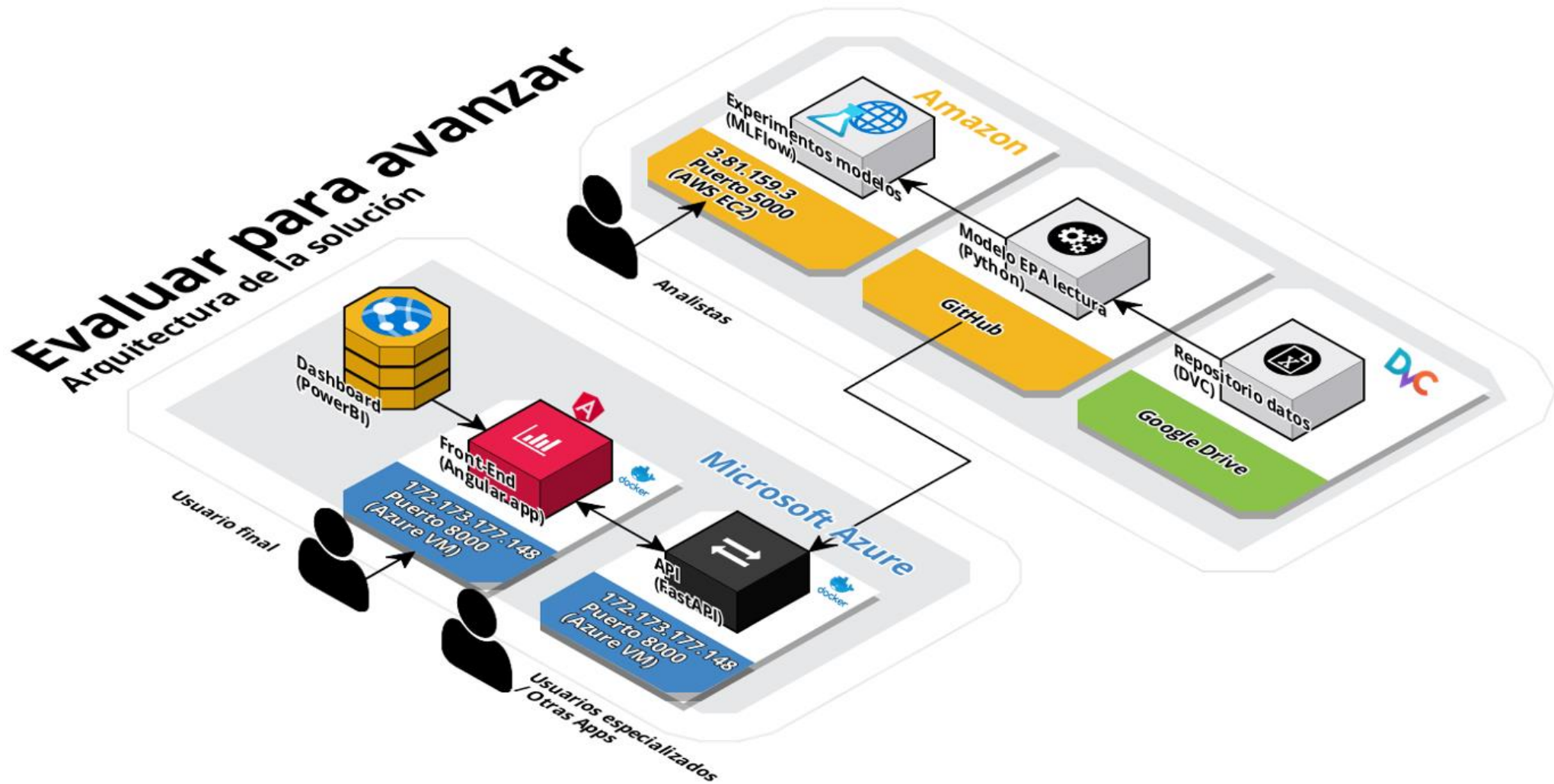


Despliegue de soluciones analíticas – MIAD

Proyecto – Entrega final – Grupo 11

Manual de instalación del tablero



Introducción

En el diagrama de arquitectura de la solución pueden observarse los componentes empleados tanto en la construcción y entrenamiento de modelos como aquellos componentes responsables de la visualización e interacción con el modelo predictivo.

Una vez la solución es implementada por primera vez y su uso es monitoreado o las necesidades de los usuarios cambian pueden requerirse cambios en uno o varios componentes de la solución. El objetivo del presente documento es describir brevemente el procedimiento de inclusión y despliegue en cada caso que se presenta a continuación. Se sugiere antes de cada caso, crear una nueva rama local a partir de la rama principal en el repositorio en GitHub.

Manual de instalación del tablero	1
Introducción	2
¿Cómo versionar cambios o inclusión de nuevos datos para el modelo?	2
¿Cómo registrar experimentos en el servidor MLFlow?	2
¿Cómo desplegar nuevas versiones del modelo EPA_ModeloLectura?	3
¿Cómo desplegar cambios en el API?	3
¿Cómo desplegar cambios en el componente Dashboard de PowerBI?	4
¿Cómo desplegar cambios en la aplicación FrontEnd?	4

¿Cómo versionar cambios o inclusión de nuevos datos para el modelo?

- Usar el comando `dvc pull` para obtener los archivos de datos de la solución. Estos archivos pueden estar clasificados en folders según su estado de procesamiento. (Ver <https://github.com/FabrizioMorales01/DataIcfesAnalysisDSA#c%C3%B3mo-est%C3%A1-estructurado-este-repositorio>)
- Incluir los archivos en el folder correspondiente
- Emplear el comando `dvc add` para agregar uno o más archivos al repositorio DVC
- Usar el comando `dvc push` para confirmar los cambios en el repositorio de DVC. Tener en cuenta que el repositorio está asociado con una cuenta en Google Drive para la que eventualmente podría requerir las credenciales al ejecutar este paso.
- Usar los comandos `git add`, `git commit` y `git push origin main` para agregar los cambios en los archivos dvc, especificar un mensaje descriptivo y confirmar los cambios en el repositorio de la solución y de esta manera hacer disponibles los datos a otro compañero del equipo.

¿Cómo registrar experimentos en el servidor MLFlow?

Para este proceso se debe tener en cuenta que el servidor de MLFlow se encuentra disponible a través de una máquina virtual a través del servicio EC2 de AWS. Esta máquina puede no estar habilitada 100% del tiempo, por lo que se requerirán credenciales de acceso al ambiente de AWS para habilitarla o

hibernarla. También se debe tener en cuenta que la IP de esta máquina se renueva tras cada inicialización.

Para registrar resultados de nuevos experimentos se deben realizar los siguientes pasos:

- Identificar los datos, hiperparámetros y algoritmo a usar para entrenar el modelo y definir un diccionario dentro de un script de python.
- Al final del script usar la función `train_model` expuesta en el script `src\models\common\training.py`, especificando el modelo, los datos, parámetros y métricas del experimento. Esta función se encargará de entrenar el modelo, haciendo la partición necesaria de entrenamiento y prueba para el entrenamiento y evaluación del desempeño del modelo y los registrará en el servidor de MLFlow deseado.
- Ajustar el script `training.py` para indicar la IP del servidor de MLFlow que se desea usar para registrar los experimentos, en la instrucción de la línea 25:

```
mlflow.set_tracking_uri("http://3.81.159.3:5000/")
```

- Ejecutar el script creado inicialmente.
- En la consola de su entorno de desarrollo se podrá observar el avance en la ejecución de experimentos.
- Una vez finalizada la ejecución de experimentos es posible acceder al servidor de MLFlow para observar los resultados registrados. Cada modelo registra automáticamente el nombre del algoritmo usado en el modelo, y asigna un identificador incremental para poder identificar, filtrar u ordenar de una manera más sencilla los últimos experimentos registrados.

¿Cómo desplegar nuevas versiones del modelo EPA_ModeloLectura?

- Implementar los ajustes del modelo en `src\models\package\model\predict.py`.
- Si los cambios incluyen nuevos parámetros considere realizar cambios en los archivos `config.yml`, `pipeline.py` y `core.py` según sea el caso que se encuentran en `src\models\package`.
- Agregar nuevas pruebas unitarias en `src\models\package\tests` según sea necesario.
- Incrementar el número de la versión en el archivo `src\models\package\model\VERSION`.
- En una consola de línea de comandos, ubicarse en la ruta `src\models\package` y ejecutar el comando `tox run -e train` para crear un ambiente virtual de tox en el que se entrene el modelo con las dependencias requeridas.
- Si el entrenamiento es exitoso, ejecutar el comando `tox run -e test_package`, de manera que se prepare un ambiente virtual con todos los requerimientos para la ejecución de las pruebas y se obtenga los resultados de dichas pruebas.
- Si todas las pruebas se ejecutan de manera exitosa se procede a construir el paquete con el comando `python3 -m build`. Esto generará un archivo `.whl` en la ruta `src\models\package\dist`, el cual permitirá distribuir el modelo para ser usado por otros usuarios desarrolladores o incluirlo dentro de otras aplicaciones como por ejemplo, un API, para realizar nuevas predicciones con la nueva versión del modelo.

¿Cómo desplegar cambios en el API?

Este componente de la solución se encuentra en: `src\api` y su responsabilidad es responder a 2 necesidades: Proveer datos para el dashboard del contexto de análisis descriptivo y permitir la interacción de un usuario especializado u otras aplicaciones con el modelo predictivo. Tenga en cuenta que el despliegue del componente se realiza a través de un contenedor Docker y por tanto,

eventualmente se requerirán credenciales de acceso a la máquina virtual en Azure para realizar este procedimiento:

- Si el cambio está relacionado con el contexto de análisis descriptivo, pueden incluirse los cambios según sea necesario.
- Si el cambio corresponde a una nueva versión del modelo copie el archivo .whl con el paquete del modelo en la ruta `src\api\epa_modelolectura_api\model-pkg`
- Luego realizar ajustes en `src\api\epa_modelolectura_api\app\schemas\predict.py` y (o) en `src\api\epa_modelolectura_api\app\api.py`, según el cambio en el modelo, así como en el archivo Dockerfile.
- Ejecutar el API y realizar pruebas en el entorno de desarrollo
- Una vez las pruebas hayan sido exitosas, en una consola de línea de comandos, ubicarse en `src\api` y ejecutar el comando `docker build -t epa_modelolectura_api .`, para construir la imagen que incorpore los nuevos cambios.
- Luego ejecutar el comando `docker run -p 8000:8000 -it -e PORT=8000 epa_modelolectura_api`. Esto ejecutará el api en un contenedor local usando Docker Desktop para windows
- Si las pruebas son exitosas en el contenedor local, subir los cambios al repositorio remoto
- Iniciar sesión en la máquina virtual y descargar la última versión
- Ejecutar en la máquina virtual los comandos docker mencionados anteriormente
- Verificar que el api se esté ejecutando exitosamente a través de: <http://172.173.177.148:8000/docs>

¿Cómo desplegar cambios en el componente Dashboard de PowerBI?

Este componente de la solución se encuentra en: `src\visualization\` y su responsabilidad es facilitar el análisis descriptivo a través de visualizaciones. Para incorporar cambios en este componente se requiere:

- Usar PowerBI desktop para realizar los ajustes según sea necesario
- Copiar el archivo .pbix en la ruta `src\visualization\`
- Subir los cambios al repositorio de la solución en GitHub.
- Publicar la nueva versión del tablero en el servidor remoto de PowerBI
- Verificar en la aplicación FrontEnd que los cambios en el tablero se vean reflejados.

¿Cómo desplegar cambios en la aplicación FrontEnd?

Este componente de la solución se encuentra en: `\dashboard` y su responsabilidad es embeber el componente de visualizaciones creado en PowerBI y permitir la interacción de un usuario final con el modelo predictivo. Tenga en cuenta que el despliegue del componente se realiza a través de un contenedor Docker y por tanto, eventualmente se requerirán credenciales de acceso a la máquina virtual en Azure para realizar este procedimiento:

- Realizar el o los ajustes de acuerdo a la necesidad del cambio.
- Si el cambio corresponde a cambios en el modelo predictivo, considerar adaptar la interfaz de usuario para solicitar nuevos parámetros o remover aquellos que hayan quedado obsoletos.
- Ejecutar la aplicación y realizar pruebas en el entorno de desarrollo
- Una vez las pruebas hayan sido exitosas, en una consola de línea de comandos, ubicarse en `\dashboard` y ejecutar el comando `ng build -prod`, para construir una versión de la aplicación que pueda ser publicada en un ambiente productivo.
- Subir los cambios al repositorio remoto

- Iniciar sesión en la máquina virtual y descargar la última versión
- Luego ejecutar el comando `docker build -t miad_dsa_ui .` para construir la imagen que incorpore los nuevos cambios.
- Luego ejecutar el comando `docker run -d -p 8080:80 miad_dsa_ui`. Esto ejecutará la aplicación FrontEnd
- Verificar que la aplicación se esté ejecutando exitosamente a través de <http://172.173.177.148:8001/>, que los cambios se hayan incorporado y que otras funcionalidades no se hayan visto afectadas por el cambio.