



Universidad
Nacional
de Loja

UNIVERSIDAD NACIONAL DE LOJA

“Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables”

Carrera de Ingeniería en Sistemas.

MATERIA : Inteligencia Artificial.

TEMA : Aprendizaje Supervisado: Predicción de los meses con mayor índice de incidentes de tránsito cometidos en Ecuador para el 2022.

CICLO : 9no “B”

DOCENTE : Ing. Luis Antonio Chamba Eras.

INTEGRANTES : Danny Michael Jaramillo Jumbo
Edmundo José Pezantes Urrego.

LOJA – ECUADOR

2022

Título

Aprendizaje Supervisado: Predicción de los meses con mayor índice de incidentes de tránsito cometidos en Ecuador para el 2022.

1. Problema

- Los índices de alertas recibidas por el ECU 911 presentan un incremento considerable con el pasar de los meses, lo cual representa un aumento en el número de emergencias atendidas demostrando que los índices de violencia y emergencias se incrementan con el pasar del tiempo.

1.1 Público Objetivo.

- Cruz Roja



- Agencia de Tránsito



- Policía Nacional



- **Servicio Integrado de Seguridad ECU 911**



1.2 Datos disponibles

- Dataset que se encuentra en el Portal de Datos Abiertos Ecuador
- Estadísticas proporcionadas por parte del Servicio Integrado de Seguridad ECU:911

1.3 Tipo de problema

Aprendizaje Supervisado

1.4 Pregunta de investigación

¿Cuáles serán los meses en que registraran la mayor cantidad de incidentes de tránsito en el año 2022 en el Ecuador, de acuerdo a los Registros del ECU 911?

2. Recopilar datos

2.1 Datos disponibles

- Conjunto de datos utilizados para el aprendizaje automático.

<https://pdae.datasketch.co/dataset/estadisticas-de-emergencias-servicio-integrado-de-seguridad-ecu-911/resource/f28ae8fb-17d6-437a-9e80-96da8d889861>

The screenshot shows the 'Datos Abiertos' portal interface. At the top, there is a navigation bar with links: Inicio, Catálogo de Datos Abiertos, Blog / Noticias, Normativa, Centro de aprendizaje, and Partida. Below the navigation bar, there is a banner with the text 'Explora los datos que se han generado durante la pandemia haciendo clic aquí'. The main content area displays the dataset 'Estadísticas de Emergencias - Servicio Integrado de Seguridad ECU 911'. It includes a 'Volver al catálogo' link, a 'Formatos disponibles' section with a table listing available formats (CSV, XLSX), and a 'Metadatos' section with details about the dataset's source, page, update date, creation date, and license. The 'Archivos para descargar' section lists two files: 'Estadísticas de emergencias - enero 2016' and 'Estadísticas de emergencias - febrero 2016', both in CSV format.

- Estadísticas Posteriores a los años evaluados Registrados por el ECU911, de esta manera se evaluará el modelo.

<https://www.ecu911.gob.ec/wp-content/uploads/2017/06/Estadisticas.pdf>

3. Preparar Datos.

3.1 Limpieza del dataset

En el dataset se procedió a realizar una limpieza de la misma y modificar el formato de algunos de sus campos, además se eliminaron datos que no aportan a la pregunta de investigación, para lo cual se escogieron las variables necesarias para poder cumplir con el objetivo planteado.

- **Base de Datos Original.**

df.head(20)

Index	CENTRO	AÑO	MES	INCIDENTE POLICIAL (TOTAL)	INCIDENTE TRÁNSITO (TOTAL)	INCIDENTE SALUD (TOTAL)	INCIDENTE SERVICIOS MUNICIPALES (TOTAL)	INCIDENTE BOMBEROS (TOTAL)	INCIDENTE GESTION DE RIESGO (TOTAL)
0	SAMBORONDÓN	2012.0	FEBRERO	17.664	2.602		2.762	3.439	690.0
1	SAMBORONDÓN	2012.0	MARZO	21.836	2.945		3.588	3.171	700.0
2	SAMBORONDÓN	2012.0	ABRIL	20.855999999999998	3.161		3.114	2.541	730.0
3	SAMBORONDÓN	2012.0	MAYO	24.164	3.736		2.8360000000000003	2.213	610.0
4	SAMBORONDÓN	2012.0	JUNIO	23.639	3.819		2.5880000000000003	1.148	611.0
5	SAMBORONDÓN	2012.0	JULIO	25.984	4.800		2.648	1.379	821.0
6	SAMBORONDÓN	2012.0	AGOSTO	23.662	3.128		2.313	767	1.073
7	SAMBORONDÓN	2012.0	SEPTIEMBRE	21.233	3.399		1.993	562	1.188
8	SAMBORONDÓN	2012.0	OCTUBRE	19.95	3.754		2.421	701	1.34
9	SAMBORONDÓN	2012.0	NOVIEMBRE	20.685	3.410		2.289	803	1.465
10	SAMBORONDÓN	2012.0	DICIEMBRE	25.57	4.604		3.1	2.066	1.518
11	TOTAL	NAN	NAN	245.273	39.358		29.642	18.790	10.847999999999999
12	SAMBORONDÓN	2013.0	ENERO	37.598	3.234		2.302	2.285	644.0
13	SAMBORONDÓN	2013.0	FEBRERO	37.823000000000005	3.660		2.674	2.201	489.0
14	SAMBORONDÓN	2013.0	MARZO	42.648999999999994	3.840		3.053	4.144	619.0
15	SAMBORONDÓN	2013.0	ABRIL	40.124	3.552		3.637	2.673	661.0
16	SAMBORONDÓN	2013.0	MAYO	40.168	3.834		3.928	1.669	824.0
17	SAMBORONDÓN	2013.0	JUNIO	40.253	3.748		3.85	1.089	764.0
18	SAMBORONDÓN	2013.0	JULIO	36.345	3.667		4.364	993	841.0
19	SAMBORONDÓN	2013.0	AGOSTO	34.37	3.628		6.165	1.114	1.357

Show 25 per page
Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

- **Base de Datos Resultante**

	CENTRO	AÑO	MES	INCIDENTE POLICIAL (TOTAL)	INCIDENTE TRÁNSITO (TOTAL)	INCIDENTE SALUD (TOTAL)	INCIDENTE SERVICIOS MUNICIPALES (TOTAL)	INCIDENTE BOMBEROS (TOTAL)	INCIDENTE GESTIÓN DE RIESGO (TOTAL)	INCIDENTE MILITAR (TOTAL)	TOTAL DE INCIDENTES RECEPTOS	INCIDENTE CLAVES ROJAS (TOTAL)	DE LLAMADAS MALINTENDIDAS O INDEBIDAS (TOTAL)	CANTIDAD DE TELEFONOS ENVIADOS A SANCIÓN	DE LLAMADAS GENERADAS POR LOS TELEFONOS SANCIONADOS	DE LLAMADAS NORMALES (TOTAL)	TOTAL DE LLAMADAS RECIBIDAS (TOTAL)
0	SAMBORONDÓN	2012	FEBRERO	17.684	2.602	2.762	3.439	690.000	327	3	27.507	13.920	116.863	578	15.762	29.609	349.355
1	SAMBORONDÓN	2012	MARZO	21.836	2.945	3.568	3.171	700.000	219	18	32.457	16.425	74.615	631	16.405	34.353	375.656
2	SAMBORONDÓN	2012	ABRIL	20.856	3.161	3.114	2.541	730.000	34	53	30.489	15.429	312.842	2.922	85.093	33.360	412.333
3	SAMBORONDÓN	2012	MAYO	24.164	3.726	2.836	2.213	610.000	15	15	33.589	16.997	360.486	3.362	89.715	40.473	429.378
4	SAMBORONDÓN	2012	JUNIO	23.629	3.619	2.598	1.148	611.000	10	12	31.837	16.111	355.638	3.258	84.625	37.254	418.794
5	SAMBORONDÓN	2012	JULIO	26.904	4.800	2.648	1.379	921.000	7	13	35.762	18.097	390.133	18.324	172.417	42.981	453.890
6	SAMBORONDÓN	2012	AGOSTO	23.662	3.128	2.313	767	1.075	1	25	30.971	15.673	351.363	16.762	153.684	37.170	410.744
7	SAMBORONDÓN	2012	SEPTIEMBRE	21.233	3.399	1.993	562	1.188	5	14	28.394	14.368	340.861	15.756	145.163	34.499	402.063
8	SAMBORONDÓN	2012	OCTUBRE	19.950	3.754	2.421	701	1.340	3	6	28.175	14.258	353.859	14.705	122.895	33.802	415.374
9	SAMBORONDÓN	2012	NOVIEMBRE	20.685	3.410	2.289	803	1.465	2	4	28.658	14.502	345.327	14.112	118.931	34.762	409.200
10	SAMBORONDÓN	2012	DICIEMBRE	25.570	4.604	3.100	2.066	1.518	7	7	36.872	18.659	330.903	14.112	109.746	43.298	407.238
11	SAMBORONDÓN	2013	ENERO	37.598	3.234	2.302	2.285	644.000	12	16	46.091	21.866	395.192	17.156	141.747	52.813	497.490
12	SAMBORONDÓN	2013	FEBRERO	37.623	3.660	2.674	2.201	489.000	45	42	46.734	21.817	374.039	18.364	173.892	54.074	474.771
13	SAMBORONDÓN	2013	MARZO	42.649	3.840	3.653	4.144	619.000	480	46	55.431	26.217	387.290	18.828	185.052	66.318	496.770
14	SAMBORONDÓN	2013	ABRIL	40.124	3.552	3.637	2.673	661.000	29	22	50.698	24.772	383.076	18.652	182.883	60.978	486.169
15	SAMBORONDÓN	2013	MAYO	40.168	3.834	3.928	1.669	824.000	6	24	50.453	25.906	359.244	17.221	158.775	60.049	461.482
16	SAMBORONDÓN	2013	JUNIO	40.253	3.748	3.850	1.089	764.000	2	6	49.712	26.426	299.728	14.189	128.403	59.611	401.490
17	SAMBORONDÓN	2013	JULIO	36.345	3.667	4.364	983	941.000	5	30	46.335	25.338	263.493	11.154	105.187	54.976	360.003
18	SAMBORONDÓN	2013	AGOSTO	34.370	3.628	5.165	1.114	1.357	14	22	45.670	26.465	237.283	8.044	84.023	54.973	333.728
19	SAMBORONDÓN	2013	SEPTIEMBRE	37.540	3.749	5.235	1.437	1.469	16	12	49.458	26.519	231.553	9.194	97.406	60.296	332.759

- **Reclasificación de la Columna INCIDENTE_TRÁNSITO_(TOTAL) según la regla de Sturges.**

$$K=1+3.3 \text{ Log}(N) = 9.78 \text{ (5)}$$

$$R= V_{\max} - V_{\min}$$

$$R= 9990 - 239 = 9751$$

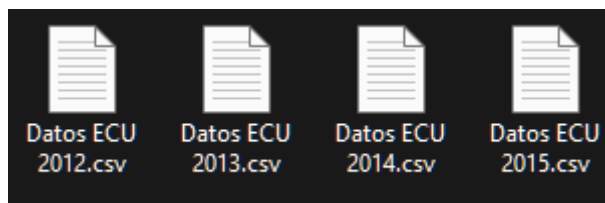
$$A= R/K$$

$$A= 9751/5 = 1950$$

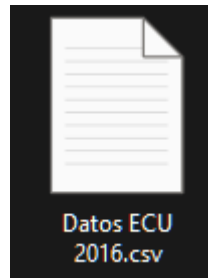
Clase	Amplitud del Intervalo	
Muy Bajo	239	2189
Bajo	2190	4140
Medio	4141	6091
Alto	6092	8042
Muy Alto	8043	9993

4. Dividir Datos

- Los datos se dividieron por año 2012,2013,2014,2015 los mismos que serán utilizados para el aprendizaje Supervisado.



- Los datos del año 2016 serán usados para realizar los Test pertinentes.



5. Modelos de Entrenamiento.

```
X = Data.drop('INCIDENTE_TRÁNSITO',axis =1).values
y = Data['INCIDENTE_TRÁNSITO'].values
```

```
#splitting Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=101)
```

```
# Multiple Liner Regression
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

6. Test

```
y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df1 = df.head(10)
df1
```

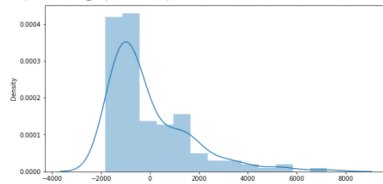
	Actual	Predicted
0	3660.0	2307.814305
1	3749.0	2090.766906
2	985.0	1804.391514
3	1123.0	1680.364430
4	1905.0	1331.975495
5	619.0	1866.405057
6	1053.0	1742.377972
7	888.0	1393.989038
8	832.0	2059.760135
9	2890.0	1997.746593

7. Medir el desempeño

```
# visualizing residuals
fig = plt.figure(figsize=(10,5))
residuals = (y_test - y_pred)
sns.distplot(residuals)

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility to 'distplot') or 'kdeplot' (an axes-level function that only shows the kernel density estimate).
warnings.warn(msg, FutureWarning)

<matplotlib.axes._subplots.AxesSubplot at 0x7fb8db7e990>
```




```
# evaluate the performance of the algorithm (MAE - MSE - RMSE)
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('VarScore:', metrics.explained_variance_score(y_test, y_pred))
```

```
MAE: 1287.1166260600355
MSE: 2735070.721266704
RMSE: 1653.8049223734654
VarScore: 0.024702617881854905
```

8. Implementación del Modelo

8.1 Modelo generado

Como primer punto se genera el archivo pkl del modelo generado en google colab para poder cargarlo al servidor para su producción.

 ml_model_regression.pkl

07/02/2022 20:59

Archivo PKL

1 KB

8.2 Creación de las API

Se procede a crear las API con fastAPI, la cual responde a las entradas de año y mes, ambas de tipo numérico.

```

1 from fastapi import FastAPI, Request
2 from hause_class import Hause
3 import pickle
4
5 app = FastAPI()
6
7 @app.on_event("startup")
8 def load_model():
9     global model
10    model = pickle.load(open("ml_model_regression.pkl", "rb"))
11
12 @app.get("/author")
13 def index():
14     return {
15         "msg": "Machine Learning",
16         "author": "Ing. Danny Jaramillo"
17     }
18
19
20 @app.post("/predict")
21 async def get_home_price(request: Request):
22     formdata = await request.form()
23     hause_attr = [
24         formdata["anio"],
25         formdata["mes"]
26     ]
27     price = model.predict(hause_attr).tolist()[0]
28     return {'Anio': formdata["anio"], 'Mes': formdata["mes"], "prediccion": price}

```

8.3 Consumiendo las API

Una vez creadas las API se procede a crear la función para consumirlas de modo que se pueda extraer la información.

```

const PeticionPut = async (props) => {
    let anio = document.getElementById('anio').value
    let mes = document.getElementById('mes').value
    console.log("añooo" + anio + "mes" + mes)
    let formData = new FormData();
    formData.append('anio', anio);
    formData.append('mes', mes);
    await fetch('/predict',
        {
            method: "POST",
            mode: 'cors',
            headers: {
                'Access-Control-Allow-Origin': 'http://localhost:8080/'
            },
            body: formData
        })
        .then(res => res.json())
        .then(response => { mostrarAlerta(response.prediccion) });
}

```


8.4 Interfaz del Modelo

Se crea un interfaz para el Modelo de Predicción, la cual presenta la petición del ingreso de un año y un mes. Además, presenta un bonton llamado predecir el cual da la posibilidad de realizar la predicción.

The screenshot shows a web interface titled "Predicción de incidentes de Tránsito con Machine Learning ECU911." Below the title, there is a form with two input fields. The first field is labeled "Año" and contains the value "2016". Below this field is a hint text: "Ingresar un año para realizar la Predicción". The second field is labeled "Ingrese un mes" and contains the value "3". Below this field is a hint text: "Ingresar un mes entre el 1 y el 12". At the bottom of the form is a red button labeled "Predecir". The background of the interface features a repeating pattern of the ECU 911 logo.

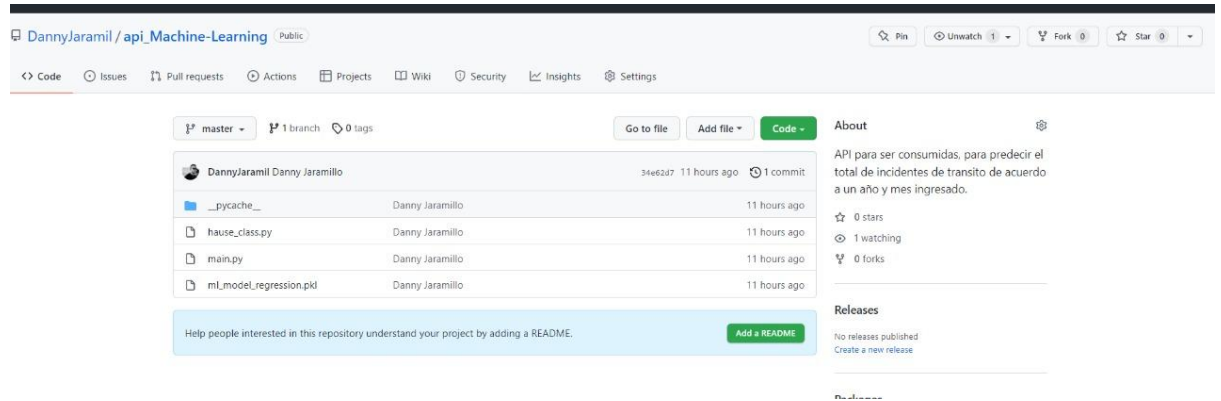
8.5 Interfaz de la predicción

Finalmente se crea una interfaz para mostrar los resultados de la predicción realizada al presionar el botón Predecir después de que se ingrese el año y mes.

The screenshot shows the same prediction interface as in 8.4, but with a modal dialog box displayed in the foreground. The dialog box has a green checkmark icon at the top and the title "Accidentes de transito". Below the title, it displays the text "Existira un total de: 1231.6407302719308". At the bottom right of the dialog box is a blue button labeled "Aceptar". The background of the interface is dimmed, showing the same form and ECU 911 logo pattern as in the previous screenshot.

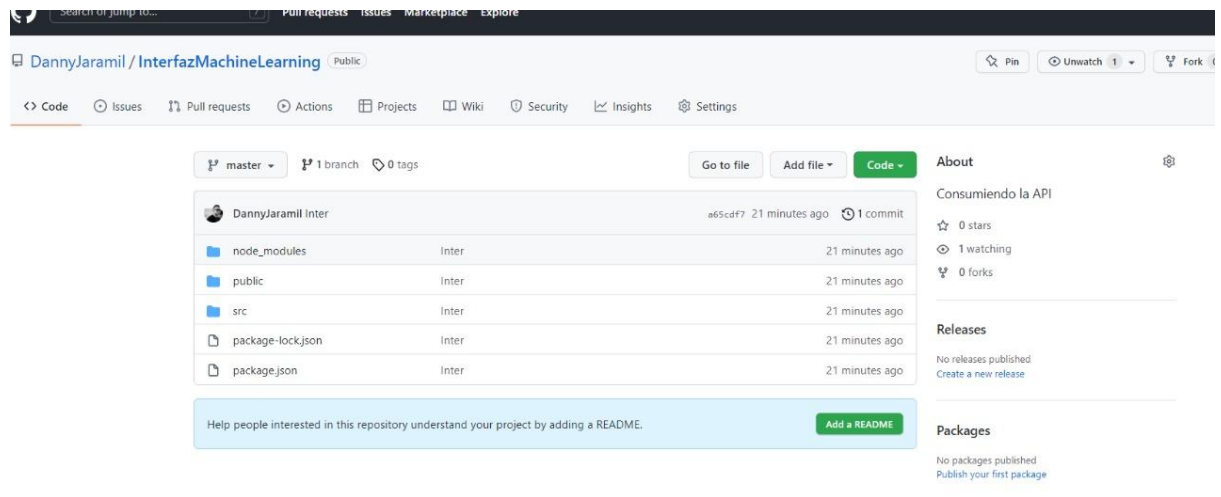
8.6 Enlace git hub del backend

https://github.com/DannyJaramil/api_Machine-Learning.git



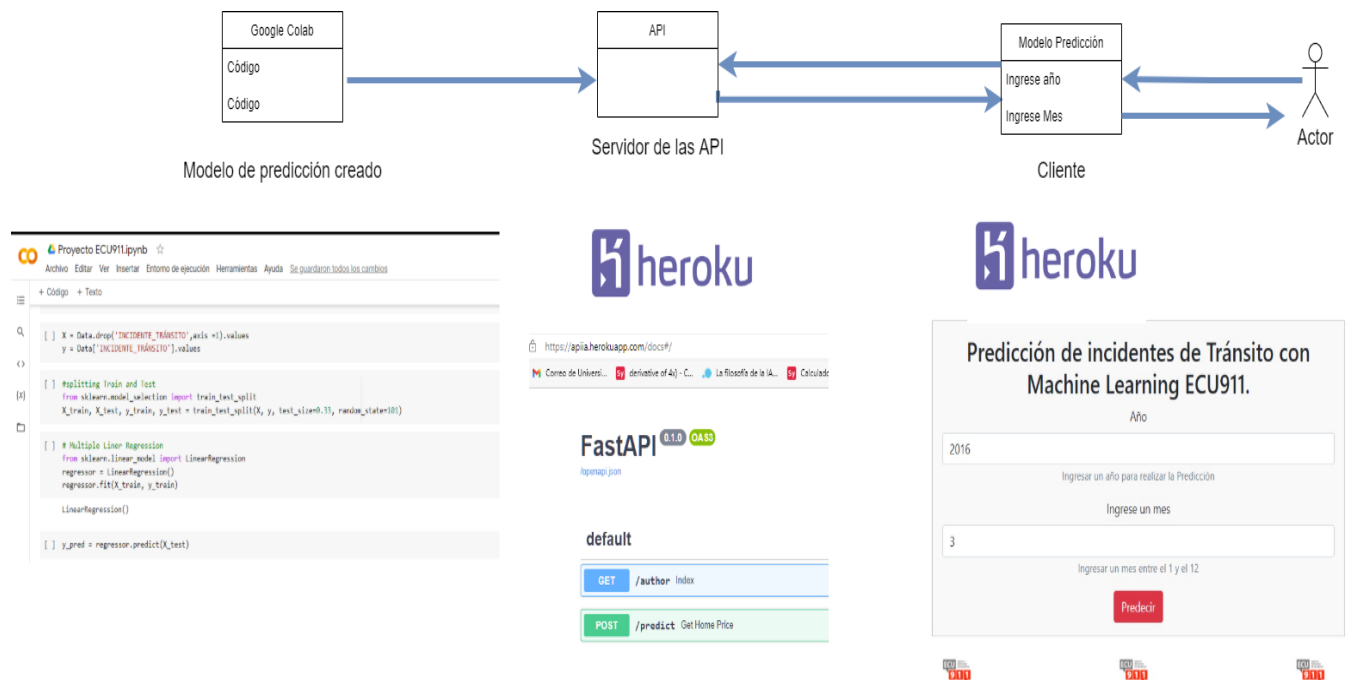
8.7 Enlace git hub del frontend

<https://github.com/DannyJaramil/InterfazMachineLearning>



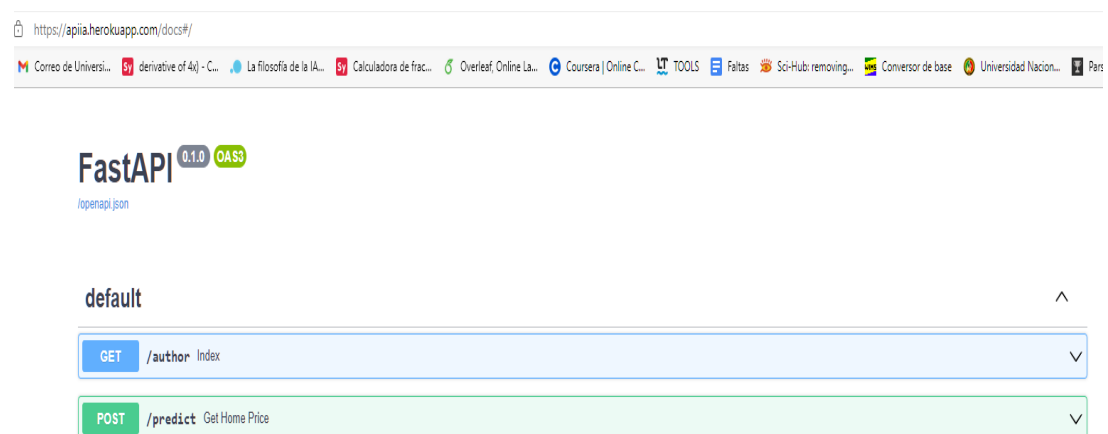
9. Puesta en producción del Modelo

Las API tanto para el backend como para el frontend son creadas en la plataforma Heroku, la cual es una plataforma en la nube que permite a las empresas construir, entregar, supervisar aplicaciones y alojarlas en la nube.



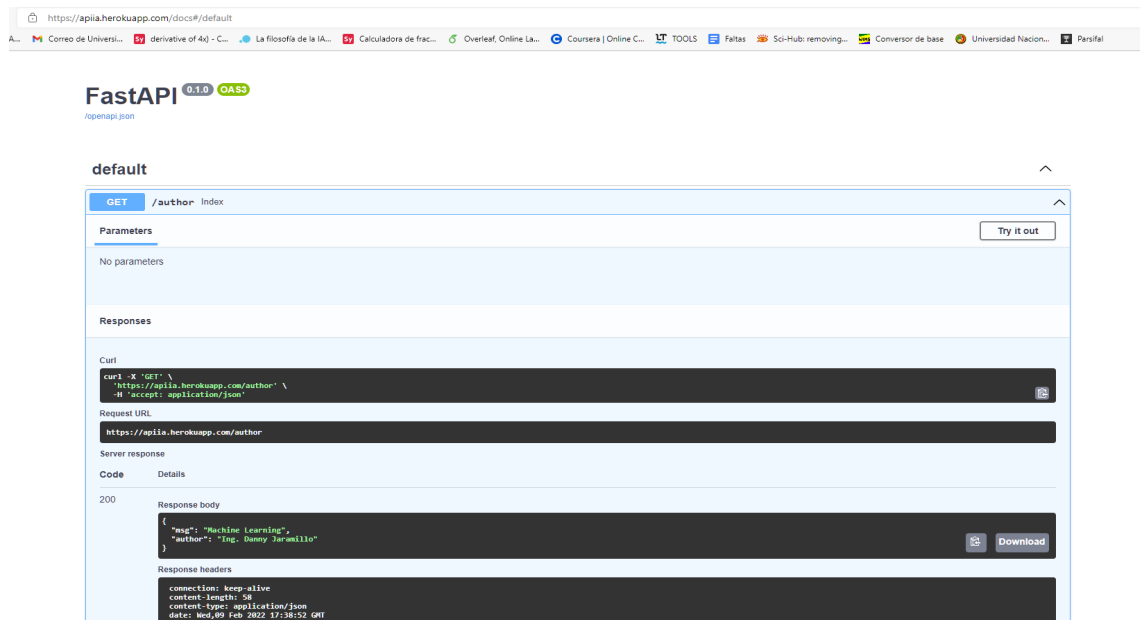
9.1 Creación de las API.

Primeramente se procede a crear las API en la cual la solicitud GET es en la que se pasa los datos como una cadena de consulta y en la solicitud POST envía los datos al cuerpo de la solicitud.



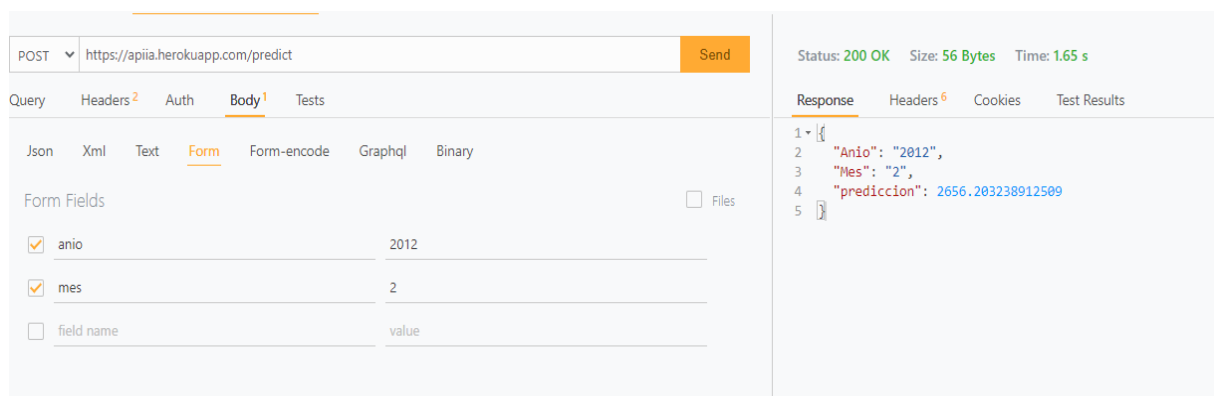
9.2 Puesta de la Api en el Servidor

Una vez creadas la API se procede a cargarlas en el servidor de modo que cuando recibe la petición comprueba el host de donde lo estás pidiendo y si se encuentra en su lista de host permitidos, te envía los datos.



9.3 Verificación de la API en el Servidor

Finalmente se procede a verificar el funcionamiento de las API en el servidor para lo cual se realiza una petición y se verifica que cargue los datos.



9.4 URL de backend en producción

- https://apiia.herokuapp.com/docs#/default/get_home_price_api_predict_post

9.5 URL frontend en producción

- <https://interfas.herokuapp.com>

9.6 Enlace del modelo puesto a producción

A continuación se presenta el enlace del modelo puesto a producción en el cual se carga la interfaz del modelo para realizar las predicciones:

- <https://interfas.herokuapp.com/>

