Calidad del aire en la Comunidad de Madrid

Grupo 7

Alfonso Gallardo

Raúl Hervás

Carmen Reina

Walter Ronceros

Susana Vara

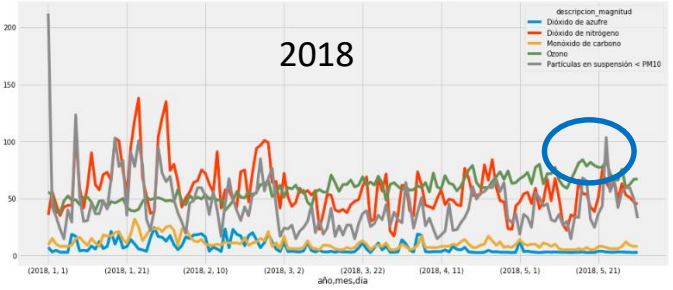
Instituto Tecnológico Telefónica

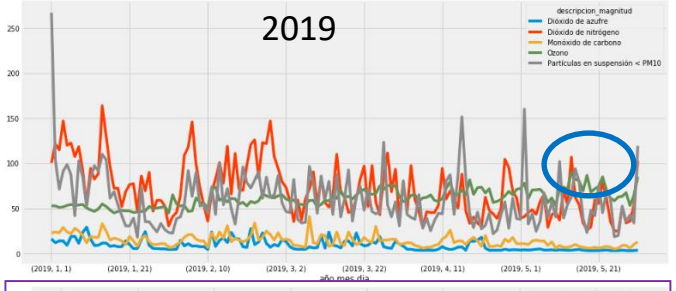

Febrero 2020
Abril 2020

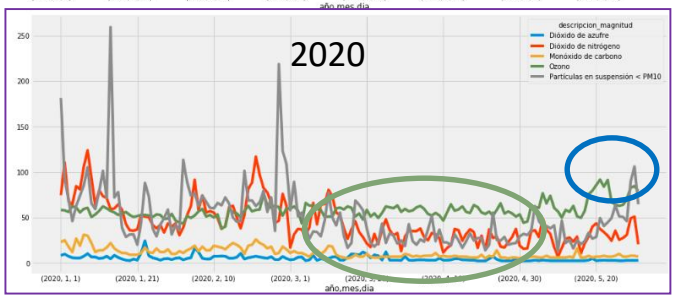
Objetivos del análisis

La salud pública y el medio ambiente son dos de los objetivos principales de los Gobiernos. Estas pueden verse mermada por **altas tasas de contaminantes en el ambiente**. Es necesario por ello su predicción y reducción.

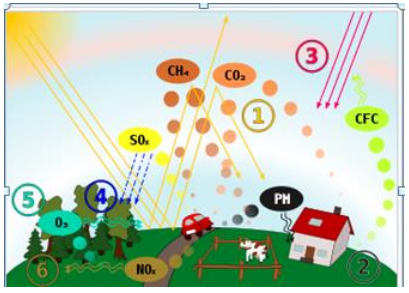
- Impacto en la reducción de los niveles de la contaminación en el confinamiento de la población durante el COVID-19







- Análisis de las magnitudes de contaminantes que generan el índice de calidad del aire (ICA).

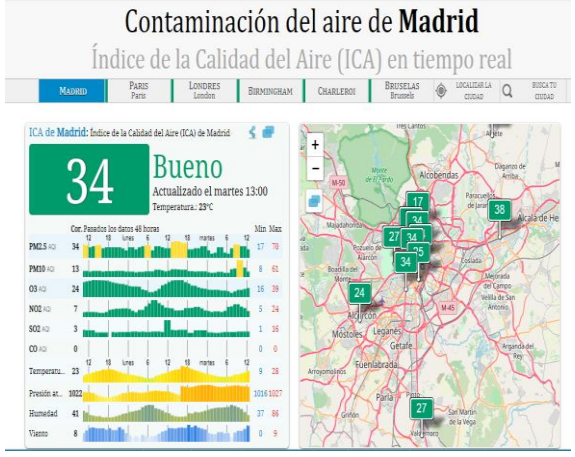


La calidad del aire a nivel mundial se mide mediante un índice denominado **AQI** (Air Quality Index) , en español es **ICA (Índice de la Calidad del Aire)**.

Este índice es el máximo de los valores equivalentes de 5 contaminantes: SO2 (**Dióxido de Azúfre**), NO2 (**Dióxido de Nitrógeno**), CO (**Monóxido de Carbono**) , O3(**Ozono**), PM10 y PM25 (**partículas)** en todas las estaciones de medida de un municipio o región.

- Predicción de dicha contaminación a través de diferentes modelos predictivos y su comparación para identificar cual es el que mejor predice la contaminación atmosférica en Madrid.

Índice	Calidad del Aire	Color Descriptivo
>= 0 y < 50	Buena	Verde
>= 50 y <100	Admisible	Amarillo
>=100 y <150	Mala	Rojo
>= 150	Muy Mala	Marrón



Fuentes de datos

Datos tanto de contaminantes como metereológicos de los portales de datos abiertos de la comunidad de Madrid y del ayuntamiento de Madrid.

- Datos calidad del aire y metereologicos:

- Formato de los dataset:

Datos horarios y en tiempo real:

Cada registro está estructurado de la siguiente forma:

PROVINCIA	MUNICIPIO	ESTACION	MAGNITUD	PUNTO_MUESTREO	ANO	MES	DIA	H01	V01	H02	V02
28	79	104	82	28079104_82_98	2019	1	1	23	V	17	V

- Magnitudes de contaminantes con sus factores de conversión para calcular el ICA **(Índice de Calidad del Aire)** Parcial:

codigo_magnitud	descripcion_magnitud	codigo_tecnica_de_medida	descripcion_tecnica_de_medida	unidad	descipcion_unidad	valor_limite_diario	factor_calculo_diario	valor_limite_horario	factor_calculo_horario
1	**Dióxido de azufre**	38	?g/m³	micrograms por metro cúbico	SO2	125	0.800	350	**0.286**
6	**Monóxido de carbono**	48	mg/m³	miligramos por metro cúbico	CO	10	10	10	**10**
7	Monóxido de nitrógeno	8	?g/m³	micrograms por metro cúbico	NO				
8	**Dióxido de nitrógeno**	8	?g/m³	micrograms por metro cúbico	NO2			200	**0.500**
9	Partículas en suspensión < PM2,5	49	?g/m³	micrograms por metro cúbico	PM2,5				
10	**Partículas en suspensión < PM10**	49	?g/m³	micrograms por metro cúbico	PM10	50	2	150	**0.667**
12	Óxidos de nitrógeno	8	?g/m³	micrograms por metro cúbico	NOX				
14	**Ozono**	6	?g/m³	micrograms por metro cúbico	O3	120	0.833	180	**0.556**
20	Tolueno	59	?g/m³	micrograms por metro cúbico	TOL				
30	Benceno	59	?g/m³	micrograms por metro cúbico	BEN				
35	Etilbenceno	59	?g/m³	micrograms por metro cúbico	EBE				
37	Metaxileno	59	?g/m³	micrograms por metro cúbico	MXY				
38	Paraxileno	59	?g/m³	micrograms por metro cúbico	PXY				
39	Ortoxileno	59	?g/m³	micrograms por metro cúbico	OXY				
42	Hidrocarburos totales				TCH				
43	Metano	2	mg/m³	miligramos por metro cúbico	CH4				
44	Hidrocarburos no metánicos	2	mg/m³	miligramos por metro cúbico	NMHC				
22	Black Carbon	7	?g/m³	micrograms por metro cúbico					
431	MetaParaXileno	59	?g/m³	micrograms por metro cúbico					

- Magnitudes metereologicas:

CÓDIGO MAGNITUD	DESCRIPCIÓN MAGNITUD	CODIGO DE TÉCNICA DE MEDIDA	UNIDAD	DESCRIPCIÓN UNIDAD
81	**Velocidad del viento**	89	m/s	metros por segundo
82	**Dirección del viento**	89	Grd	grados
83	**Temperatura**	89	ºc	grados centígrados
86	**Humedad relativa**	89	%	porcentaje
87	**Presión atmosférica**	89	mbar	milibar
88	**Radiación solar**	89	W/m2	vatios por metro cuadrado
89	**Precipitación**	89	l/m2	litros por metro cuadrado

- Estaciones de medida:

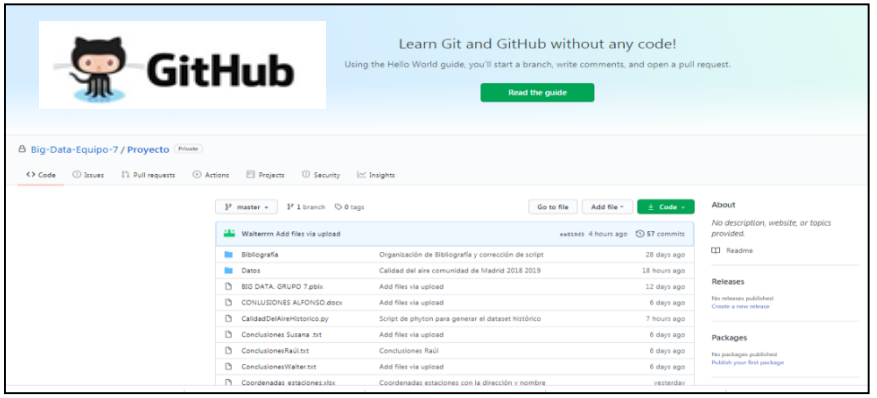
CÓDIGO NACIONAL	CÓDIGO MUNICIPIO	NOMBRE ESTACIÓN
28005002	5	ALCALÁ DE HENARES
28006004	6	ALCOBENDAS
28007004	7	ALCORCÓN
28009001	9	ALGETE
28013002	13	ARANJUEZ
28014002	14	ARGANDA DEL REY
28016001	16	EL ATAZAR
28045002	45	COLMENAR VIEJO
28047002	47	COLLADO VILLALBA
28049003	49	COSLADA
28058004	58	FUENLABRADA
28065014	65	GETAFE
28067001	67	GUADALIX DE LA SIERRA
28074007	74	LEGANÉS
28080003	80	MAJADAHONDA
28092005	92	MOSTOLES
28102001	102	ORUSCO DE TAJUÑA
28120001	120	PUERTO DE COTOS
28123002	123	RIVAS-VACIAMADRID
28133002	133	SAN MARTÍN DE VALDEIGLESIAS
28148004	148	TORREJÓN DE ARDOZ
28161001	161	VALDEMORO
28171001	171	VILLA DEL PRADO
28180001	180	VILLAREJO DE SALVANÉS

ANEXO I. CÓDIGOS DE ESTACIONES

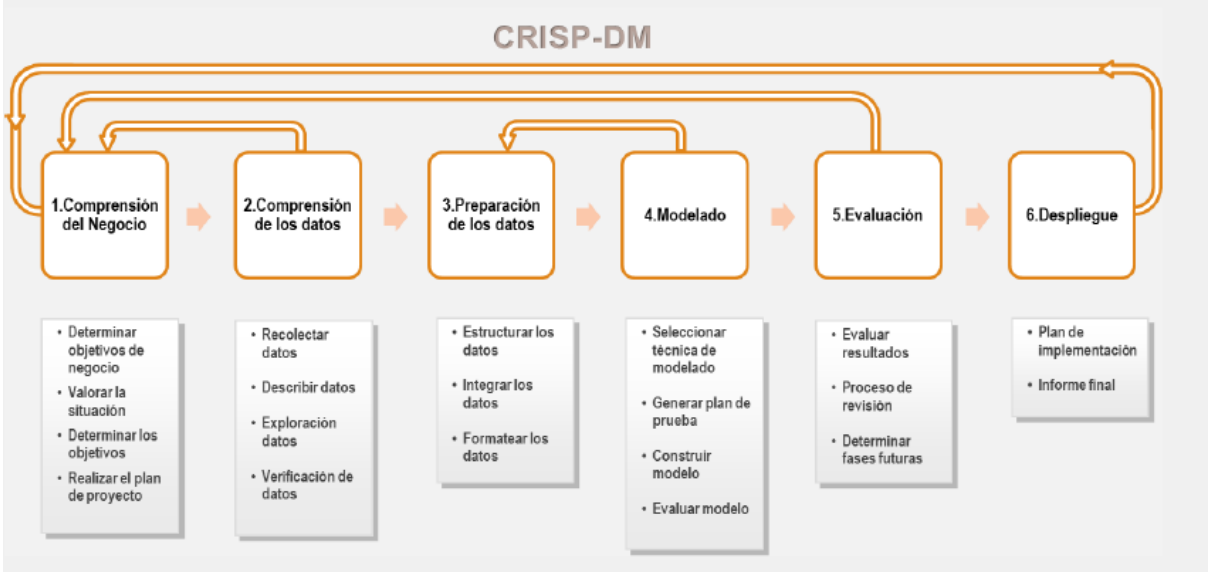
CÓDIGO	ESTACIÓN
28079102	J.M.D. Moratalaz
28079103	J.M.D. Villaverde
28079104	E.D.A.R. La China
28079106	Centro Mpal. De Acústica
28079107	J.M.D. Hortaleza
28079108	Peñagrande
28079109	J.M.D.Chamberí
28079110	J.M.D.Centro
28079111	J.M.D.Chamartín
28079112	J.M.D.Vallecas 1
28079113	J.M.D.Vallecas 2
28079114	Matadero 01
28079115	Matadero 02
28079004	Plaza España
28079008	Escuelas Aguirre
28079016	Arturo Soria
28079018	Farolillo
28079024	Casa de Campo
28079035	Plaza del Carmen
28079036	Moratalaz
28079038	Cuatro Caminos
28079039	Barrio del Pilar
28079054	Ensanche de Vallecas
28079056	Plaza Elíptica
28079058	El Pardo
28079059	Juan Carlos I

Metodología empleada y espacio colaborativo

- Coordinación del equipo mediante Trello y Github:



- Siguiendo las fases de la metodología CRIP-DM:



Roles

SQUAD

ANALISTA DE DATOS

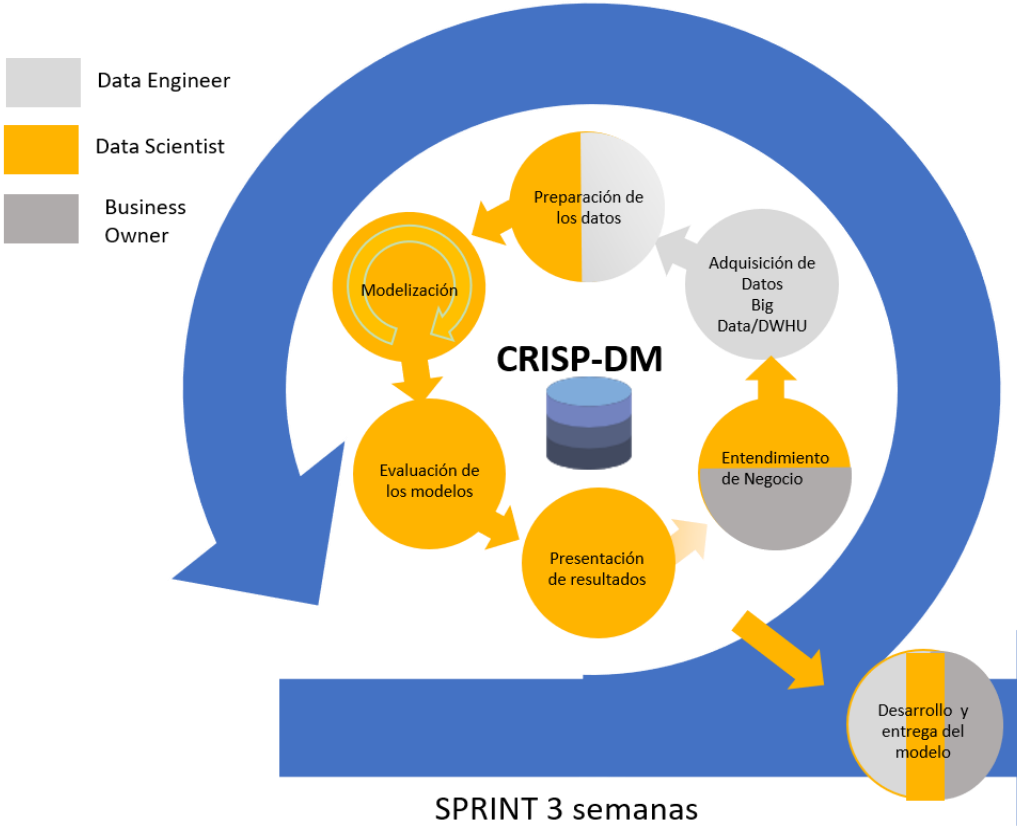
INGENIERO DE DATOS

CIENTIFICO DE DATOS

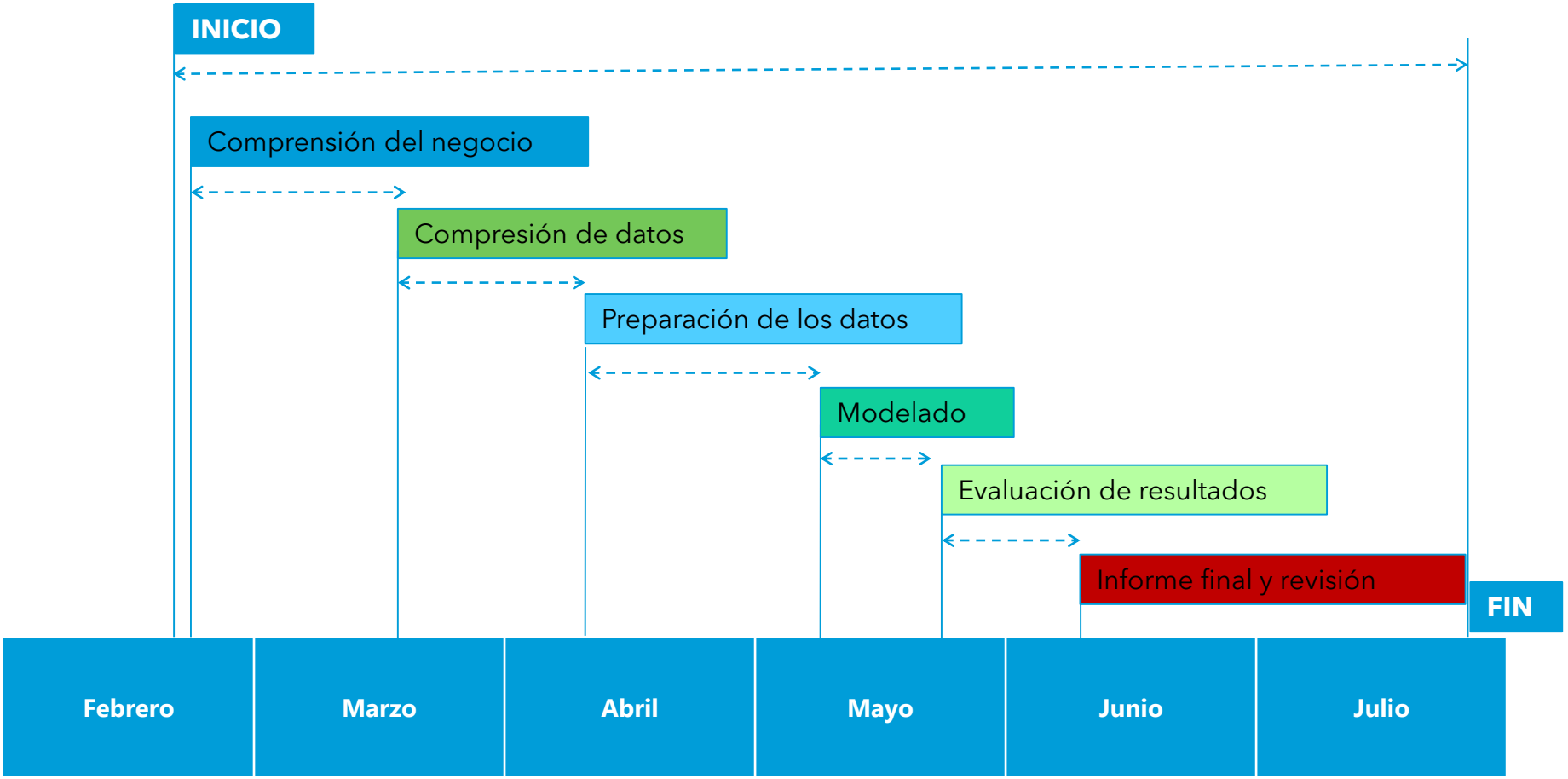
ARQUITECTO DE DATOS

ANALISTA DE NEGOCIO

AGILE + **CRISP-DM** = **AGILE DATA SCIENCE**



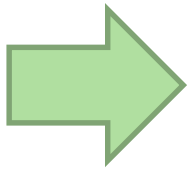
Timeline

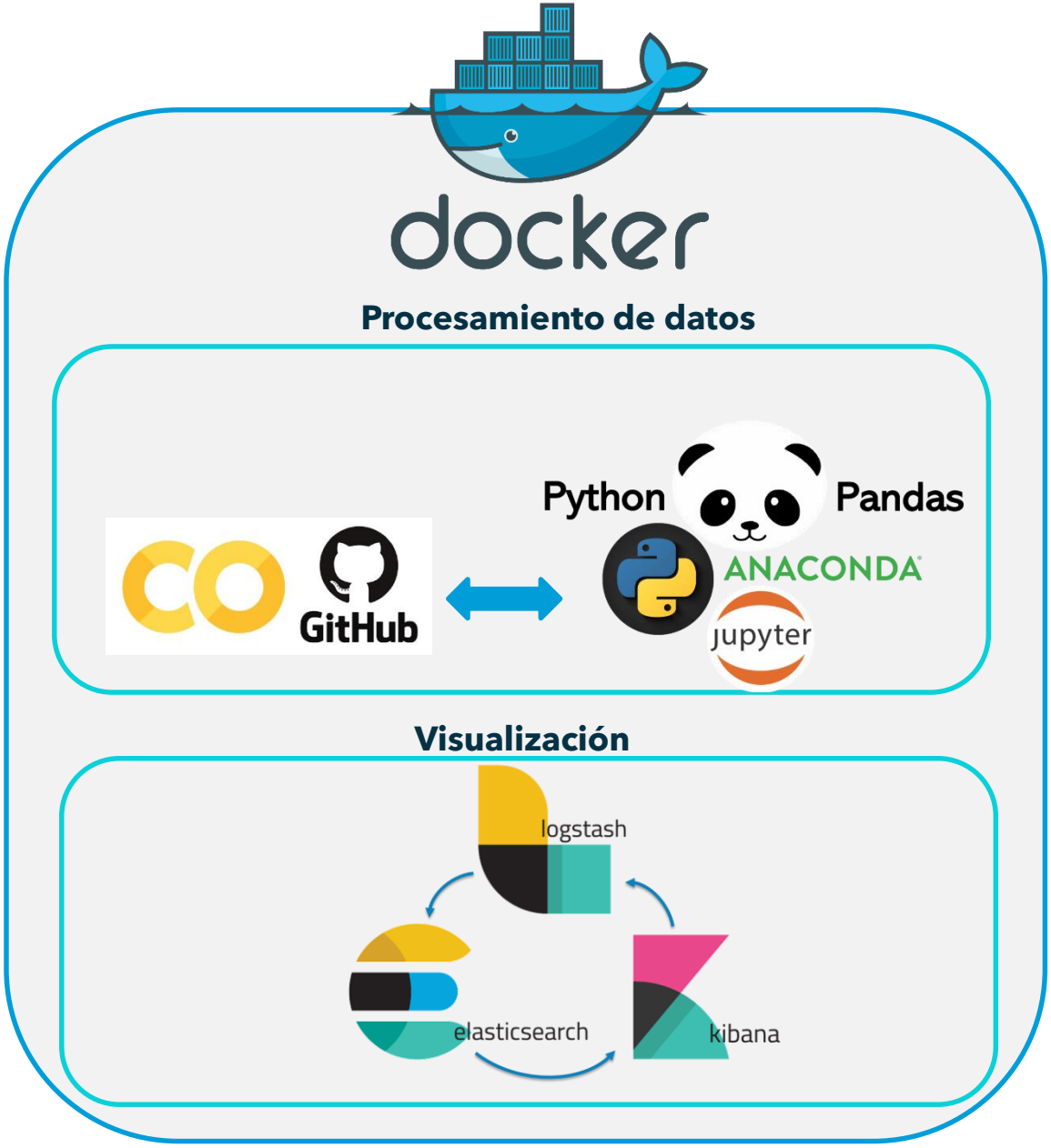


TIMELINE

Arquitectura

datos abiertos





Preparación de los datos

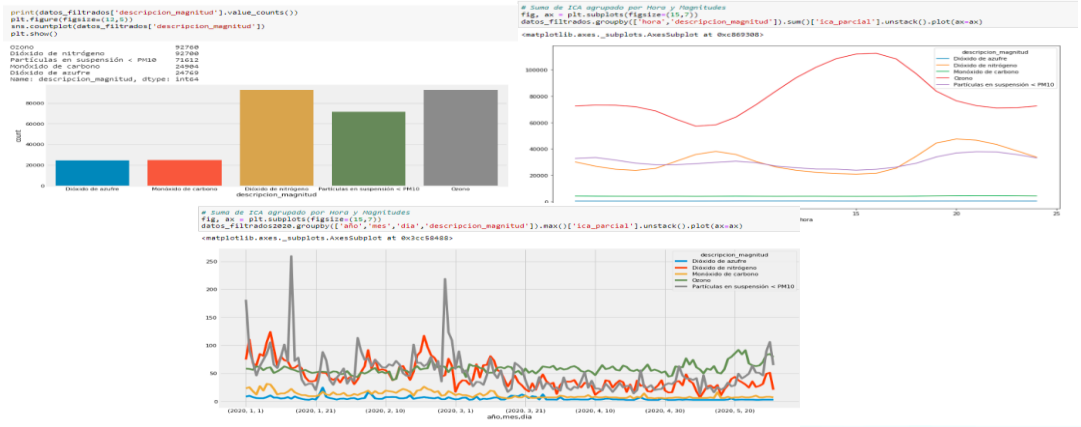
- Preparación de los dataset con **phyton**:

Los datos originales hay que tratarlos para calcular el valor del Índice de calidad del aire (ICA) de cada contaminante en cada estación de medida.

```
#Concatenamos Datos Calidad del aire del Ayuntamiento de Madrid y Comunidad de Madrid para unificarlos.
datosCA = pd.concat([datosCACM, datosCAAM ], sort=True)

#Creamos Estacion Real
datosCA['estacion_real'] = datosCA['punto_muestreo'].str[:8]
#datosCA['estacion_real'] = datosCA.punto_muestreo.apply(lambda x: x[:8])

#Creamos Fecha
datosCA['fecha'] =  datosCA.ano.apply(str) + '-' + datosCA.mes.apply(str) + '-' + datosCA.dia.apply(str)

#Obtenemos Magnitudes Calidad del Aire
urlMagnitudesCA = '/Datos/Magnitudes Calidad del Aire.csv'
magnitudesCA = pd.read_csv(urlMagnitudesCA, sep=";")

#Exportar a CSV fichero de calidad de aire
datosCA.to_csv('datosCaliadAire.csv', sep=";")

#Hacemos el merge para introducir los valores descriptivos de cada magnitud, así como los valores límite y factor de cálcul
mergeCA = datosCA.merge(magnitudesCA, left_on='magnitud', right_on='codigo_magnitud')

#mergeCA["factor_calculo_horario"] = pd.to_numeric(mergeCA["factor_calculo_horario"], downcast="float")
#mergeCA["h01"] = pd.to_numeric(mergeCA["h01"], downcast="float")

mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h01'] = mergeCA['factor_calculo_horario'] * mergeCA['h01']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h02'] = mergeCA['factor_calculo_horario'] * mergeCA['h02']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h03'] = mergeCA['factor_calculo_horario'] * mergeCA['h03']
mergeCA.loc[mergeCA['factor_calculo_horario'].notnull(), 'ica_h04'] = mergeCA['factor_calculo_horario'] * mergeCA['h04']
```

- El pre-procesamiento y análisis lo realizamos con **colab de google** y **jupyter notebook**:

Analizamos las variables de calidad del aire, las metereológicas y el Índice de Calidad del Aire (ICA)



Instituto Tecnológico Telefónica

Preparación de los datos

- Preparación de los datos para el modelado, normalización de las variables y del índice de calidad de aire (ICA) que será nuestra variable a predecir. Observamos los datos atípicos y la matriz de correlaciones

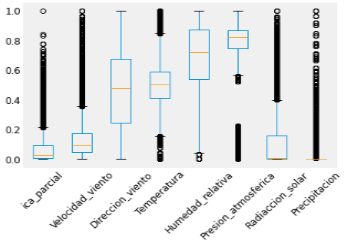
```
#datos_normalizados.describe()
datos_normalizados.dtypes
```

```
Fecha                object
ica_parcial         float64
Velocidad_viento    float64
Direccion_viento    float64
Temperatura         float64
Humedad_relativa    float64
Presion_atmosferica float64
Radiaccion_solar    float64
Precipitacion       float64
dtype: object
```

```
# Correlacion entre variables
datos_normalizados.corr()
```

	ica_parcial	Velocidad_viento	Direccion_viento	Temperatura	Humedad_relativa	Presion_atmosferica	Radiaccion_solar	Precipitacion
ica_parcial	1.000000	-0.005117	0.029837	0.066666	-0.125869	-0.017760	0.043789	-0.003836
Velocidad_viento	-0.005117	1.000000	0.077269	0.067277	-0.179103	-0.172627	0.176901	0.029434
Direccion_viento	0.029837	0.077269	1.000000	0.156068	-0.051785	-0.020500	0.034052	-0.020661
Temperatura	0.066666	0.067277	0.156068	1.000000	-0.544084	0.210312	0.466563	-0.053249
Humedad_relativa	-0.125869	-0.179103	-0.051785	-0.544084	1.000000	-0.045112	-0.476870	0.137374
Presion_atmosferica	-0.017760	-0.172627	-0.020500	0.210312	-0.045112	1.000000	-0.026128	-0.109467
Radiaccion_solar	0.043789	0.176901	0.034052	0.466563	-0.476870	-0.026128	1.000000	-0.057845
Precipitacion	-0.003836	0.029434	-0.020661	-0.053249	0.137374	-0.109467	-0.057845	1.000000

```
### Consultamos diagramas box y whisker (cajas y bigotes)
boxplot = datos_normalizados.boxplot(grid=False, rot=45, fontsize=12)
```




```
#correlacion con datos
f,ax = plt.subplots(figsize=(9, 9))
sns.heatmap(datos_normalizados.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```

- Preparamos el dataframe para estudiar con los algoritmos:

- Creamos el Índice de calidad del aire binario (0/1) para aplicar los modelos

```
# Preparamos dataframe
datos_normalizados = datos_normalizados.drop(['Fecha'], axis=1) # eliminamos fecha
datos_normalizados.insert(1, "ica", datos_filtrados["ica"]) # incluimos ica binario >60 1
datos_normalizados.head(5)
```

	ica	ica_parcial	Velocidad_viento	Direccion_viento	Temperatura	Humedad_relativa	Presion_atmosferica	Radiaccion_solar	Precipitacion
0	1	0.000000	0.079137	0.041667	0.267241	0.849462	0.697872	0.008362	0.0
1	1	0.010472	0.079137	0.041667	0.267241	0.849462	0.697872	0.008362	0.0
2	1	0.014330	0.079137	0.041667	0.267241	0.849462	0.697872	0.008362	0.0
3	1	0.019485	0.079137	0.041667	0.267241	0.849462	0.697872	0.008362	0.0
4	1	0.082561	0.079137	0.041667	0.267241	0.849462	0.697872	0.008362	0.0

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import AdaBoostClassifier, GradientBoostingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score

X = datos_normalizados.drop('ica', axis=1)
y = datos_normalizados.ica

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
```

Modelado. Algoritmos

K-NN

```
knMod = KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,
                             metric='minkowski', metric_params=None)
```

```
knMod.fit(X_train, y_train)
knMod.score(X_test, y_test)
```

```
0.9963227144105259
```

Regresión logística

```
glmMod = LogisticRegression(penalty='l1', dual=False, tol=0.0001, C=1.0, fit_intercept=True,
                            intercept_scaling=1, class_weight=None,
                            random_state=None, solver='liblinear', max_iter=100,
                            multi_class='ovr', verbose=2)
```

```
glmMod.fit(X_train, y_train)
glmMod.score(X_test, y_test)
```

```
[LibLinear]0.9998565597819709
```

```
test_labels=glmMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)
```

```
0.9999998751893564
```

AdaBoost

```
adaMod = AdaBoostClassifier(base_estimator=None, n_estimators=200, learning_rate=1.0)
```

```
adaMod.fit(X_train, y_train)
adaMod.score(X_test, y_test)
```

```
1.0
```

```
test_labels=adaMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)
```

```
1.0
```

GradientBoosting

```
gbMod = GradientBoostingClassifier(loss='deviance', learning_rate=0.1, n_estimators=200, subsample=1.0,
                                   min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
                                   max_depth=3,
                                   init=None, random_state=None, max_features=None, verbose=0)
```

```
gbMod.fit(X_train, y_train)
gbMod.score(X_test, y_test)
```

```
1.0
```

```
test_labels=gbMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)
```

```
1.0
```

RandomForest

```
rfMod = RandomForestClassifier(n_estimators=10, criterion='gini', max_depth=None, min_samples_split=2,
                               min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto',
                               max_leaf_nodes=None, bootstrap=True, oob_score=False, n_jobs=1,
                               random_state=None, verbose=0)
```

```
rfMod.fit(X_train, y_train)
rfMod.score(X_test, y_test)
```

```
0.9999869599801792
```

```
test_labels=rfMod.predict_proba(np.array(X_test.values))[:,1]
roc_auc_score(y_test,test_labels , average='macro', sample_weight=None)
```

```
1.0
```

Los resultados con los algoritmos predictivos usados demuestran que tenemos un modelo de datos sesgado, esto se debe a una serie de factores, tales como:

- ❑ Análisis de un periodo de tiempo demasiado breve (inferior a 6 meses).
- ❑ Mediciones con demasiados outliers en las magnitudes meteorológicas.
- ❑ El valor de la variable ICA parcial se basa en el peor valor de los 5 contaminantes analizados.
- ❑ Datos faltantes por motivos diversos averías en el suministro eléctrico, en las comunicaciones, o en el propio equipo analizador.

Modelado. Métricas

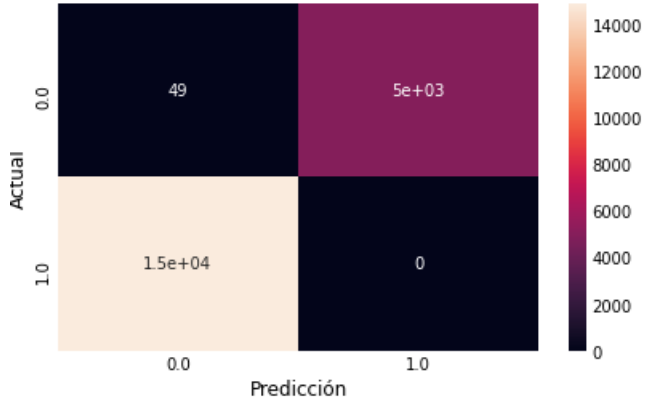
MATRIZ DE CONFUSIÓN

La Matriz de Confusión es una de las métricas más intuitivas y sencillas que se utiliza para encontrar la precisión y exactitud del modelo.

```
# Creamos array datos Train y Test
data = {'y_Actual':    y_train,
        'y_Predicted': y_test
        }
```

```
# Creamos dataFrame con columnas del array
datos = pd.DataFrame(data, columns=['y_Actual','y_Predicted'])
```

```
# Dibujamos la matrix de Confusión
confusion_matrix = pd.crosstab(datostop['y_Actual'], datostop['y_Predicted'], rownames=['Actual'], colnames=['Predicción'])

sns.heatmap(confusion_matrix, annot=True)
plt.show()
```




```
# Resultados matrix confusion
results = confusion_matrix(datostop['y_Actual'], datostop['y_Predicted'])

print ('Confusion Matrix :')
print(results)
print ('Accuracy Score :'),accuracy_score(datostop['y_Actual'], datostop['y_Predicted'])
print (')Report : ')
print( classification_report(datostop['y_Actual'], datostop['y_Predicted']) )
```

```
Confusion Matrix :
[[   49  4995]
 [14956     0]]
Accuracy Score :
)Report :
              precision    recall  f1-score   support

         0.0       0.00      0.01      0.00      5044
         1.0       0.00      0.00      0.00     14956

    accuracy                           0.00     20000
   macro avg       0.00      0.00      0.00     20000
weighted avg       0.00      0.00      0.00     20000
```

Evaluación Modelos

CURVA ROC

La curva AUC-ROC es la métrica de selección de modelo para el problema de clasificación de dos clases múltiples.

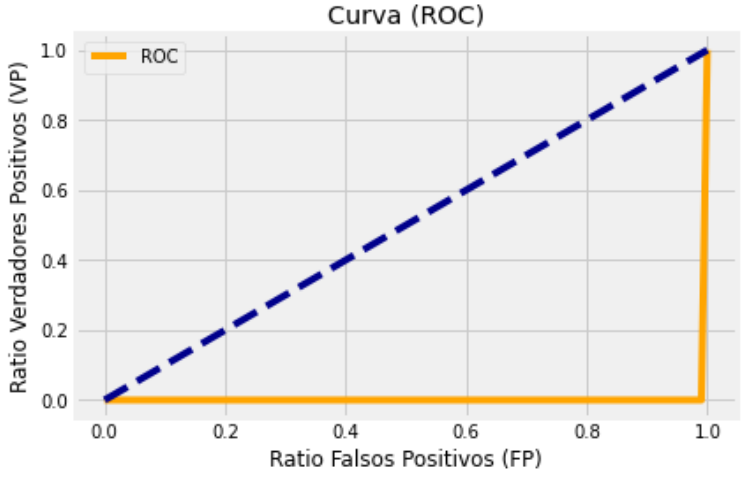
```
# Definimos funcion para trazar la Curva ROC
def plot_roc_curve(fpr, tpr):
    plt.plot(fpr, tpr, color='orange', label='ROC')
    plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
    plt.xlabel('Ratio Falsos Positivos (FP)')
    plt.ylabel('Ratio Verdadores Positivos (VP)')
    plt.title('Curva (ROC)')
    plt.legend()
    plt.show()
```

```
# Obtener la curva ROC.
fpr, tpr, thresholds = roc_curve(datostop['y_Actual'], datostop['y_Predicted'])
```

```
# Dibujamos la curva ROC
plot_roc_curve(fpr, tpr)
```

```
# Ajustamos el modelo con los datos
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

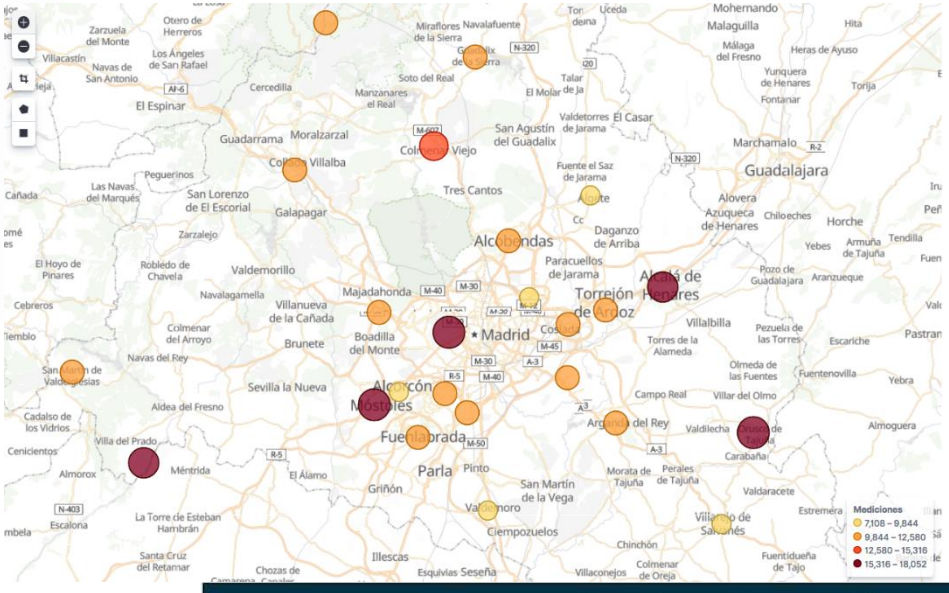
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```



Visualización

- Gráfica por TAGs sobre ICAs parciales.
  - - Agrupación por magnitudes
  - - Valores ICAs parciales más repetidos

- Gráfica geolocalizada por estación.
  - - Agrupación por estación
  - - Nº de mediciones realizadas en 5 meses



Instituto Tecnológico Telefónica

Implementación

- Entorno de visualización preparado con ELK apoyándonos en Dockers.
  - ✓ Instalación de Docker Desktop
  - ✓ Descarga de contenedor sebp/elk
  - ✓ Arrancar el contender donde se levanta distintos puertos para Elastic, Logstash y Kibana
  - ✓ Copiar dataset al contenedor para procesarlo con Logstash
  - ✓ Crear config para Logstash para su carga posterior en ElasticSearch

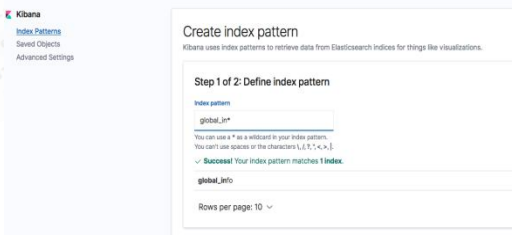
```
input {
  file {
    path => "/opt/kibana/src/plugins/home/server/services/sample_data/data_sets/calidadaire.csv"
    start_position => "beginning"
  }
}
filter {
  csv {
    columns => [ "id", "id_merge", "fechahora", "fecha", "hora", "estacion_real", "magnitud", "descrip
cion_magnitud", "factor_calculo_horario", "ica_parcial", "valor_magnitud", "provincia", "municipio", "
dia_de_la_semana", "mag81_vel_viento", "mag82_dir_viento", "mag83_temperatura", "mag86_humedad", "mag8
7_presion_atm", "mag88_radiacionUV", "mag89_precipitacion", "ESTACION", "DIRECCION", "location" ]
    separator => ";"
  }
  date{
    match => ["fechahora","yyyy-M-d HH:mm"]
    target => "fechahora"
  }
}
output {
  elasticsearch {
  hosts => ["59f96e65d9bd:9200"]
  index => "global_info"
  }
}
```

- En la UI de ElasticSearch levantada, se creción un mapping para el tipo de dato **geo_point**

```
PUT global_info
{
  "mappings": {
    "properties": {
      "location": {
        "type": "geo_point"
      }
    }
  }
}
```

- Ejecutamos logstash para realizar carga con el comando: *logstash -f /opt/logstash/config/grupo7_BigData_global.config*

- Creación de indexPattern desde la UI de ElastichSearch tras la carga desde Logstash para pasar a la parte Discover de Kibana.



Conclusiones y estudio posterior

Conclusiones:

- ✓ Las **medidas de distanciamiento** han tenido un **efecto** enorme en los **niveles de contaminación** atmosférica
- ✓ Pese a lo que pueda parecer a priori, las **condiciones meteorológicas influyen mucho menos en la calidad del aire** de lo que cabía esperar, no existiendo una fuerte relación entre ninguna de las magnitudes meteorológicas y las magnitudes de calidad del aire.
- ✓ El **Índice de Calidad del Aire (ICA**) **debería estar** conformado de forma **ponderada** por todos los agentes **contaminantes** del aire que se consideren (actualmente 6) por que, ¿Qué ocurre si el ICA parcial de cada uno de ellos quedara en 74? Estaríamos con un ICA indicando calidad del aire buena cuando todos ellos están en el umbral y muy probablemente el aire sea bastante perjudicial para la salud para cierto sector de la población.
- ✓ El análisis de las observaciones horarias de NO2 en Madrid, indica una reducción promedio respectiva de 62%. Otro resultado destacado es que los valores pico máximos por hora también muestran reducciones significativas, con relaciones entre 1.2 y 1.7. **La mejora en la calidad del aire ha ocurrido ampliamente, afectando tanto a los centros de las ciudades como a las áreas periféricas.**
- ✓ El **periodo de confinamiento ha demostrado por tanto que el confinamiento de la sociedad ha influido positivamente en la calidad del aire,** siendo (como es lógico) la acción social el verdadero impulsor de la contaminación ambiental. Y aunque se ha apreciado una disminución de monóxido de carbono a partir de marzo aunque no es la magnitud más próxima a convertirse en ICA en ninguna de las mediciones.

Propuesta de Estudio posterior:

- A la vista de las conclusiones obtenidas en el trabajo, proponemos un **estudio posterior sobre la composición del ICA**.
- El ICA debería estar conformado de forma ponderada por todos los agentes contaminantes del aire que se consideren (actualmente 6) por que, ¿Qué ocurre si el ICA parcial de cada uno de ellos quedara en 74? Estaríamos con un ICA indicando calidad del aire buena cuando todos ellos están en el umbral y muy probablemente el aire sea bastante perjudicial para la salud para cierto sector de la población.

En recuerdo a todas las personas afectas directa o indirectamente por la covid-19.

Instituto Tecnológico Telefónica