Y aqui es muy importante el tipo de rol a designar que esta entre Guest, reporter, developer, maintainer y owner.

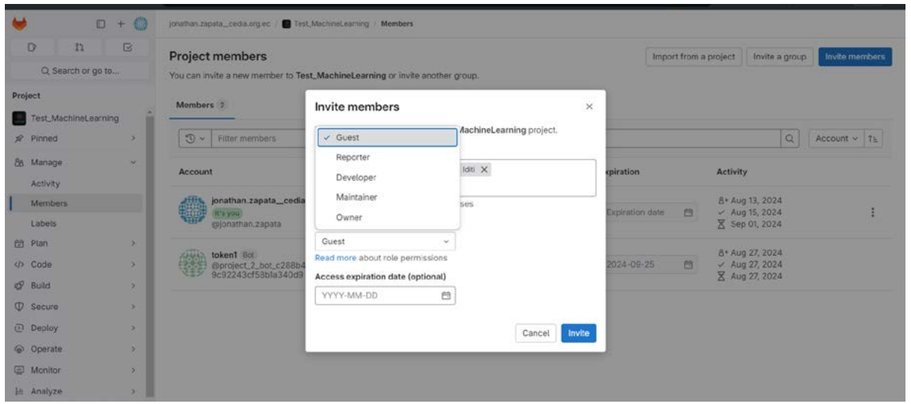
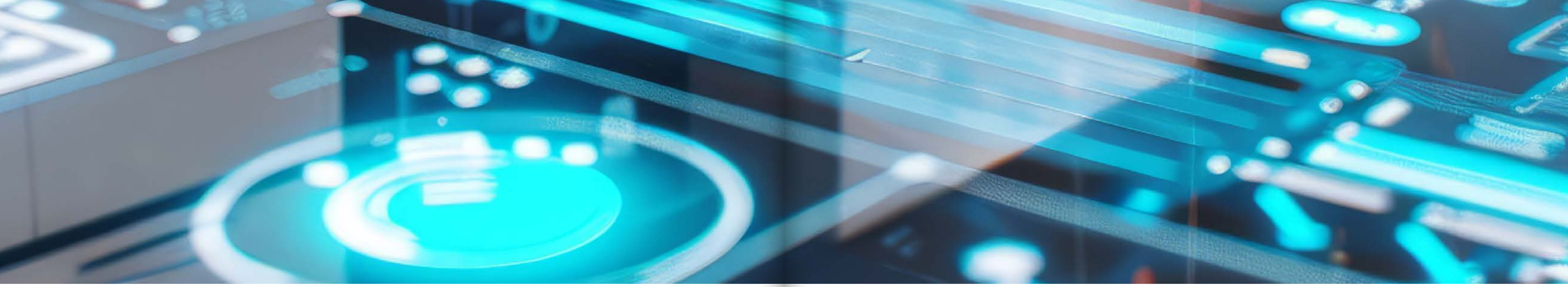


figura 60.3



En este caso, los designaremos como desarrolladores (developers), quienes se encargarán de codificar, modificar o mejorar el código, asegurando que los procesos de la investigación sigan el rumbo adecuado y se mantengan en la dirección correcta.

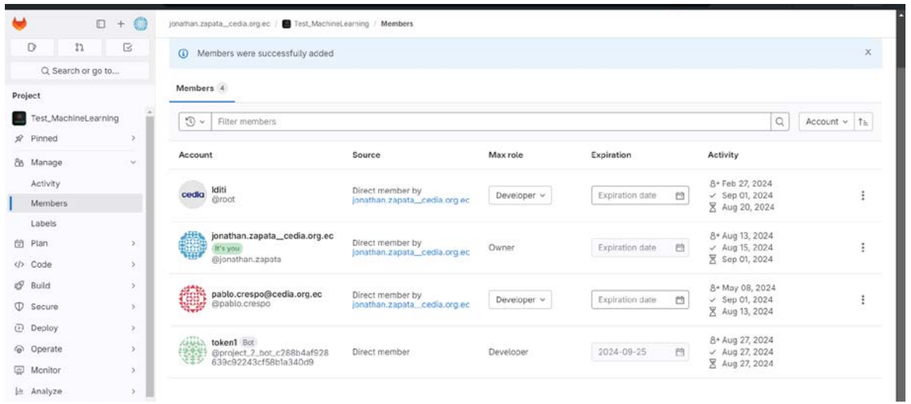


figura 60.4

Ahora procederemos a la asignación de permisos para cada colaborador en la rama principal. En GitHub, la rama principal (branch) representa la línea principal de desarrollo del código.
Nos dirigimos a Settings > Repository y configuramos los permisos para que tanto los mantenedores (maintainers) como los desarrolladores (developers) tengan acceso a la rama principal.

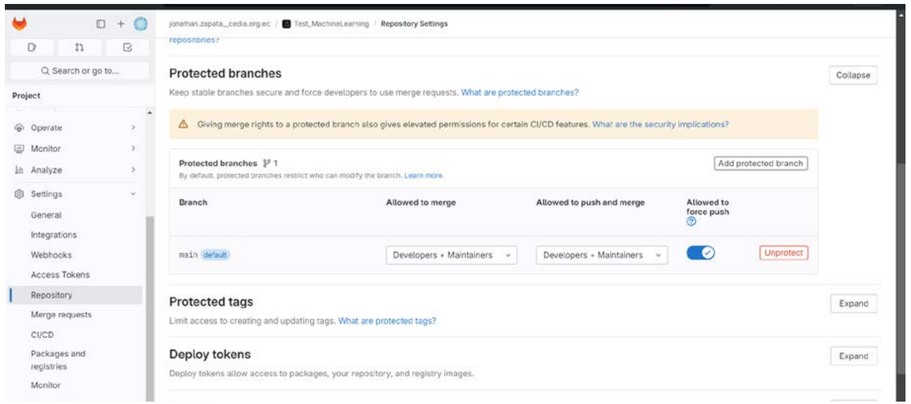


figura 60.5

También es posible crear ramas protegidas, permitiendo que solo el propietario o líder del proyecto tenga acceso para modificarlas. Esto asegura que las ramas críticas del desarrollo estén bajo control, evitando cambios no autorizados.

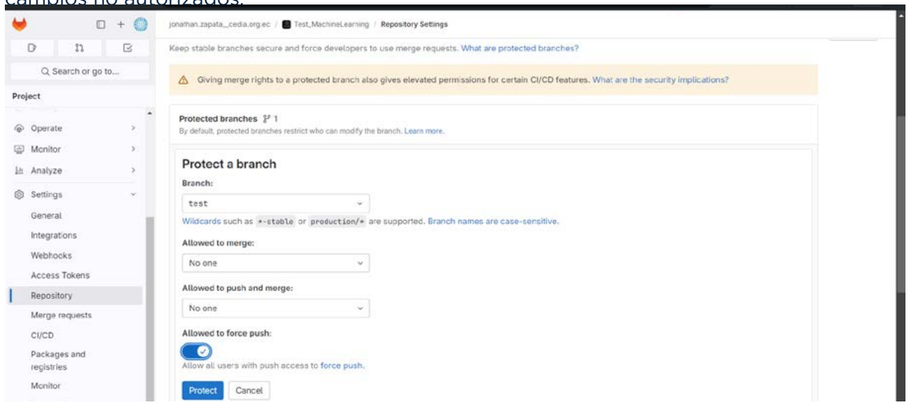


figura 60.6

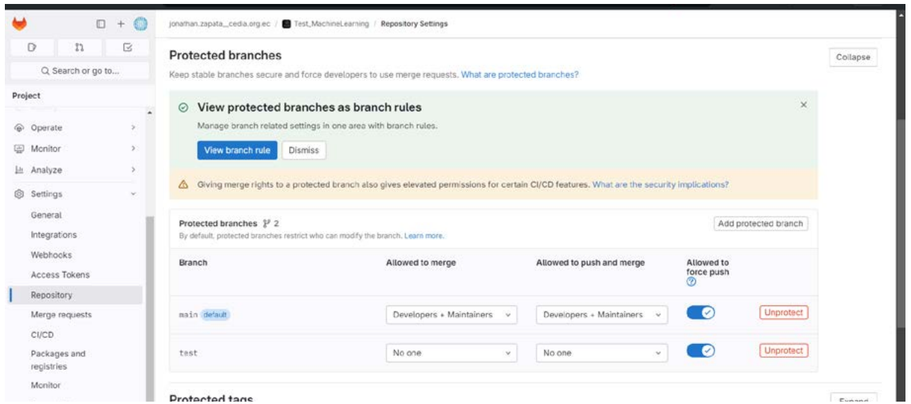


figura 60.7

Finalmente, para la revisión o la designación de un revisor en la investigación, se le puede asignar un token de acceso en caso de que el repositorio sea privado. Este token le permitirá ingresar al repositorio de Git, y se puede generar en la sección de Settings > Access Tokens.

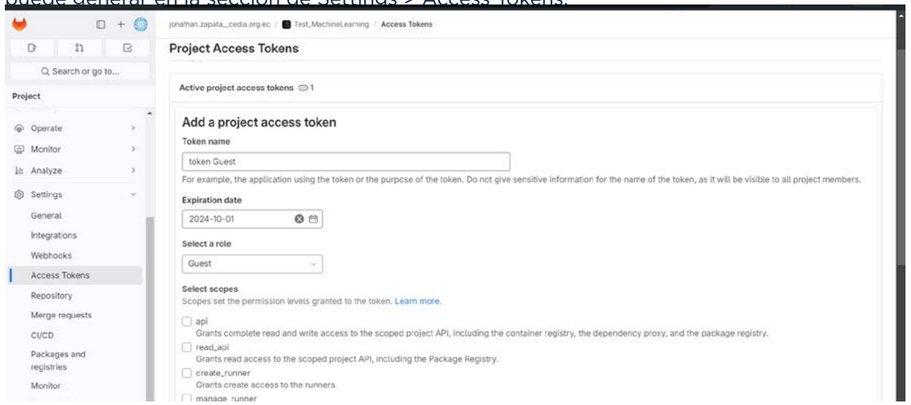


figura 60.8

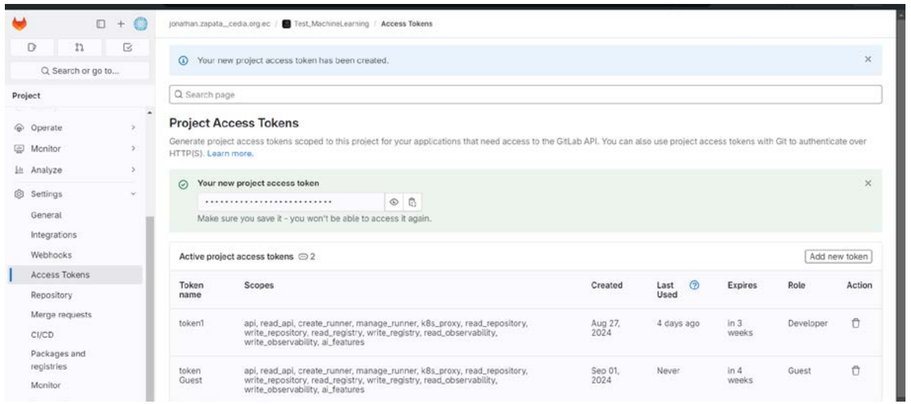


figura 60.9

Una vez se le entrega el token correspondiente, el revisor podrá clonar el repositorio y tener acceso al código.

1.5

USO DE LA HERRAMIENTA MATTERMOST

Mattermost

Finalmente, se utiliza Mattermost como la herramienta principal de comunicación entre los participantes de la investigación. Esta plataforma permite coordinar reuniones para discutir los resultados y el avance del proyecto. Se configura un canal con el nombre del proyecto o una palabra clave relevante, como “Machine Learning Ransomware”, y se invita a los colaboradores involucrados en la investigación para facilitar la colaboración y el intercambio de ideas en tiempo real.

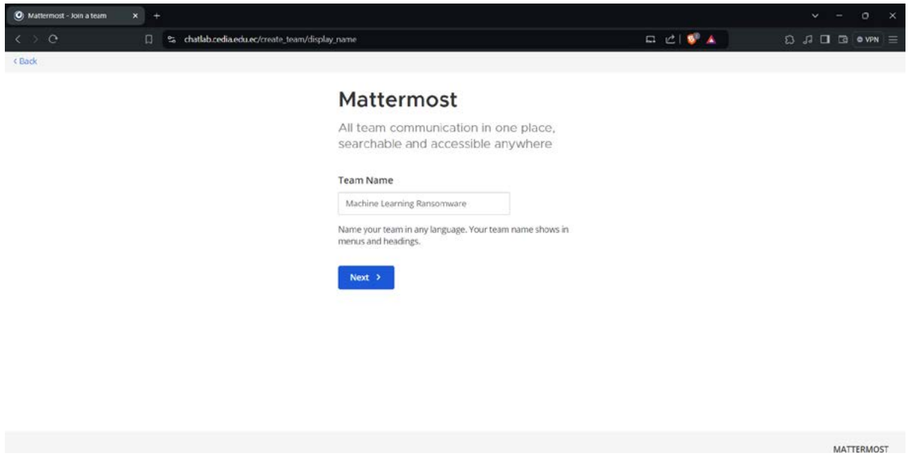


figura 61

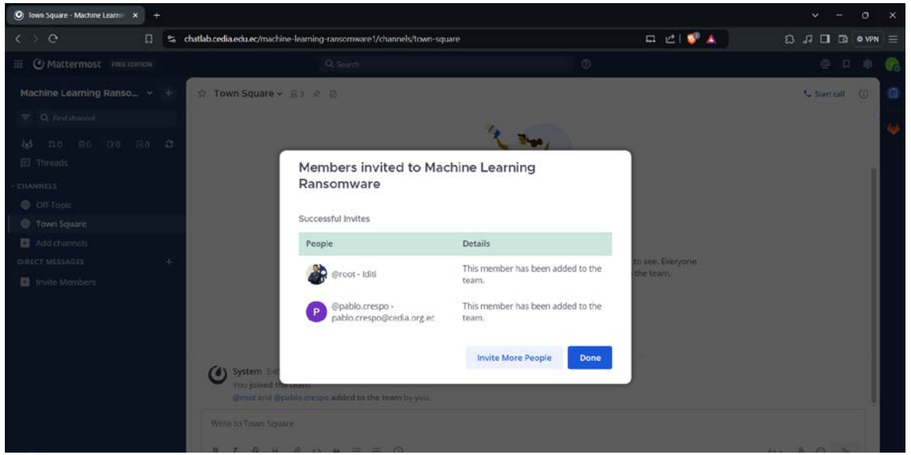


figura 62

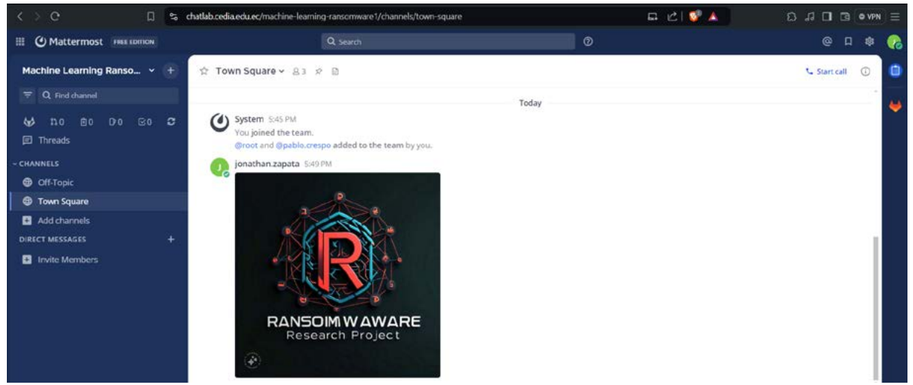


figura 63

Mattermost ofrece una amplia gama de herramientas, como la personalización del perfil, notificaciones, configuración de estado, y la capacidad de identificar y unirse a canales públicos o privados, así como enviar mensajes directos.

Además, es posible compartir código en formato Python, lo que permite a los investigadores visualizar el código que se está utilizando y evaluar posibles cambios o mejoras de manera más efectiva.

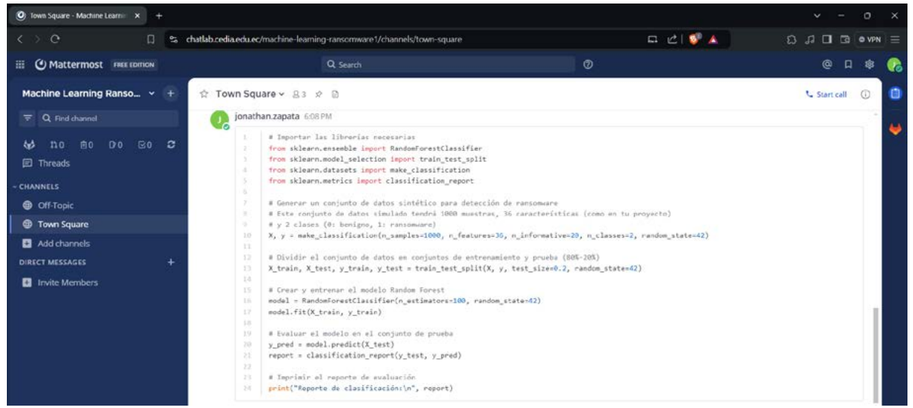


figura 64

Creación de PLAYBOOKS

Colección de procedimientos y directrices documentadas que guían a los equipos en la gestión de tareas o procesos repetitivos

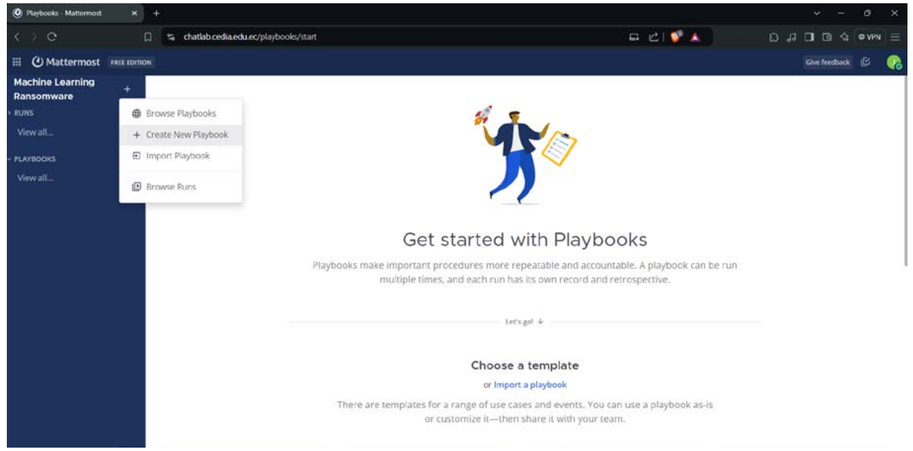


figura 65

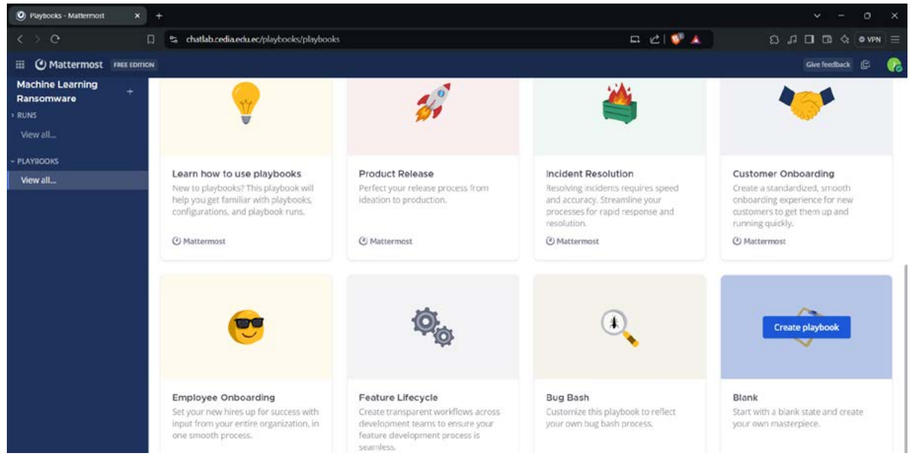


figura 66

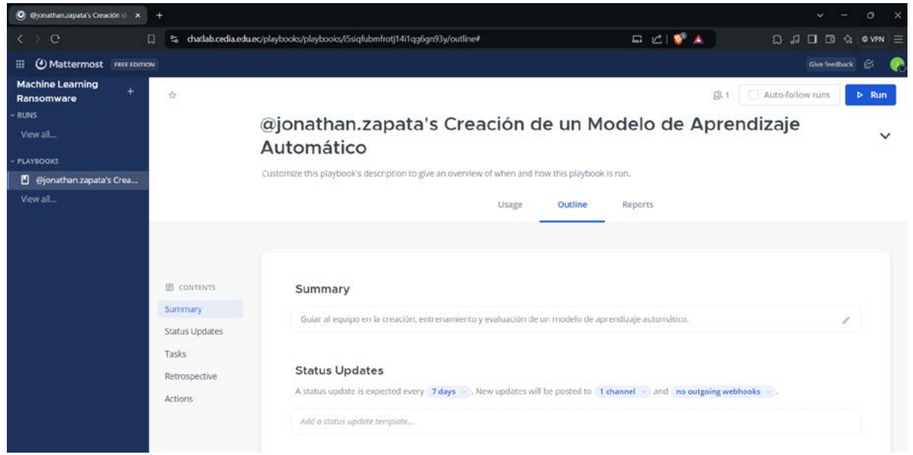


figura 67

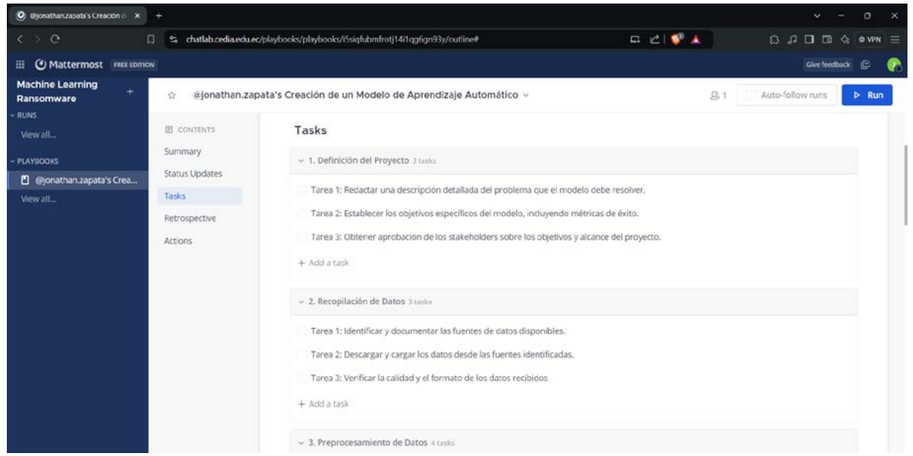


figura 68

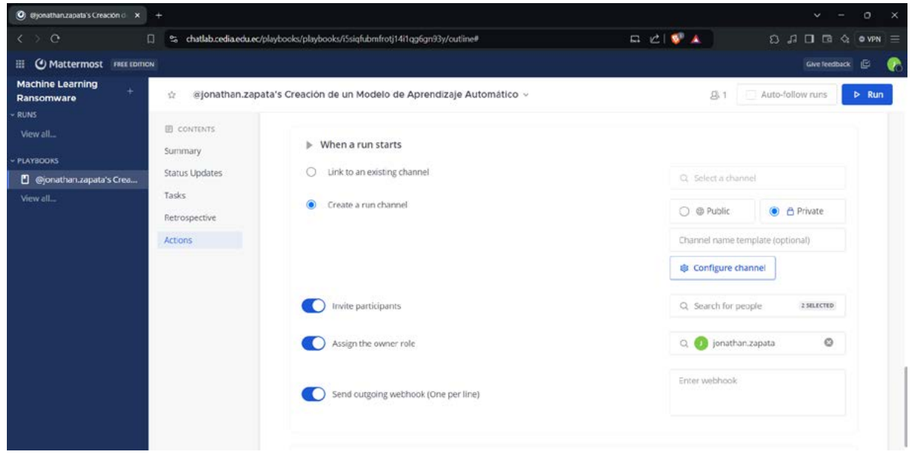


figura 69

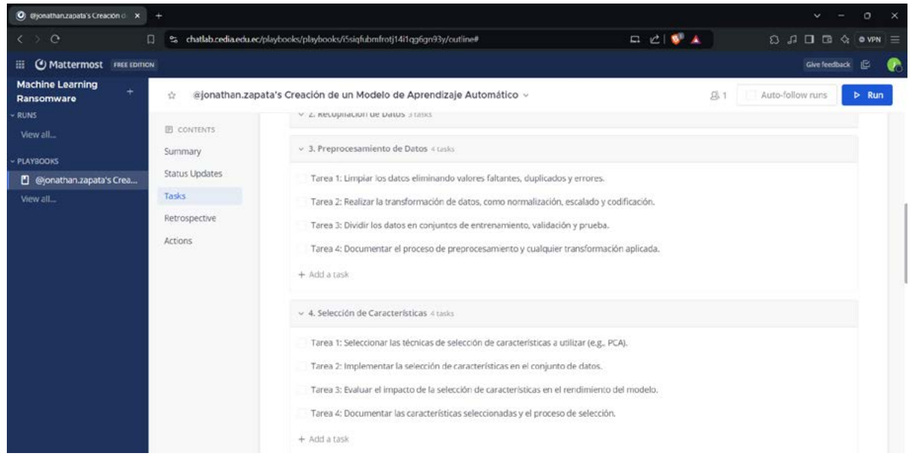


figura 70

Finalmente lo ejecutamos

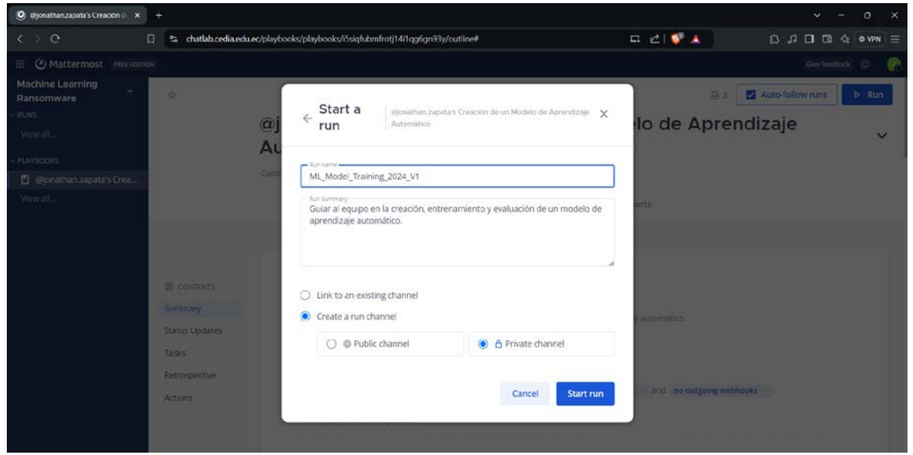


figura 71

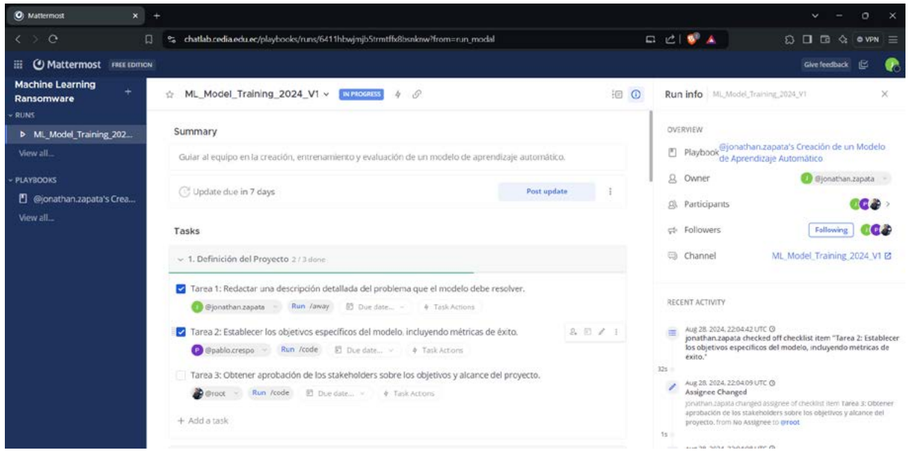


figura 72

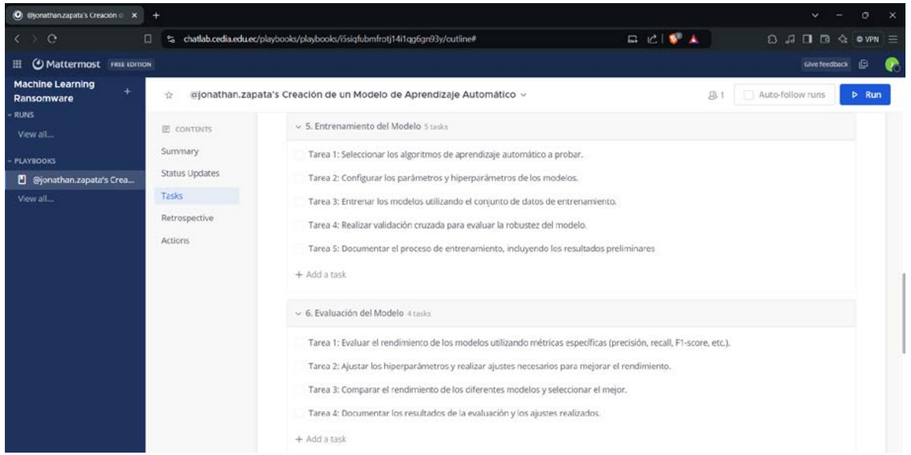


figura 73

Así tenemos un playbook bien elaborado que actúa como una guía integral que organiza y facilita el proceso de desarrollo de un modelo de aprendizaje automático, mejorando la eficiencia, la comunicación y la calidad del proyecto, del mismo modo podemos editar el playbook editando o agregando más tareas.

En este caso configurado a una actualización en 7 días, del mismo modo registrar las tareas ya hechas y realizar un seguimiento de estas, y con la designación de tareas a cada uno de los integrantes que fueron invitados al grupo.

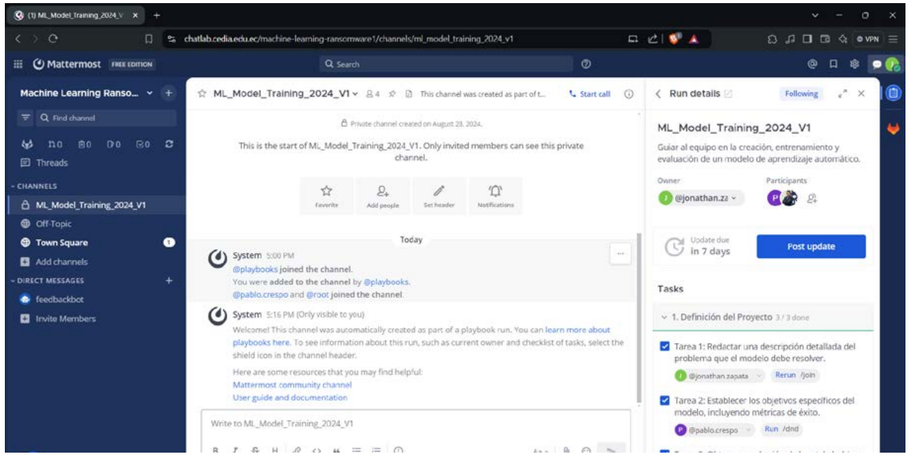


figura 74

En el slah command ubicamos el código /dnd que significa “La función No molestar está activada. No recibirás notificaciones push en tu computadora de escritorio o en tu dispositivo móvil hasta que la función No molestar esté desactivada.

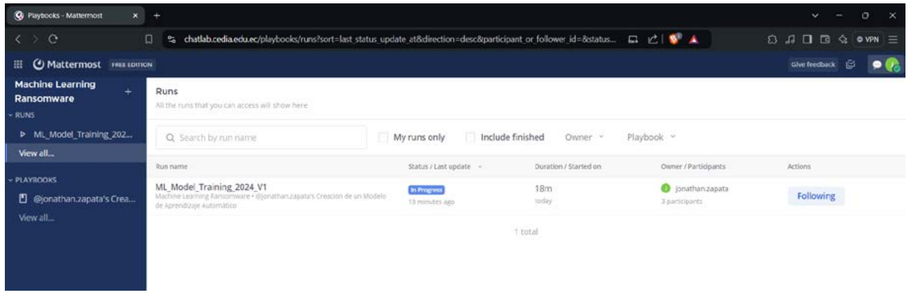


figura 75

Asimismo, Podemos publicar una actualización del proyecto

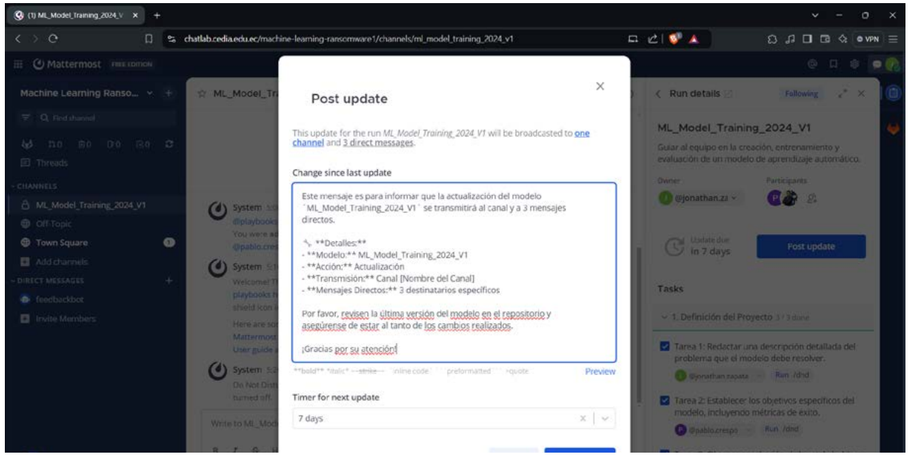


figura 76

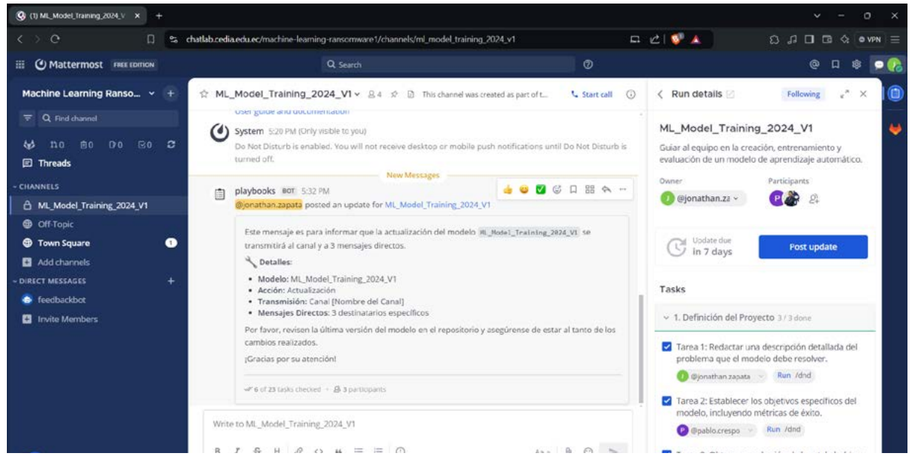


figura 77

Y se trasmitirá al canal principal

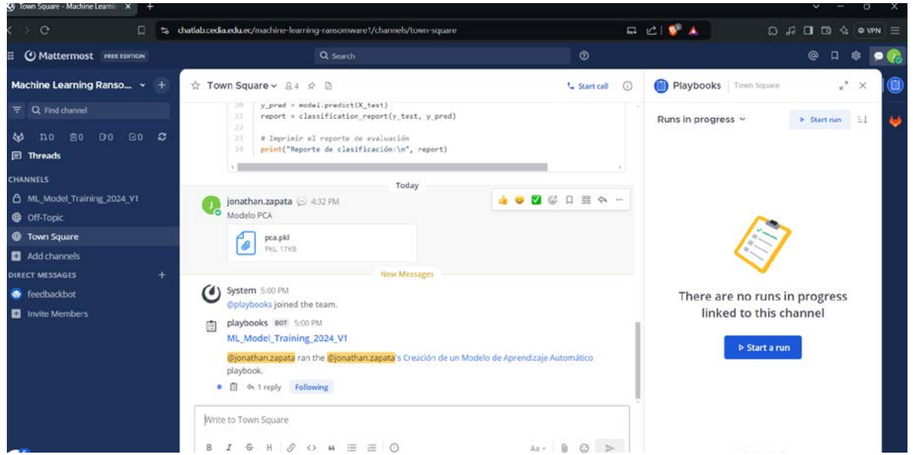


figura 78

1.6

USO DE LA HERRAMIENTA FILESENDER

filesender

Al utilizar la herramienta Filesender para enviar archivos de gran tamaño, designamos el archivo que queremos compartir—en este caso, nuestro dataset de aproximadamente 75.3 MB. Establecemos la fecha límite de expiración como el 11/09/2024 y procedemos a enviar el archivo al destinatario.

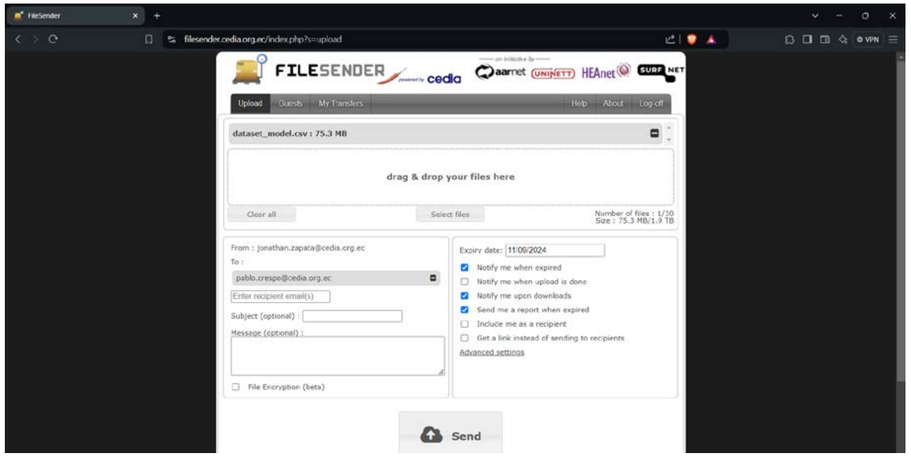


figura 79

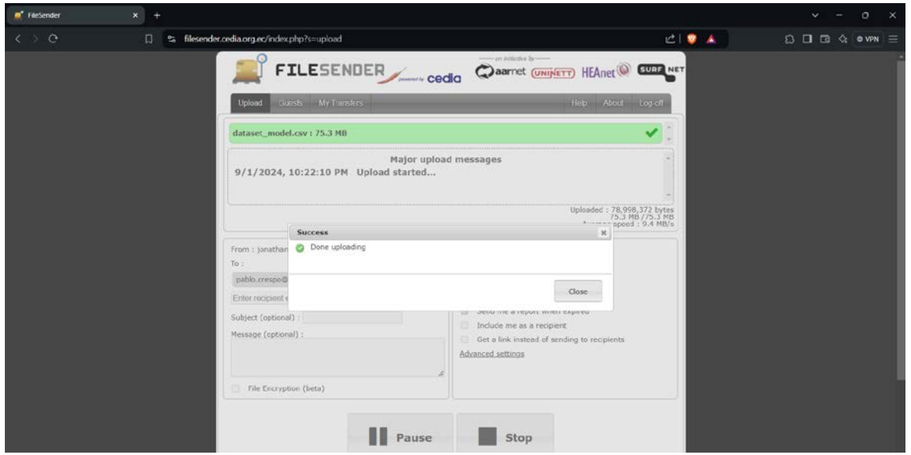


figura 80

research
lab
by cedia

cedia

“Conectando ideas, transformando sociedades”

cedia.edu.ec

@CEDIAec

CUE
Oficinas_
Gonzalo Cordero 2-122
y J. Fajardo esquina.
DIGITALS_
Miguel Moreno y Av. 10
de Agosto

UIO
Av. 12 de Octubre y
Lizardo García
Edificio Alto Aragón
Oficina 8A

GYE
Av. Del Bombero,
Km 6.5 - Edificio
La Vista de San Eduardo
5to piso Ofics. 510 y 511

PORTOVIEJO
Av. Metropolitana
Eloy Alfaro #2005
y Av. Olimpica.
Univ. San Gregorio

MANTA
Av. Circunvalación
Vía a San Mateo
ULEAM - EP