E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Diseño y Desarrollo de Sistemas de Analítica Avanzada

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Grado en Ingeniería
en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Youcef Kaced Souami

Mikel Sagüés

Pamplona, 10/06/2022

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ÍNDICE

1.-Introducción

1.1-Resumen

Este Trabajo Fin de Grado trata sobre el diseño y desarrollo de sistemas de analítica avanzada. La memoria está comprendida por tres bloques principales: un "Vocabulario Visual en Tableau", la elaboración de un cuadro de mando sobre la campaña de la declaración de la renta del año 2021, y una PoC (Prueba de Concepto) de análisis predictivo.

El primer capítulo consiste en elaborar un vocabulario visual en Tableau que sirva de guía a desarrolladores en Tableau. Su objetivo es el de ayudar a decidir cuándo es mejor utilizar cada uno de los gráficos. De guía a negocio, se emplearía para que puedan tomar decisiones en cuanto a qué gráficos son mejores para mostrar la información en sus cuadros de mando.

De cara a la segunda sección, se pretende confeccionar un cuadro de mando de estadísticas de accesos a la web de renta.navarra.es en la campaña de 2021, que sirva a las usuarias de Hacienda Foral de Navarra para ser autónomas en poder consultar qué accesos se están produciendo a la web en campaña de renta por parte de los contribuyentes. Asimismo, puede emplearse para publicarlo al ciudadano, y principalmente de cara la toma de decisiones a la hora de mejorar la infraestructura web que se corresponde con la campaña de la declaración de la renta.

Finalmente, en la tercera y última parte se va a llevar a cabo una regresión lineal sobre la calidad del aire en Navarra en los últimos años, con el objetivo de analizar la validez del modelo. Todo ello para ser aplicado en ámbitos en los que interese hacer predicciones futuras, como por ejemplo medio ambiente.

2.- Vocabulario Visual Tableau

Este capítulo se presenta como preámbulo dentro de la formación en la institución (Gobierno de Navarra). Se ha detectado por parte del personal del Gobierno de Navarra, un gran margen de aprendizaje referente al programa Tableau y las posibilidades que este ofrece. Debido a la falta de una guía oficial dentro del Gobierno de Navarra, se ha optado por hacer un "Vocabulario Visual en Tableau ".

Tableau ofrece una serie de gráficos predeterminados. Si se profundiza algo más, se puede llegar a representar muchos gráficos más, lo cual enriquece en gran medida la calidad de los cuadros de mando que se puedan desarrollar después.

El trabajo a desarrollar ha consistido en el análisis y representación gráfica de las herramientas que ofrece Tableau. Para ello, se han clasificado en función del ámbito al que pertenecen, aportando, además, una breve descripción de cada uno de los gráficos, con el objetivo de ayudar a identificar en qué situaciones sería optimo utilizar un gráfico determinado.

2.1- Tecnologías a utilizar

Tableau, Excel

2.2- Tipos de gráficos y cuándo emplearlos

Previo al desarrollo de esta sección, cabe mencionar que todos los gráficos se mostrarán de manera visual en el siguiente apartado (2.3), debido a que, para ser mostrados en sus correctas proporciones, se debe disponer el documento en horizontal. Si se pretendiese ver el conjunto en un mismo apartado, o bien se debería reducir considerablemente el tamaño de las imágenes (lo cual dificultaría su observación), o bien surgirían una gran cantidad de espacios en blanco, redundantes. Es por ello por lo que se ha determinado disponerlo en dos apartados diferentes.

De entre todos los gráficos a emplear, se ha decidido separarlos en las siguientes secciones:

- **Desviación:** Este tipo de gráficos es óptimo, como su propio nombre indica, a la hora de ver la desviación entre apartados para un mismo indicador. Por ejemplo, si se pretende ver los beneficios/pérdidas de una empresa a lo largo de los 12 meses de 2021. Asimismo, podrían ser de aplicación a la hora de representar pirámides poblacionales.

    - Diverging Bar
    - Spine Chart
    - Surplus/Deficit Line

- **Correlación:** Este es el adecuado a la hora de representar relación entre 2 o más variables para un mismo indicador. Por ejemplo, sería de gran utilidad si se quisiese ver el posicionamiento de un equipo de fútbol entre la media de la posesión de todos sus partidos, y la media de puntos ganados por partido. De este modo, se consigue ver de manera global la efectividad que tiene su posesión.

    - Scatterplot
    - Line + Column
    - Connected Scatterplot
    - Bubble
    - XY Heatmap

- **Posicionamiento:** El indicado si el objetivo consiste en apreciar la diferencia entre por ejemplo unos meses y otros en cuanto a ventas en una empresa. Podría valer si lo que se quiere es ver cuál es la temporada alta y la baja.

    - Slope
    - Ordered Bar
    - Ordered Column

- **Distribución:** Como su propio nombre indica, son adecuados para ver cómo se distribuye una variable. Un claro ejemplo serían las pirámides poblacionales.

    - Boxplot

- Population Piramid
- Cumulative Curve
- Dot Plot

- **Cambios temporales:** Este tipo de gráficos tiene su principal utilidad en ver cómo una o más variables fluyen/cambian a lo largo de un periodo determinado de tiempo.

  - Area Chart
  - Line + Column
  - Fan Chart
  - Calendar Heatmap
  - Circle Timeline
  - Priestley Timeline

- **Proporción:** Aplicables a la hora de ver la proporción de cada subapartado respecto de una variable, más que ver su cantidad parcial. Puede que interese ver cuánto se ha vendido en marzo respecto al total anual, más que la cantidad exacta de ventas en marzo.

  - Pie Chart
  - Sunburst
  - Proportional Stacked Bar
  - Tree Map
  - Grid Plot

- **Magnitud:** Mediante este tipo de gráficos se pueden relacionar múltiples variables que en sentido global están ciertamente relacionadas. Por poner un ejemplo, se podría ver el global de un futbolista relacionando su valoración en físico, goles marcados, asistencias, partidos jugados…

  - Radar Chart

- Parallel Coordinates

- **Espacial:** Ideal para apreciar ciertos indicadores que tienen más sentido vistos espacialmente mediante mapas geográficos. Pueden ser usados para ver cómo van unas elecciones en tiempo real, o por ejemplo también son útiles para ver la incidencia del paro en una región, etc.

    - Heat Map
    - Flow Map
    - Proportional Symbol
    - Scaled Cartogram
    - Dot Density

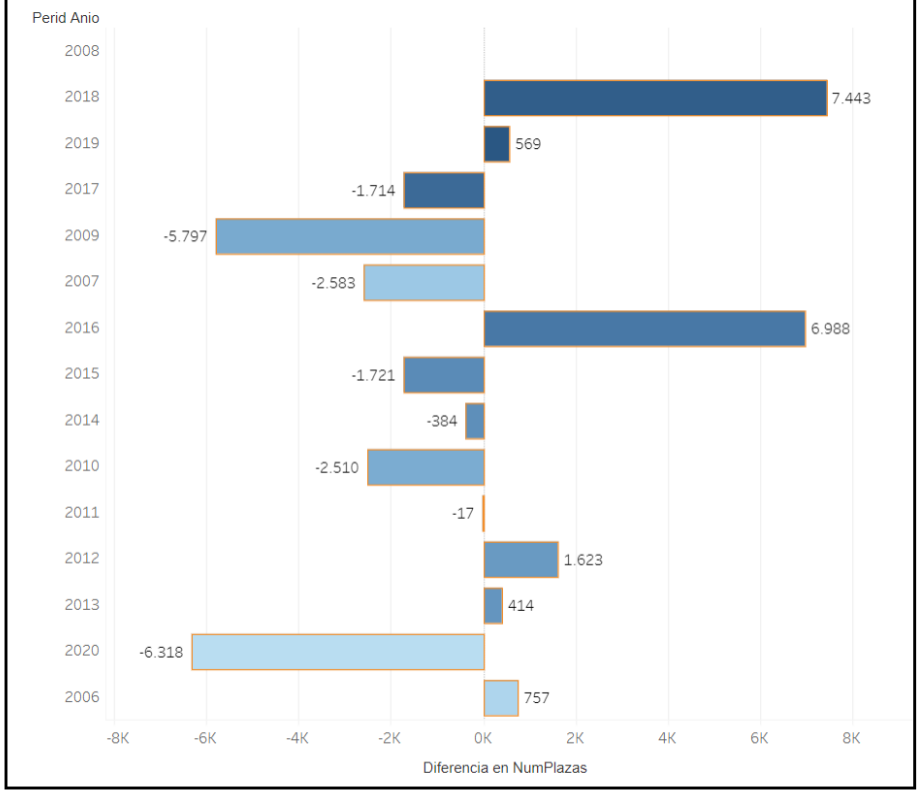
*Cabe destacar que es posible que se vea alguna vez un mismo gráfico en dos secciones iguales, ya que este puede ser empleado en dos ámbitos diferentes.

2.3- Resultados

DESVIACIÓN

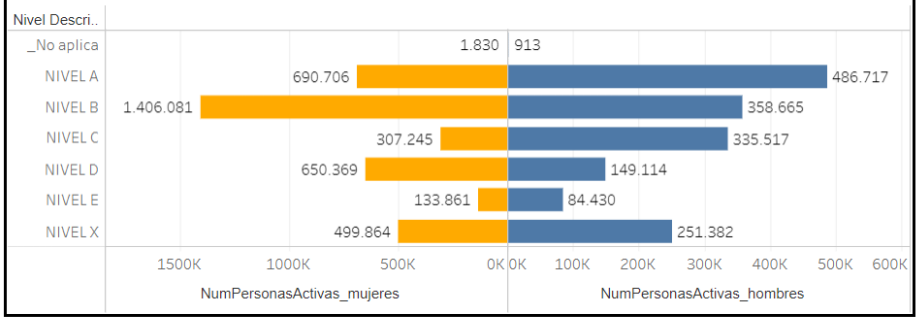
Diverging Bar

Un gráfico de barras que tenga valores tanto positivos como negativos



Spine Chart

Aplicable cuando queremos pirámides poblacionales



Surplus/Deficit Line

Un gráfico de barras que tenga valores tanto positivos como negativos

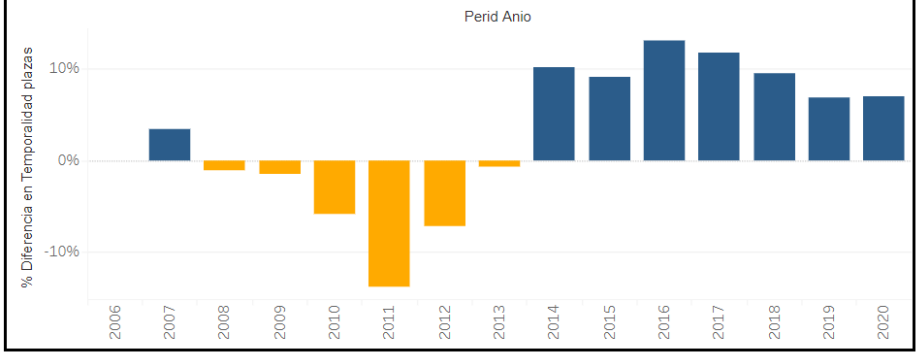
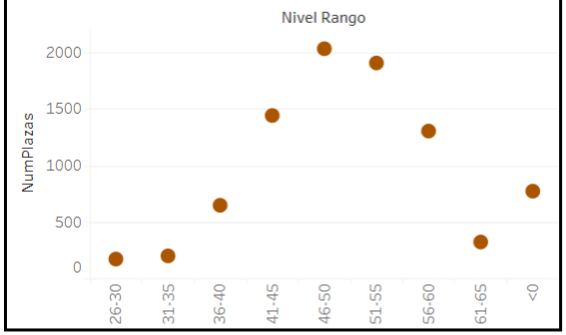


Figura 1: Desviación

CORRELACIÓN

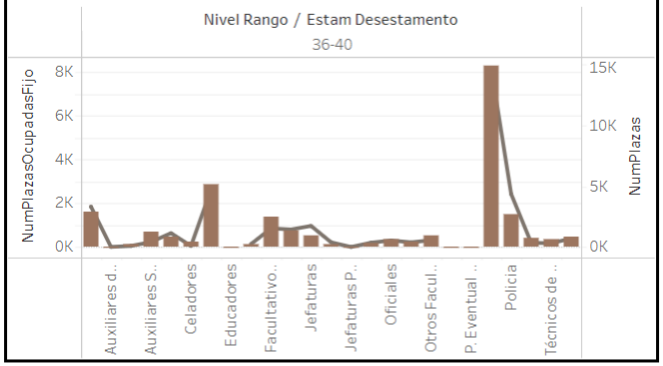
Scatterplot

Esta es el mejor gráfico para visualizar la relación entre 2 variables continuas



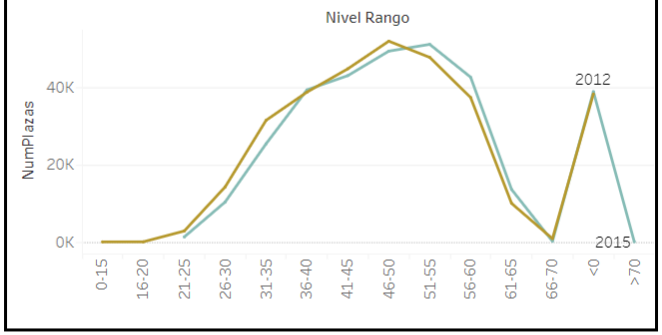
Line + Column

Como un gráfico de barras con una línea de tendencia añadida



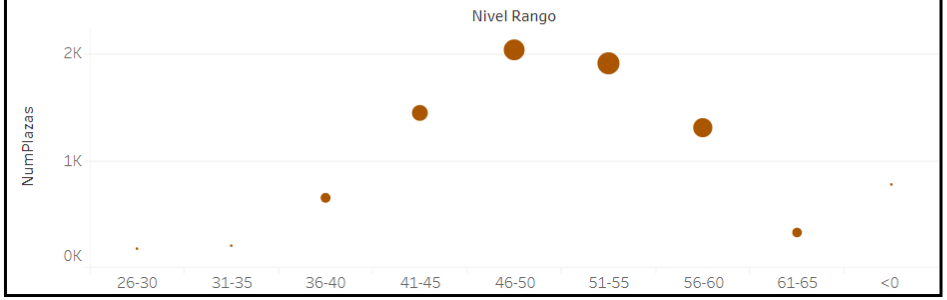
Connected Scatterplot

Este es el mejor gráfico para visualizar la relación entre 2 variables continuas



Bubble

Es como un scatter plot en el que añadimos una 3ª variable apreciable en el tamaño de cada burbuja



XY Heatmap

Aplicable cuando queremos cómo se distrubuye la variable, distinguiendo claramente altas de bajas concentraciones

	Nivel Codigo						
Nivel Rango	-1	A	B	C	D	E	X
0-15		75,00%			25,00%		
16-20			2,01%	2,24%	59,51%	36,24%	
21-25	0,22%	8,22%	70,90%	8,69%	7,25%	4,73%	
26-30	0,21%	17,78%	61,63%	11,05%	7,17%	2,16%	
31-35	0,33%	23,95%	46,94%	15,04%	11,46%	2,28%	
36-40	0,31%	25,47%	37,80%	18,19%	15,20%	3,03%	0,00%
41-45	0,51%	26,08%	34,25%	18,31%	16,69%	4,16%	0,01%
46-50	0,70%	27,10%	33,85%	16,52%	16,60%	5,23%	0,00%
51-55	0,72%	26,97%	36,35%	14,14%	16,15%	5,66%	0,00%
56-60	0,58%	26,94%	38,44%	12,77%	15,85%	5,42%	
61-65	0,69%	33,88%	28,73%	13,04%	17,59%	6,07%	0,00%
66-70	1,97%	58,70%	19,88%	8,17%	7,46%	3,82%	
<0	5,50%	33,63%	38,21%	8,39%	11,01%	3,25%	0,00%
>70	69,57%					30,43%	

Figura 2: Correlación

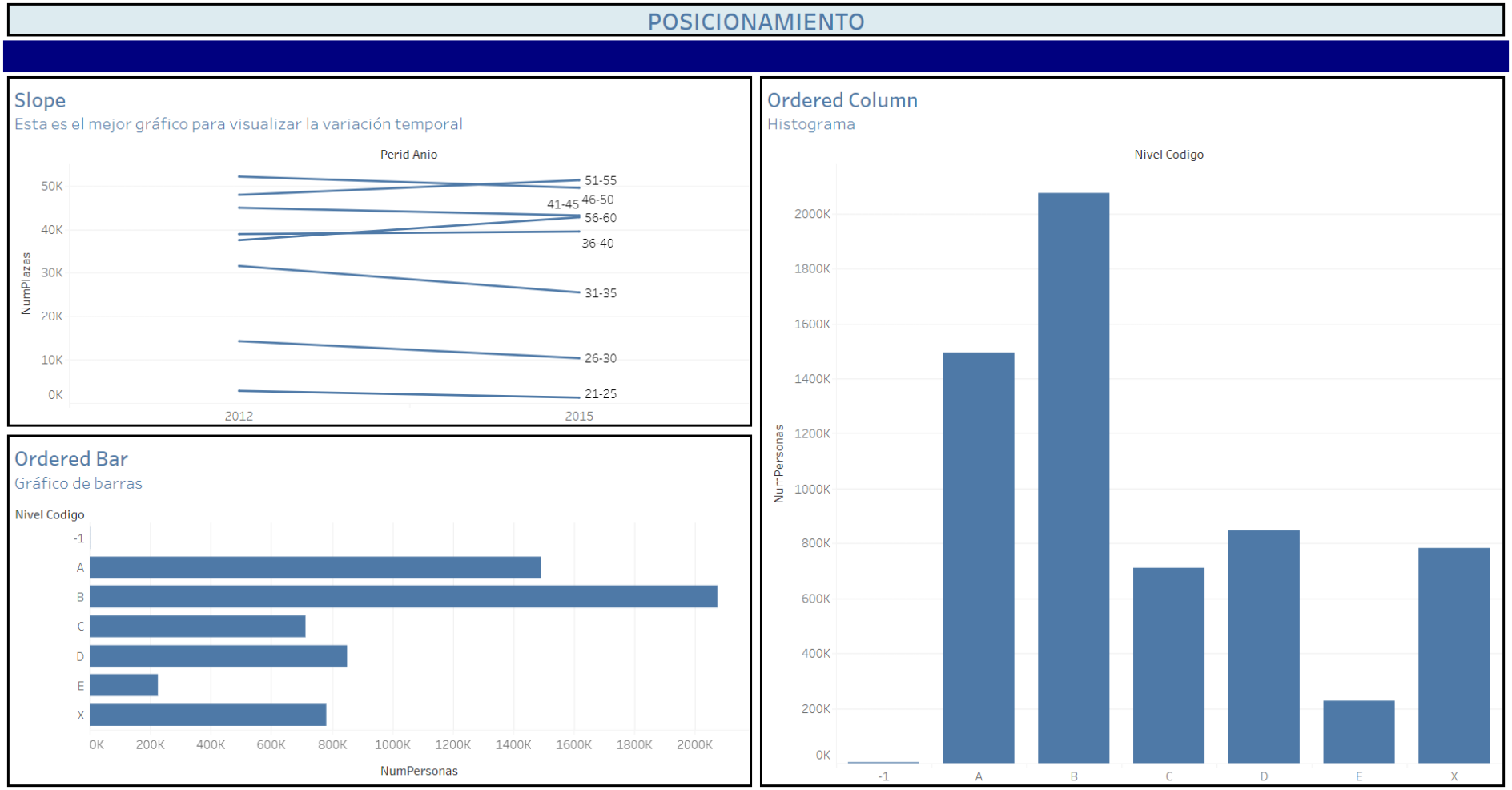


Figura 3: Posicionamiento

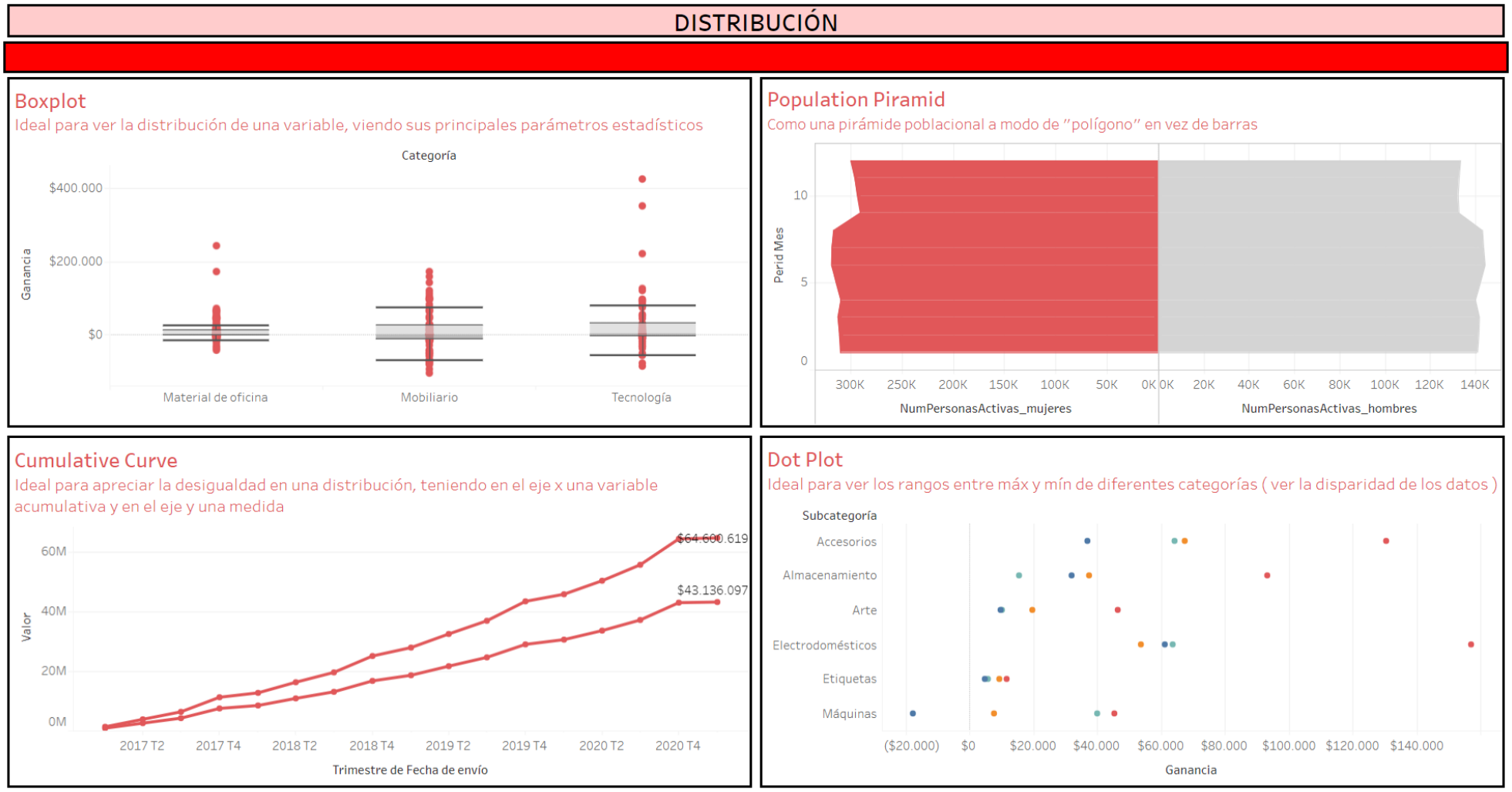


Figura 4: Distribución

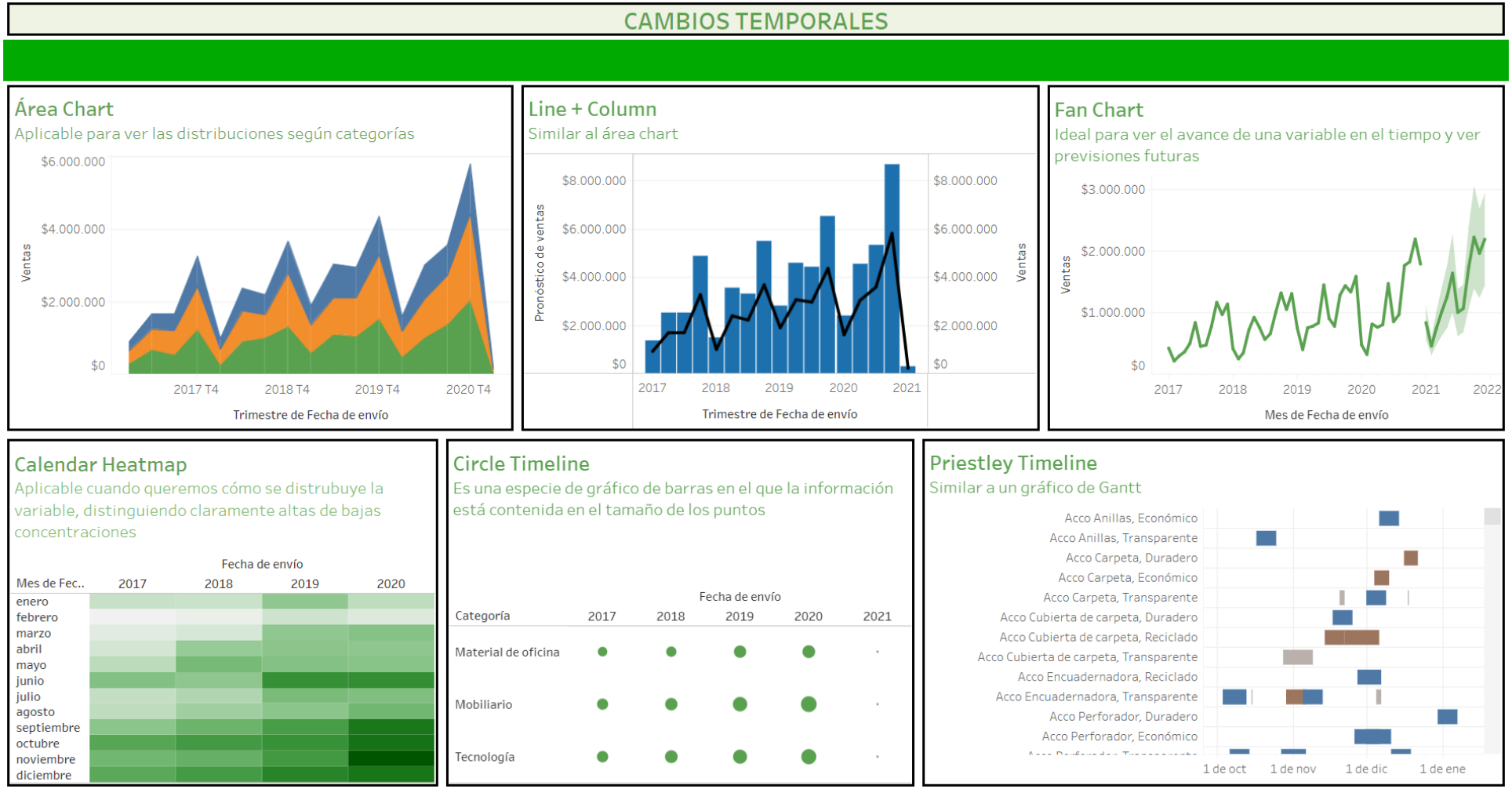


Figura 5: Cambios temporales

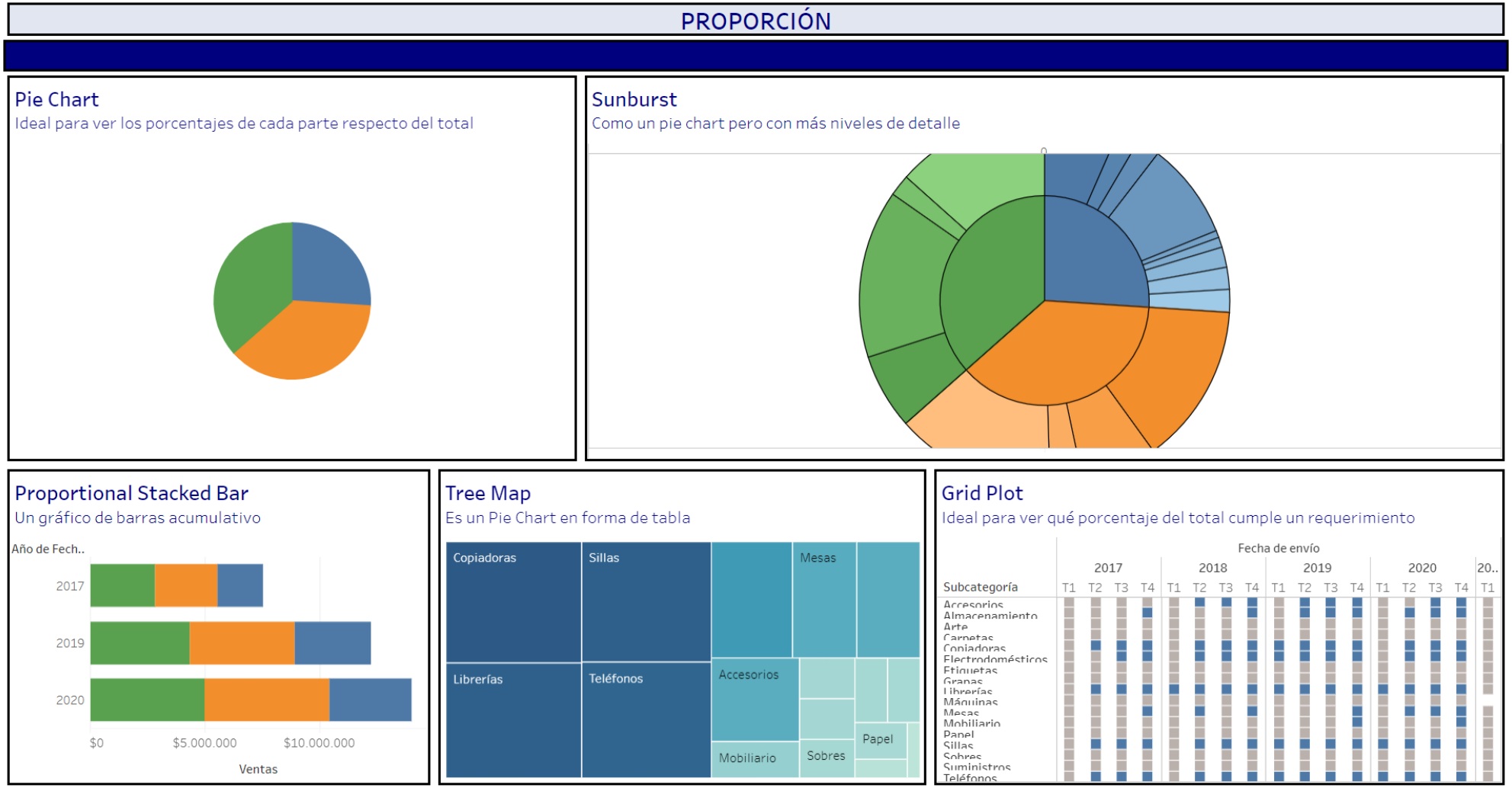


Figura 6: Proporción

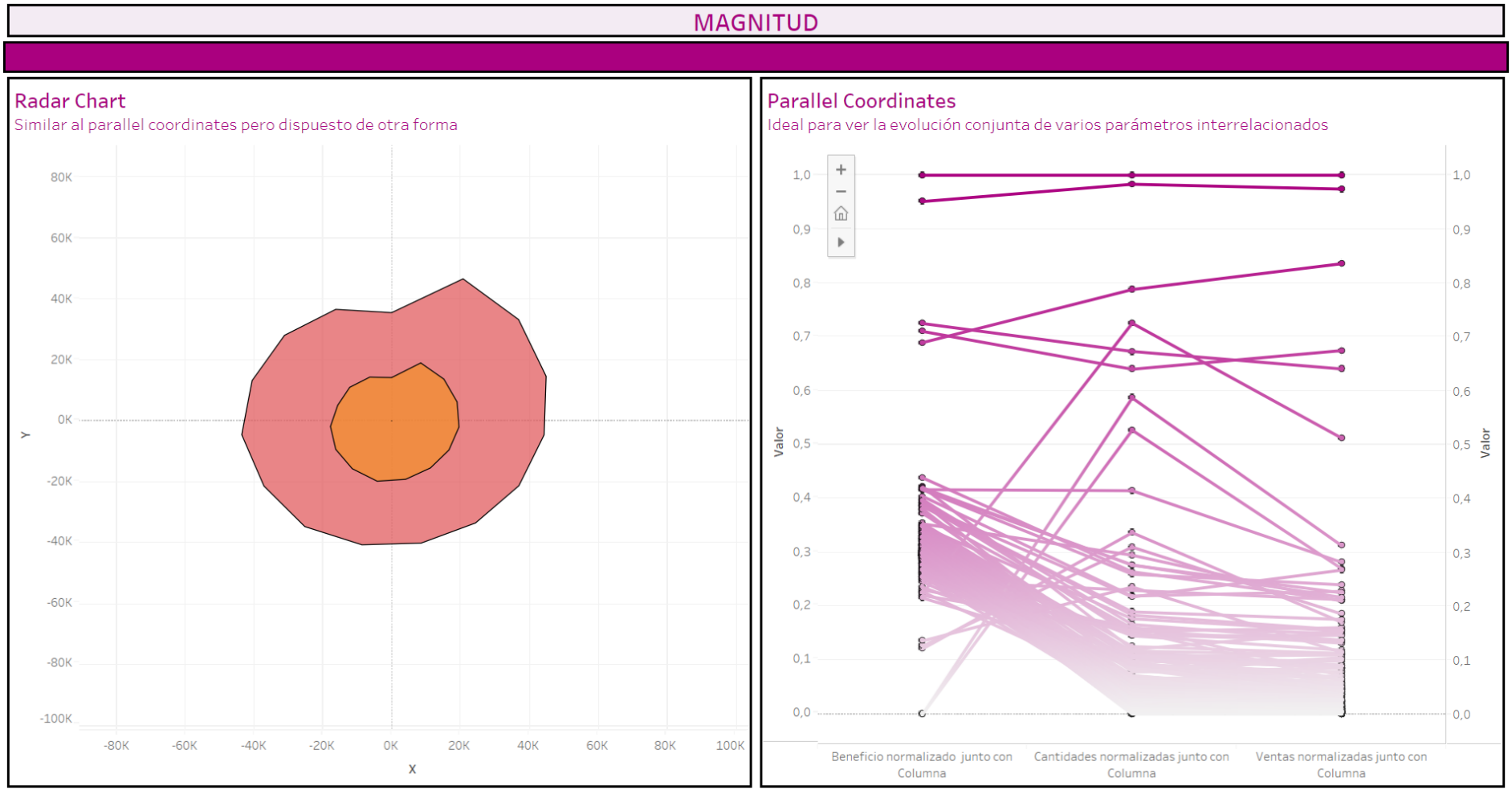


Figura 7: Magnitud

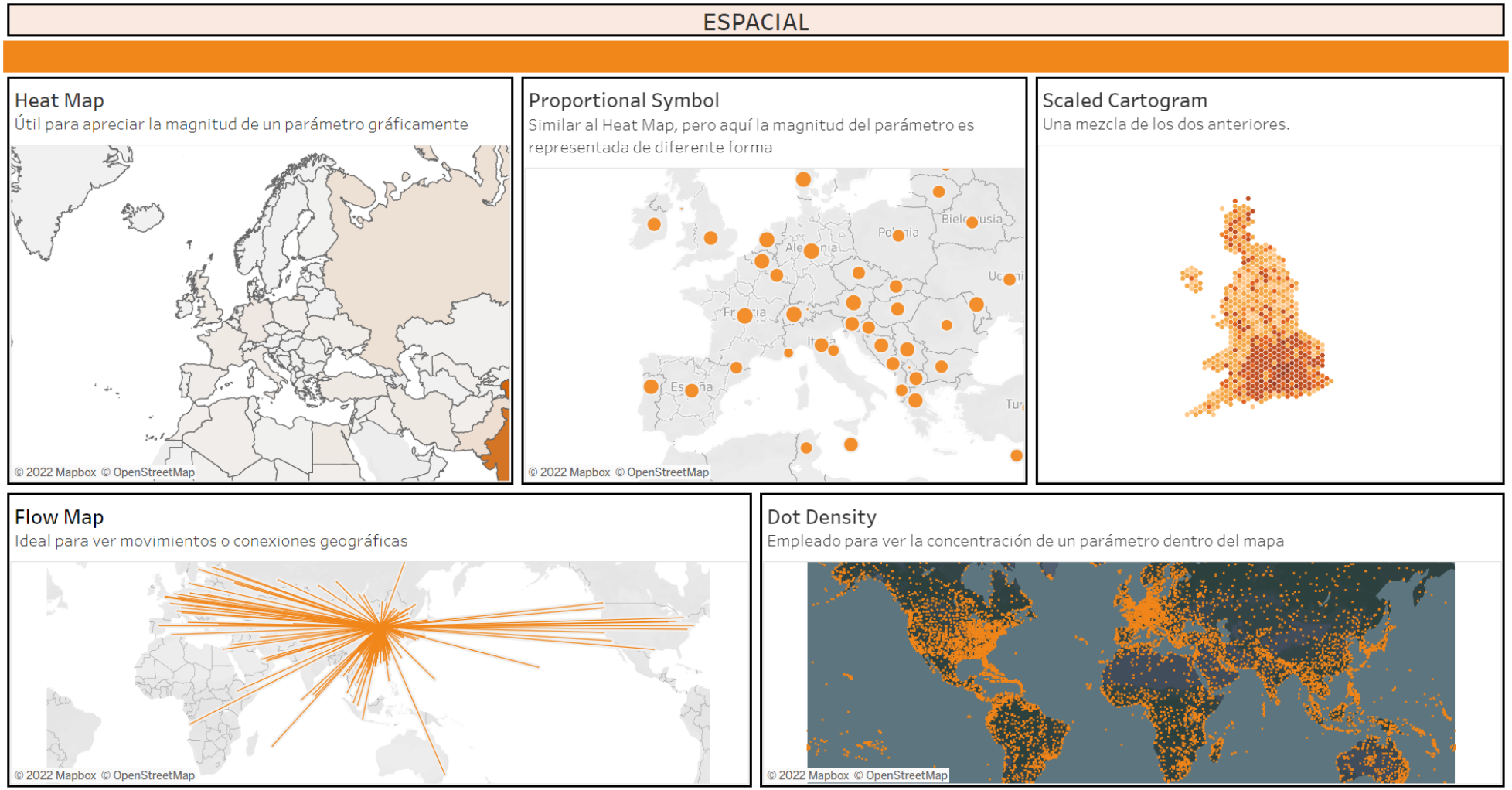


Figura 8: Espacial

3.- Cuadro de Mando Renta

3.1- Introducción

En este capítulo se describe la necesidad de mejorar el servicio dedicado a la declaración de la renta por parte del Gobierno de Navarra. Se pretende hacer un estudio sobre ciertas webs dedicadas específicamente a esta campaña. El objetivo consiste en extraer conocimiento relativo al tráfico que reciben estas webs, cuáles son los periodos con más tráfico, qué webs son las más útiles… Todo esto con el fin de mejorar este servicio de cara a campañas de años futuros.

El estudio a realizar es sobre la campaña de la declaración de la renta del año 2021. Las webs sobre las que se van a extraer los datos han sido proporcionadas por personal del Gobierno de Navarra.

3.1.1- Bussiness inteligence

Se denomina Business Intelligence (BI) al conjunto de estrategias, aplicaciones, datos, productos, tecnologías y arquitectura técnicas, los cuales están enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa. [1]

Business Intelligence (inteligencia de negocio) surge debido a la falta de estrategias para depurar la información contenida en grandes volúmenes, y convertir esta en aprendizaje/conocimiento para su posterior uso en favor de la propia institución. Está formada por el conjunto de herramientas cuyos objetivos son:

- Extracción de los datos (en bruto)

- Transformación de estos datos en información

- Generar conocimiento a partir de esta información, para ser empleada en áreas de administración y gestión empresarial.

El BI tiene una serie de procesos, los cuáles siempre siguen un patrón.

La serie de pasos a seguir para la consecución de este aprendizaje a la empresa/ institución en cuestión son los siguientes:

- Se da la existencia una serie de sistemas que albergan toda esta información desordenada en forma de datos, desde los cuales se van a volcar estos datos. Estos sistemas pueden ser propios de la empresa como:

    - Bases de datos propias.
    - Sistemas de ERP (Enterprise Resource Planning).
    - Sistemas de CRM (Customer Relationship Management).
    - Gestores Documentales (EMC – Enterprise Content Management).

  Asimismo, también puede llegar a darse el caso de que estos estén hallados en fuentes externas que igualmente afecten al devenir de la propia empresa, ya sean documentos oficiales, redes sociales, prensa, información pública en internet, etc….

- Los datos provenientes de estos sistemas deben ser añadidos al repositorio de información de la organización, pero deben pasar previamente un proceso para filtrar los datos, normalizarlos, formatearlos, contextualizarlos y estructurarlos correctamente. En otras palabras, no es suficiente con ir seleccionando la información de manera aleatoria. Es necesario aportar un significado a los datos, realizar las transformaciones necesarias para que los datos tengan un formato correcto y unificado, saber elegir qué indicadores dan sentido a los datos y qué indicadores no aportan forma al conjunto global de datos. Todo esto con el principal y único objetivo de transformar datos en información útil. Esta labor es realizada principalmente por las herramientas ETL (Extract, Transform, Load).

- Ahora sí pueden ser incorporados a los sistemas de información del sistema BI.

- Una vez que la información útil se encuentra en el repositorio de información empresarial, puede ser analizada por las herramientas de BI avanzadas destinadas a la obtención de conocimiento, como consultas e informes (reporting), elaboración de cubos para su posterior análisis, inspección de los datos a través del datamining y/o su visualización y estudio mediante las interfaces avanzadas de las herramientas. En el caso de estudio a realizar, la visualización de este conocimiento se realizará mediante cuadros de mando con la herramienta Tableau.

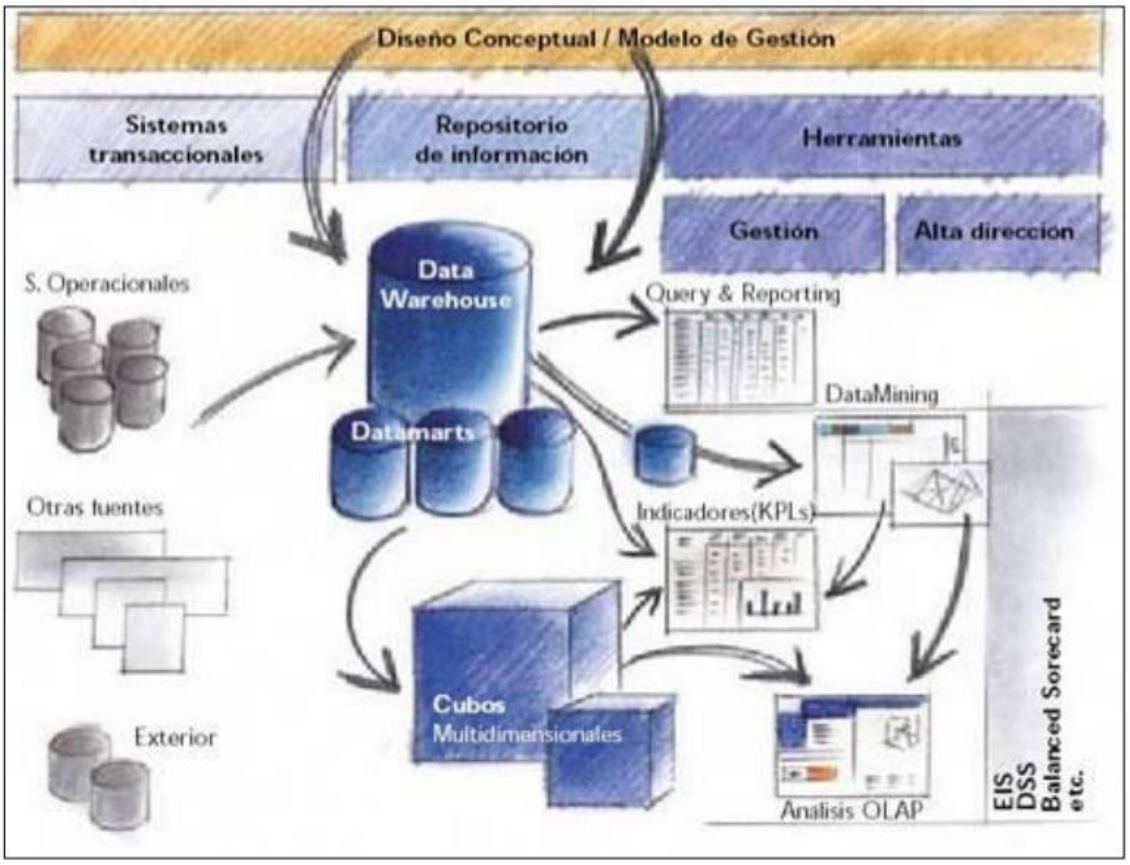


Figura 9: Procesos BI [2]

TIPOS DE HERRAMIENTAS

Posteriormente, habría que hacer distinción entre los diferentes tipos de herramientas BI:

- **Extracción de datos:** son las llamadas herramientas ETL, de extracción, transformación y carga de los datos en los sistemas de información.

- **Arquitectura de información**: es la parte central de un sistema de BI. Aquí están los Data warehouse, Data marts, metadatos, etc., los cuales son grandes almacenes de datos estructurados y listos para ser empleados por el último tipo de herramientas: las de consulta y análisis.

- **Consulta y Análisis:** aquí es donde se incluyen las principales herramientas de BI. Los otros dos tipos crean más o menos lo que vendría a ser el lugar bien construido y asentado, en el que la información ya está depurada y organizada para luego poder extraer conocimiento útil para la empresa. Estas son las herramientas que los usuarios utilizan para consultar y analizar la información. De entre este tipo de herramientas, las hay de varios subtipos:

  - **Herramientas de Reporting:** Estas son las que empleamos para hacer consultas contra los sistemas de información y presentar los resultados en informes sencillos. Existen herramientas de este tipo que proporcionan un entorno amigable para que usuarios no expertos puedan diseñar tanto las consultas como los informes de manera intuitiva.

  - **Servidores OLAP (Procesamiento Analítico en Línea):** Proporciona al usuario una gran capacidad de análisis en varias dimensiones de la información almacenada en los data warehouse y data marts. El objetivo o más bien beneficio de esto es poder ver nuestro bloque total de información desde distintas perspectivas, para interpretar los datos de todas las formas posibles y llegar así a un conocimiento/aprendizaje mucho más maduro.

  - **Herramientas de datamining o minería de datos:** Son sistemas expertos, dedicadas principalmente al manejo de grandes volúmenes de datos con el objetivo de aislar e identificar patrones útiles para el negocio. Para ello se emplean modelos matemáticos, métodos estadísticos, redes neuronales, ... Este tipo de herramientas están más enfocadas a usuarios profesionales.

  - **Cuadros de Mando Integrales (CMI), Sistemas de Soporte a la Decisión (DSS) y Sistemas de Información Ejecutiva (EIS)**

Figura 10: Herramientas BI [3]

3.1.2. ¿Qué es un cuadro de mando?

El Cuadro de Mando Integral (CMI) es un modelo de gestión que traduce la estrategia en objetivos relacionados entre sí, medidos a través de indicadores y ligados a unos planes de acción que permiten alinear el comportamiento de los miembros de la organización con la estrategia de la empresa.
Se trata de una herramienta de control de gestión, cuya función primordial es la implantación y comunicación de la estrategia a toda la empresa.[4]

Cabe destacar que es de vital importancia para cualquier tipo de organización, ya sea una empresa, negocio, administración pública… el hecho de saber qué está pasando en tiempo real, ya que poder acceder a esta forma de visión global y real hace bastante más efectiva toma de decisiones ante situaciones relacionadas tanto con las tendencias del mercado, solucionar rápido percances surgidos en la empresa, estar siempre al tanto de la competencia… El cuadro de mando llega para facilitar todo lo mencionado anteriormente. Su función vital es la de para monitorizar todos los parámetros de la empresa y disponer de una imagen real de lo que ocurre dentro y fuera de la misma.

En resumen, un cuadro de mando es una “herramienta” a partir de la cual se lleva parte de la gestión de la empresa/institución, con el objetivo de medir la situación de esta misma, todo ello desde una perspectiva amplia y efectiva. En el cuadro de mando se podrán ver diferentes indicadores, los cuales vendrán representados en gráficos de todo tipo que ofrecerán una perspectiva general, con la menor subjetividad posible y en tiempo real, todo ello para monitorizar todos los parámetros de la empresa y disponer de una imagen real de lo que ocurre dentro y fuera de la misma.

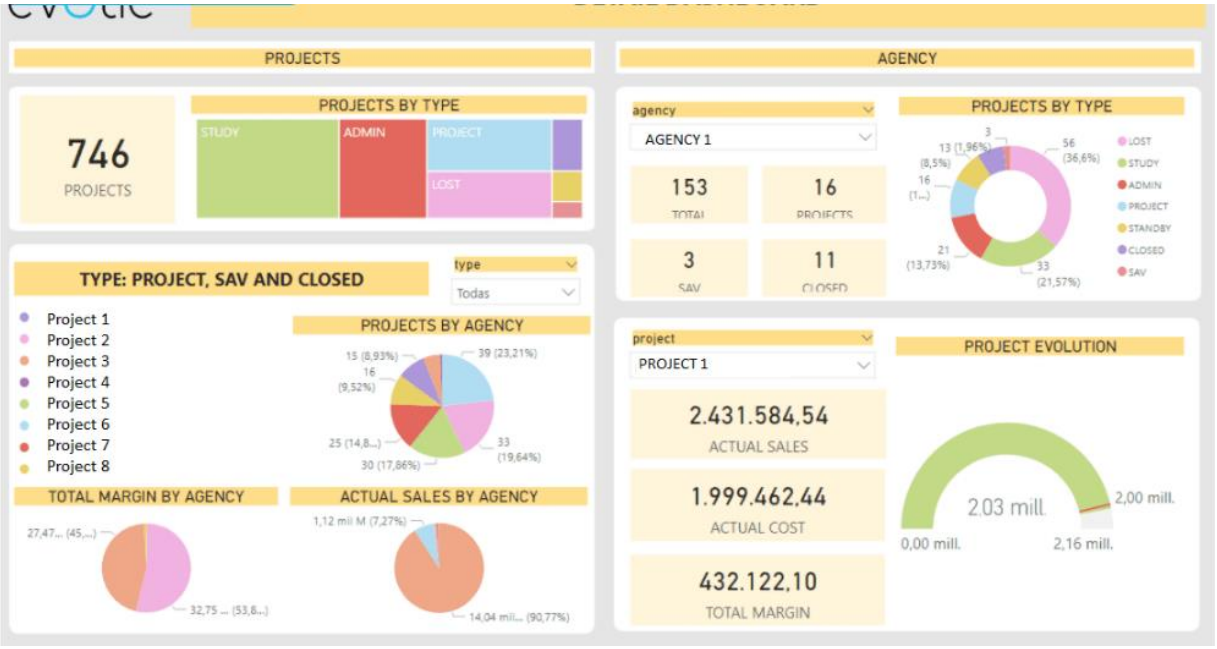


Figura 11: Cuadro de Mando [5]

Los indicadores, también llamados métricas o KPIs que vienen representados dentro de un cuadro de mando no pueden ser elegidos de forma arbitraria, ya que dependiendo de su elección vendrá dado el sentido y la efectividad del cuadro de mando, lo cual acaba derivando en buenas/malas decisiones a nivel de empresa/institución. En otras palabras, la elección de estos indicadores es el principio de una cadena de suma importancia, por lo que hay que darle el mérito que conlleva. Para poder tener una visión global y real de lo que sucede en la empresa estos indicadores deben representarse tanto en cifras como de manera visual.

Figura 12: Indicadores Cuadro de Mando [6]

De entre las ventajas que ofrece el empleo de esta herramienta, destacan:

- Visión global de la empresa/institución
- Facilita el diseño y la planificación de estrategias
- Reducción de riesgos potenciales
- Mejora del diálogo interno
- Da una idea del éxito/fracaso de la estrategia a analizar

3.2- Tecnologías a utilizar

Google Analytics, Python, Excel, Microsoft SQL Server, Tableau, MobaxTerm.

3.3- Desarrollo del cuadro de mando

3.3.1- Creación de las cuentas

Lo primero con lo que se debió proceder fue tener acceso a los datos que se pretendía explotar. Estos datos se encontraban en la herramienta Google Analytics. La primera idea del plan a partir del cual se va a proceder consiste en tener permisos para visualizar los datos de las siguientes webs:

https://www.navarra.es/es/tramites/on/-/line/Impresion-de-declaraciones?back=true&pageBackId=5722676

https://www.navarra.es/es/tramites/on/-/line/Seleccion-de-la-forma-de-comunicacion-de-la-propuesta-de-autoliquidacion-de-Renta?back=true&pageBackId=5722676

https://www.navarra.es/es/tramites/on/-/line/Impresion-de-certificados-de-retenciones-fiscales?back=true&pageBackId=5722676

https://www.navarra.es/es/tramites/on/-/line/Impresion-de-declaraciones?back=true&pageBackId=5722676

https://www.navarra.es/es/tramites/on/-/line/Gestion-del-PIN?back=true&pageBackId=5722676

https://www.navarra.es/es/tramites/on/-/line/Cita-previa-Hacienda-Foral-de-Navarra

https://www.navarra.es/es/tramites/on/-/line/Obtencion-de-datos-fiscales

https://www.navarra.es/es/tramites/on/-/line/cita-previa-para-hacer-la-declaracion-de-la-renta?back=true&pageBackId=5722676

https://www.navarra.es/es/tramites/on/-/line/Simulador-de-la-declaracion-del-IRPF?back=true&pageBackId=5722676

https://www.navarra.es/es/hacienda/te-atendemos

https://www.navarra.es/es/tramites/ayuda-para-tramitar-por-internet

https://www.navarra.es/es/tramites/ayuda-para-tramitar-por-internet/clave

https://www.navarra.es/es/tramites/ayuda-para-tramitar-por-internet/certificado-digital

https://www.navarra.es/es/tramites/ayuda-para-tramitar-por-internet/otras-credenciales

Google Analytics permite ver una infinidad de indicadores, de entre los cuales, pocos acaban sirviendo para este estudio. De entre ellos tenemos:

- **Fecha y Hora**: este permitirá ver responder a la pregunta de cuándo se dan los picos de tráfico

- **Número de visualizaciones de la página**

- **Número de visualizaciones únicas de la página**

- **Tiempo en la página:** este dará una idea de si es útil la página, si está clara, si no es intuitiva, …

- **Porcentaje de rebote:** este indicador es vital ya que proporciona información acerca del número de visualizaciones sobre el total, en el que el tiempo interactuando con la página ha sido de 0 segundos, es decir, está indicando si es una web de paso o es una web útil para el ciudadano.

3.3.2- Extracción de datos

Ahora llega el momento de la extracción de los datos. Hay varias maneras de hacerlo, de entre las cuales las más óptimas han sido consideradas:

- **Google Analytics Query:** Se puede usar esta herramienta paralela a Google Analytics. El problema es que tiene un gran inconveniente. No deja ver/extraer más de 10000 datos en una vista. Como se pretende ver el tráfico en un periodo temporal algo extenso (abril-junio 2021), el total de los datos es bastante superior al límite de esta herramienta. Además, el modo de

exportación es un archivo con extensión .csv, que tampoco acaba de convencer para el trabajo.

- **Python:** Esta alternativa es bastante más compleja, pero tiene bastantes ventajas una vez acabada. La idea consiste en emplear un fichero con extensión .py para volcar directamente la información que queremos (con las métricas, dimensiones y filtros que se pretende), desde Google Analytics a Microsoft SQL Server, que es el programa con el cual se va a trabajar en base de datos. Lo complejo de esta opción es programar el script Python en su totalidad, aunque ofrece la ventaja de que una vez acabado, solamente hay que ejecutarlo cada vez que queramos actualizar información, y se tendría la información perfectamente insertada en Microsoft SQL. Si se pretende cambiar alguna métrica, dimensión, filtro… simplemente habría que modificarlo dentro del script (muy sencillo). La primera vez cuesta, pero para posteriores veces es bastante cómodo y sencillo.

```python
import pandas as pd
import numpy as np
import httplib2
from apiclient.discovery import build
from oauth2client.service_account import ServiceAccountCredentials
from datetime import datetime,date,timedelta
import sqlalchemy


############ VARIABLES ############
fecha_start="2021-04-06"
fecha_end="2021-06-30"

SCOPES = ['https://www.googleapis.com/auth/analytics.readonly']
DISCOVERY_URI = ('https://analyticsreporting.googleapis.com/$discovery/rest')
KEY_FILE_LOCATION = 'core-computer-343509-63bd3b8d75fd.json'
VIEW_ID = '87914512'

BDSERVER = 'DC1GVALSQL003,51433'
DATABASE = 'ACCESOS_WEB_HFN_OLAP'

QUERIES = [
                {
                    'writeName' : 'GA_PAGE_TRACKING_1',
                    'writeFormat' : 'BD',
                    'metrics' : ["pageviews", "uniquePageviews", "avgTimeOnPage", "bounceRate"],
                    'dimensions' : ["pagePath"],
                    "filtersExpression":["pagePath=@/tramites/on/-/", "pagePath=@/tramites/ayuda-para-tramitar-por-internet/", "pagePath=@/hacienda/te-atendemos"]
                }
           ]
```

Figura 13: Script Python

```
############ FUNCIONES ############
def initBD():
    engine = sqlalchemy.create_engine('mssql+pyodbc://'+ BDSERVER +'/'+ DATABASE +'?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server')
    return engine.connect()

def initialize_analyticsreporting():
    """Initializes an analyticsreporting service object.

    Returns:
    analytics an authorized analyticsreporting service object.
    """

    credentials = ServiceAccountCredentials.from_json_keyfile_name(KEY_FILE_LOCATION, scopes=SCOPES)

    http = credentials.authorize(httplib2.Http())

    # Build the service object.
    analytics = build('analytics', 'v4', http=http, discoveryServiceUrl=DISCOVERY_URI)

    return analytics

def get_report(analytics, metrics, dimensions,filtersExpression):
    # Use the Analytics Service Object to query the Analytics Reporting API V4.
    def formatMetric(name):
        return {'expression': 'ga:'+ name}

    def formatDim(name):
        return {'name': 'ga:'+ name}

    return analytics.reports().batchGet(
        body={
            'reportRequests': [
                {
                    'viewId': VIEW_ID,
                    'dateRanges': [{'startDate': fecha_start, 'endDate': fecha_end}],
                    'includeEmptyRows': True,
                    'samplingLevel':'LARGE',
                    'metrics':  list(map(formatMetric,metrics)),
```

Figura 14: Script Python

```
                    'metrics':  list(map(formatMetric,metrics)),
                    "dimensions": list(map(formatDim,dimensions)),
                    "filtersExpression":"ga:pagePath=@/tramites/on/-/,ga:pagePath=@/tramites/ayuda-para-tramitar-por-internet/,ga:pagePath=@/hacienda/te-atendemos",

                }
            ]
        }
    ).execute()

 def htmlToDf(response):

       list = []
       # get report data
       for report in response.get('reports', []):
         # set column headers
         columnHeader = report.get('columnHeader', {})
         dimensionHeaders = columnHeader.get('dimensions', [])
         metricHeaders = columnHeader.get('metricHeader', {}).get('metricHeaderEntries', [])
         rows = report.get('data', {}).get('rows', [])

         for row in rows:
             # create dict for each row
             dict = {}
             dimensions = row.get('dimensions', [])
             dateRangeValues = row.get('metrics', [])

             # fill dict with dimension header (key) and dimension value (value)
             for header, dimension in zip(dimensionHeaders, dimensions):
               dict[header] = dimension

             # fill dict with metric header (key) and metric value (value)
             for i, values in enumerate(dateRangeValues):
               for metric, value in zip(metricHeaders, values.get('values')):
                 #set int as int, float a float
                 if ',' in value or '.' in value:
                   dict[metric.get('name')] = float(value)
                 else:
                   dict[metric.get('name')] = int(value)

             list.append(dict)
```

Figura 15: Script Python

```
            list.append(dict)

        df = pd.DataFrame(list)
        return df

def writeReport (df, name, format, cnxn):
    #escribimos los parametros temporales
    df["F_DESDE"] = fecha_start
    df["F_HASTA"] = fecha_end
    df["F_MODIFICACION"] = datetime.now()

    if format == "CSV":
        df.to_csv(name + '.csv')
    elif format == "BD":
        df.to_sql(name,cnxn,schema='dbo',if_exists="append",index=False)
    else:
        print("Sin implementar")

    return

def runQueries():
    analytics = initialize_analyticsreporting()

    ibd = 0
    for qformat in QUERIES:
        if qformat['writeFormat'] == "BD":
            bdcon = initBD()
            ibd = 1
            break

    for query in QUERIES:
        response = get_report(analytics,query['metrics'],query['dimensions'],query['filtersExpression'])
        df = htmlToDf(response)
        if query["writeFormat"] == "BD":
            writeReport(df,query['writeName'],query["writeFormat"],bdcon)
        else:
            writeReport(df,query['writeName'],query["writeFormat"],"")

    if ibd == 1:
        bdcon.close()
```

Figura 16: Script Python

```
    if ibd == 1:
        bdcon.close()

if __name__ == "__main__":
    dfecha_start = datetime.strptime(fecha_start, '%Y-%m-%d')
    dfecha_end = datetime.strptime(fecha_end, '%Y-%m-%d')
    fecha_start=date.strftime(dfecha_start, '%Y-%m-%d')
    fecha_end=date.strftime(dfecha_start, '%Y-%m-%d')
    while (dfecha_start <= dfecha_end):
        runQueries()
        dfecha_start = dfecha_start + timedelta(days=1)
        fecha_start=date.strftime(dfecha_start, '%Y-%m-%d')
        fecha_end=date.strftime(dfecha_start, '%Y-%m-%d')
```

Figura 17: Script Python

Como se puede ver en las figuras 13-17, el script está dividido principalmente en 2 bloques.

En el primero se declaran las variables. Estas son el nombre del servidor en Microsoft SQL, los datos de la vista en Google Analytics, el nombre de la nueva tabla a crear en base de datos, y las métricas, dimensiones y filtros que se quieren aplicar a la vista de GA.

En el segundo bloque ya se desarrollan todas las funciones, que se dedican principalmente a crear la nueva vacía tabla en base de datos, consultar todos los datos de GA, seleccionar de todos ellos según las métricas, dimensiones y filtros, y volcar todos estos datos en Microsoft SQL en la tabla creada con ese fin. Estas son las

funciones initBD e initialize_analyticsreporting (inicializan la nueva tabla y se conectan a ella, dentro del servidor correspondiente en Microsoft SQL), get_report y write_report (seleccionan los datos necesarios de GA), htmlToDf y runQueries (esta última sería como el "main", que ejecuta todas las demás funciones en su correspondiente orden).

Para que este script funcione correctamente, se debe cumplir con varias cosas. Antes que nada, cabe destacar que cuando se creó la cuenta de google para este trabajo, hubo que meterse a Google Analytics, crear un proyecto dentro de este, y crear una clave privada para nuestra cuenta de google en este proyecto. Esta clave se puede crear en dos formatos, de entre los cuales se ha elegido el formato .json.

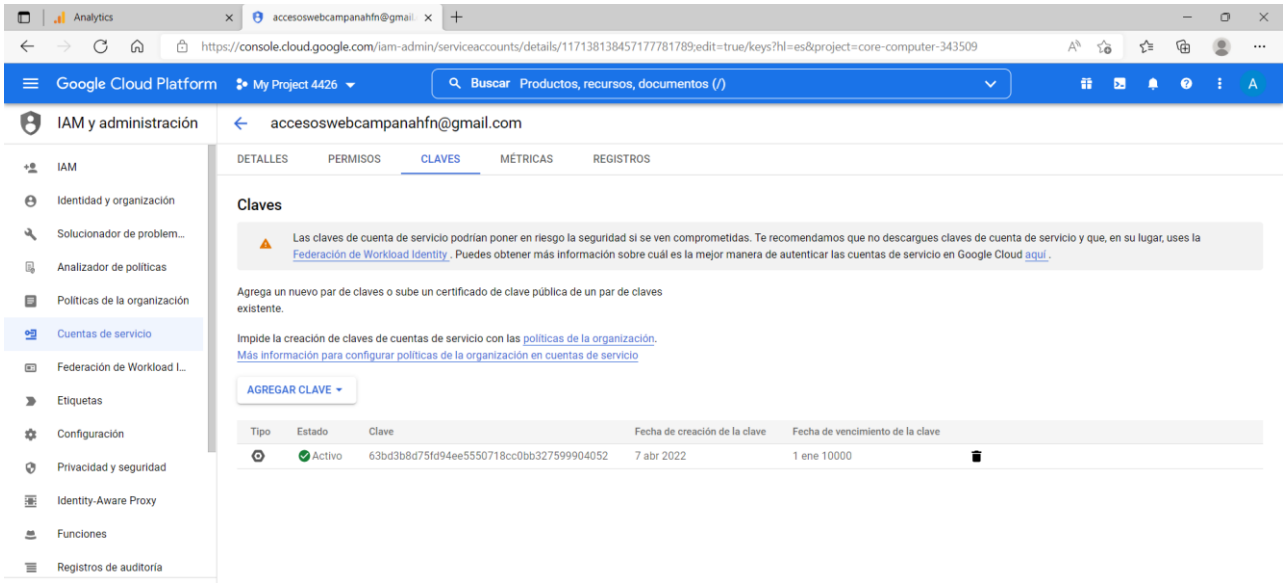


Figura 18: Claves cuenta GA

Algo también que destacar es que cuando se crea esta cuenta de google, a su vez se crea algo llamado "cuenta del automatismo". Si se pretende que todo funcione, hay que dar permisos en Google Analytics tanto a la cuenta normal como a la cuenta del automatismo.

Estas webs no han sido elegidas de forma aleatoria; han sido proporcionadas por personal del Gobierno de Navarra. El fin es estudiar el comportamiento del tráfico de estas.

Ahora bien, las únicas personas que pueden ver el tráfico que pasa por ellas pertenecen al Gobierno de Navarra. Para poder comenzar con el estudio, se ha creado

una cuenta de Google y se le han otorgado los permisos necesarios para ver en Google Analytics la información relativa a las urls.

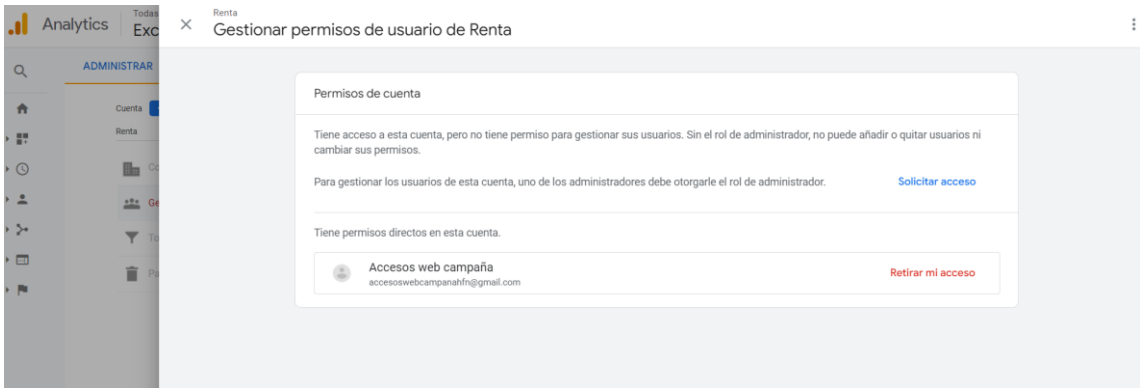


Figura 19: Permisos Google Analytics

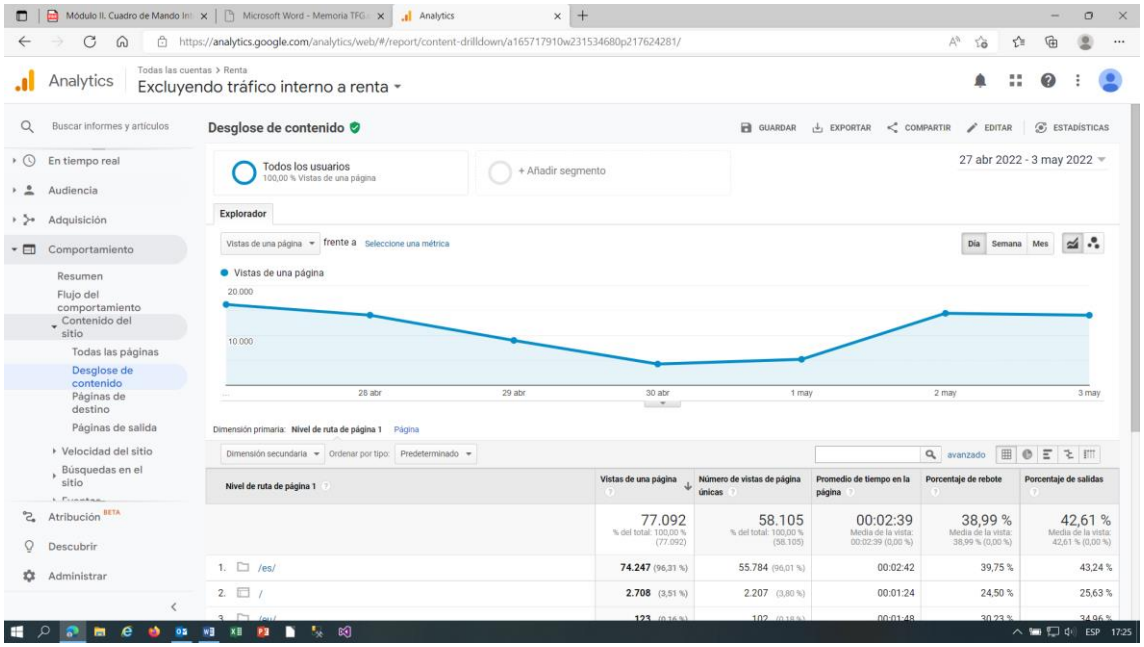


Figura 20: Ejemplo Visualización Google Analytics

3.3.3- Ordenar los datos en Microsoft SQL Server

Gracias a los filtros empleados en el script Python, se ha visto que las urls guardan patrones comunes en su estructura. Es por ello por lo que se han empleado tres filtros combinados con la estructura OR, para que el script capture todas las urls que cumplen al menos uno de los 3 filtros. Gracias a lo anterior, se ha podido hacer una gran criba de entre todos los datos disponibles en el periodo temporal del estudio. Esta fase ha sido fundamental de cara al posterior tratamiento de los datos.

3.3.4- Tableau

Una vez en Tableau, lo único que queda sería pasar la información a gráficos. Por cada indicador, se va a emplear un gráfico distinto, los cuales se agruparán finalmente en un cuadro de mando para darle sentido a todo y ver como se relaciona la información en conjunto para poder sacar conclusiones.

3.3.5- Análisis de los datos obtenidos

Gracias al cuadro de mando que se puede ver en las figuras de abajo, se han podido ir extrayendo las siguientes conclusiones:

- Como se puede ver en la figura 21, de entre las webs a estudiar, la mayor parte del tráfico (visitas totales) está repartida entre 3:

    - /es/tramites/ayuda-para-tramitar-por-internet/clave: **141.310 visitas**
    - /es/tramites/ayuda-para-tramitar-por-internet/certificado-digital: **104.410 visitas**
    - /es/tramites/on/-/line/Cita-previa-Hacienda-Foral-de-Navarra: **89.541 visitas**

    La interpretación que se le da a esto es que el ciudadano ha tenido bastantes dudas/problemas en la última campaña con el tema del registro, ya que suele ser o con el sistema “clave”, o con un certificado digital.

- En cuanto a las visitas únicas (figura 22), se aprecia que sigue el mismo patrón que el de las visitas totales, siendo algo inferior en número, obviamente.

- Respecto al porcentaje de rebote (figura 25), se da que todas las webs rondan el 45 – 50 % de rebote, lo cual es lo habitual. Hay un par de excepciones, una que se aleja mucho por encima y otra que se aleja mucho por debajo:

    - /es/tramites/on/-/line/Gestion-del-PIN?back=true&pageBackId=5722676: **27,04 %**
    - /es/tramites/on/-/line/cita-previa-para-hacer-la-declaracion-de-la-renta: **66,39 %**

  No obstante, por muy dispares que sean estos datos, ninguno está en situación de dar problemas. Mientras estos se sitúen entre el 20% y el 80%, no hay motivo por el cual preocuparse.

- Finalmente, en cuanto a la distribución temporal (figura 24), en todas las webs se ve cómo se da un pico muy alto al principio de la campaña, ya que es donde más agitada está la situación. Según qué url, puede aparecer algún otro pico por la mitad de la campaña, aunque poco significativo. Hay otro denominador común, que es el de un ligero repunte del tráfico justo al final de la campaña, ya que ahí siempre aparecen los últimos rezagados.

- Para el tiempo promedio (figura 23) en la web no hay mucho que mencionar, ya que todas las webs aportan valores comunes, entre los 100 y los 250 segundos.

VISITAS TOTALES

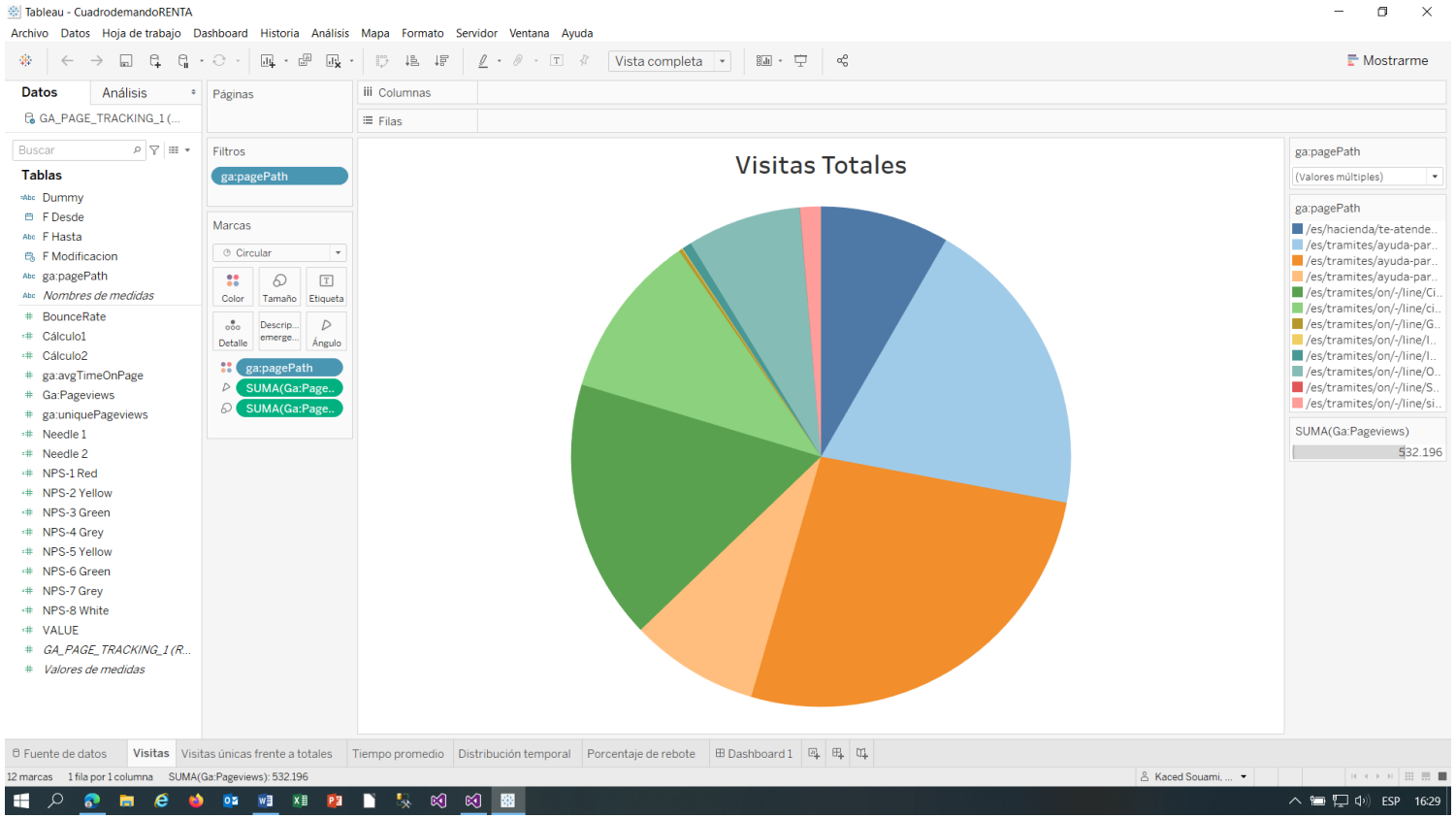


Figura 21: Claves cuenta GA

VISITAS ÚNICAS FRENTE A TOTALES

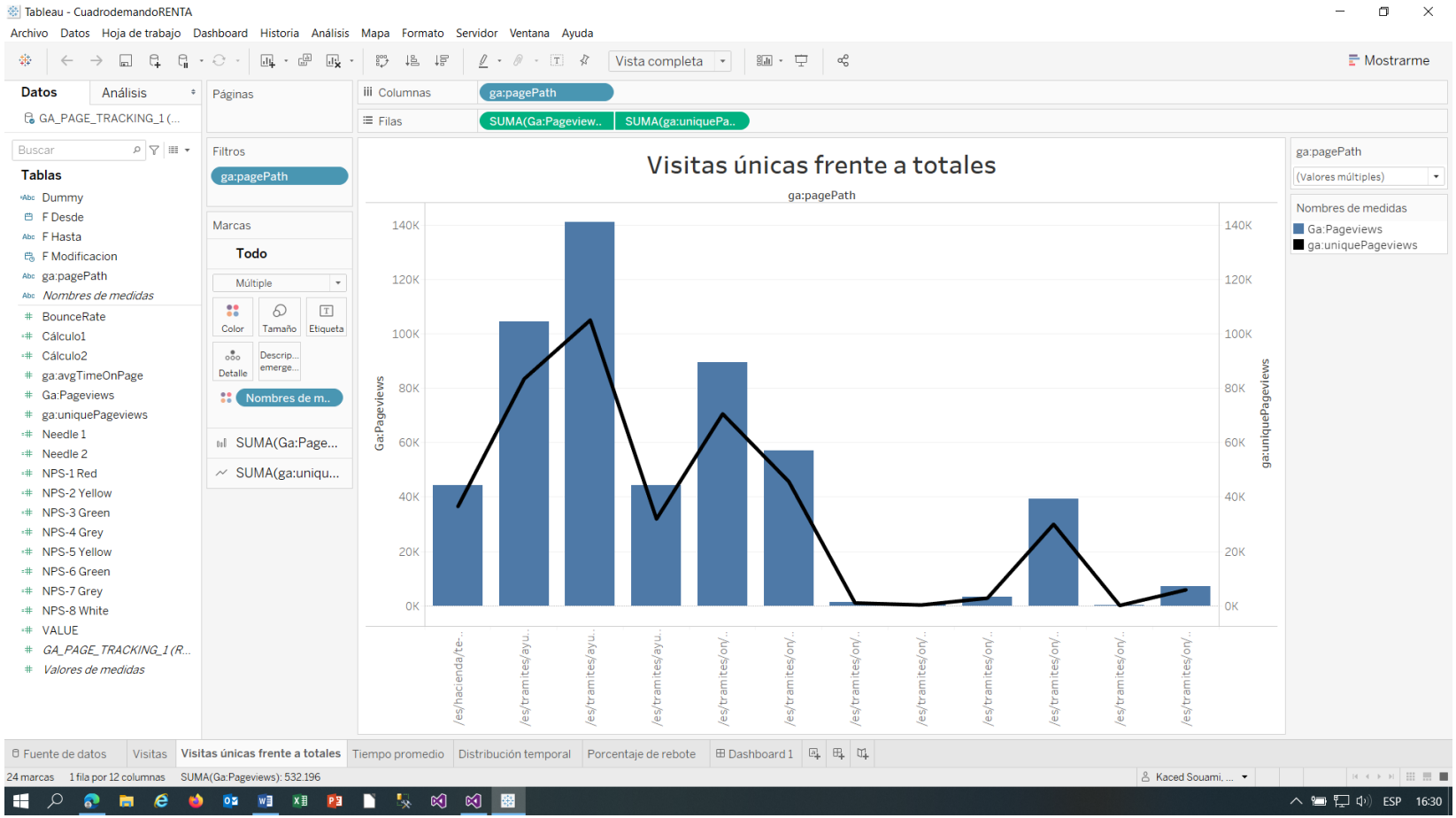


Figura 22: Claves cuenta GA

TIEMPO PROMEDIO

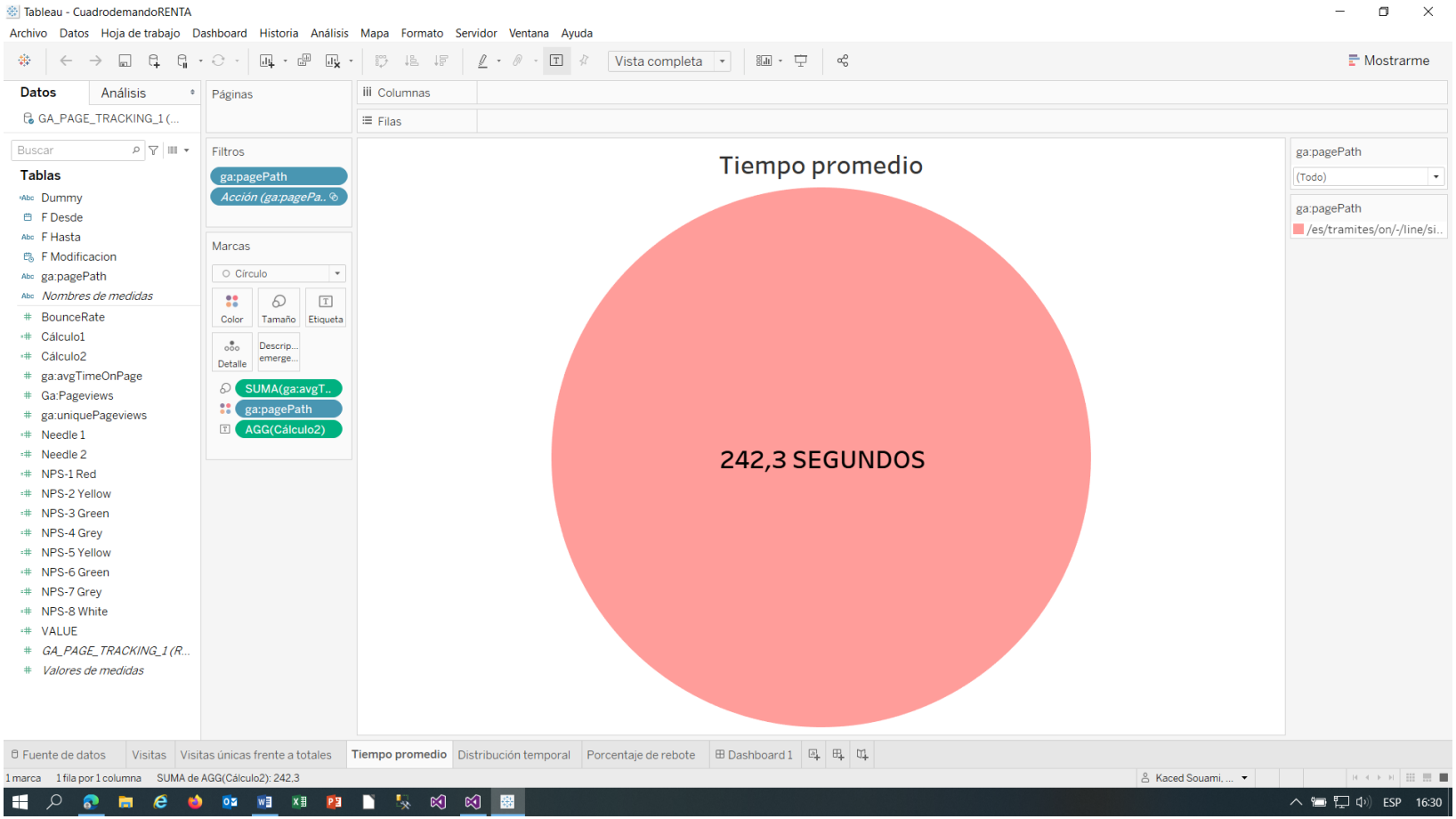


Figura 23: Claves cuenta GA

DISTRIBUCIÓN TEMPORAL

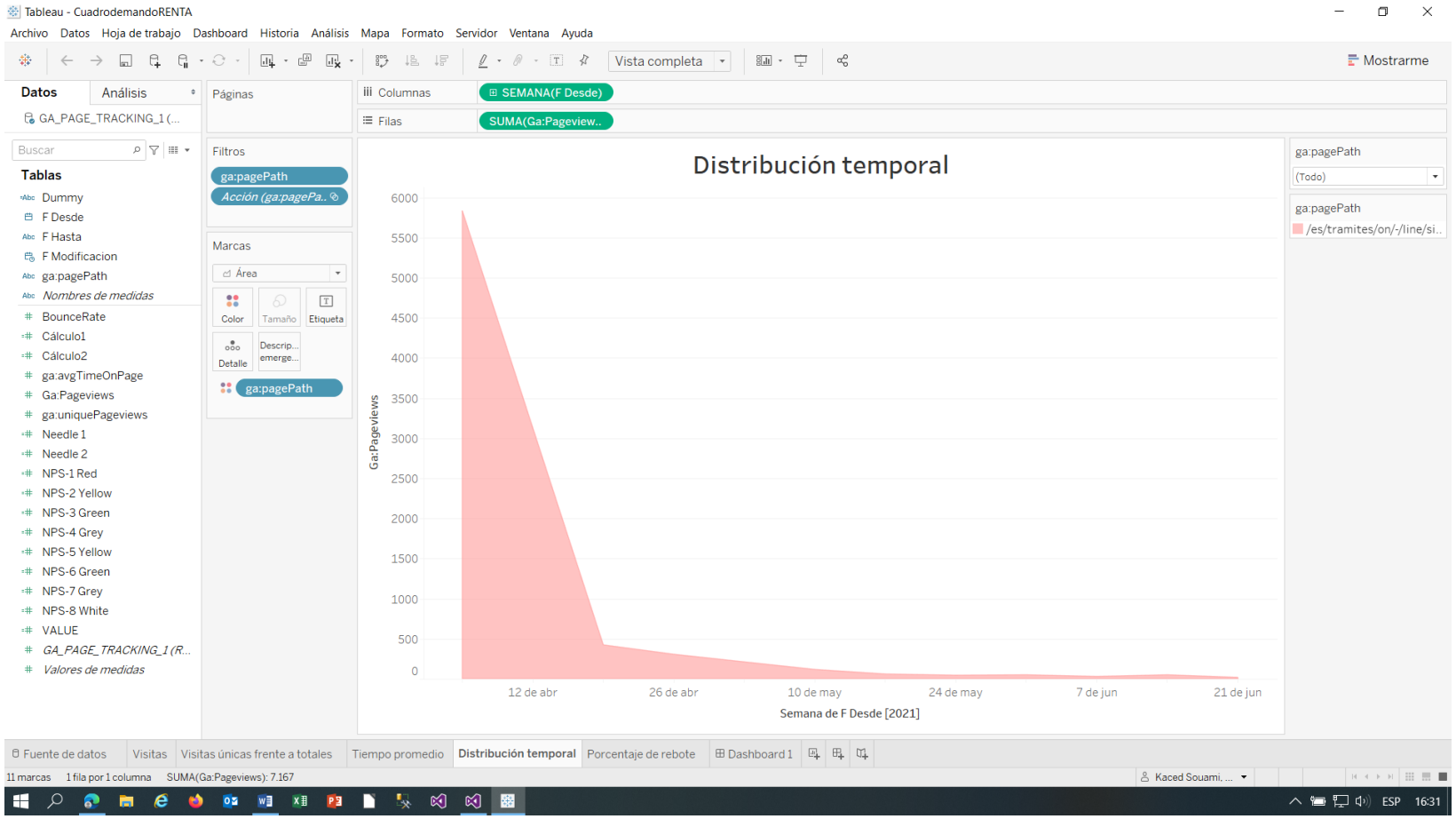


Figura 24: Claves cuenta GA

PORCENTAJE DE REBOTE

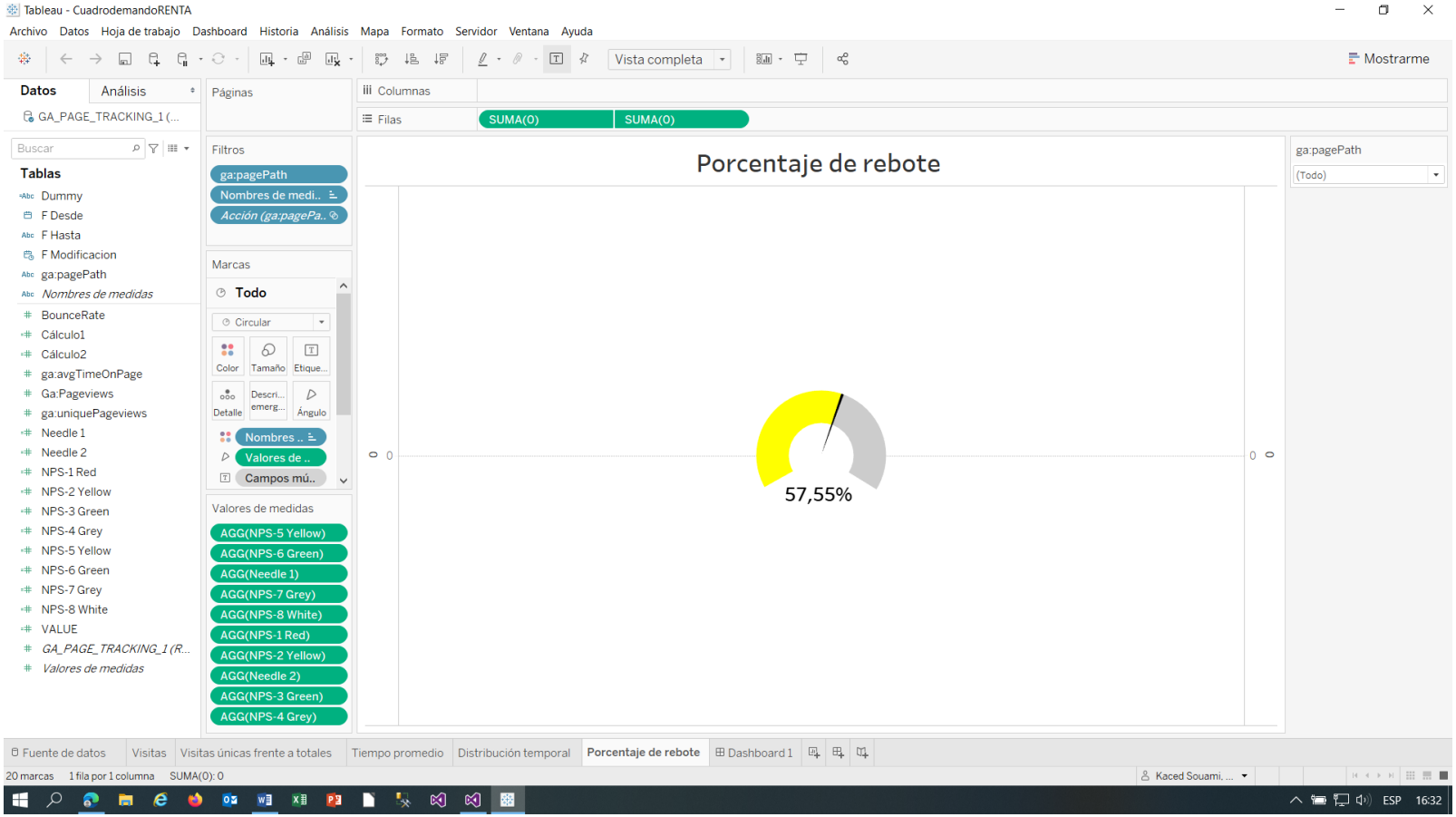


Figura 25: Claves cuenta GA

CUADRO DE MANDO

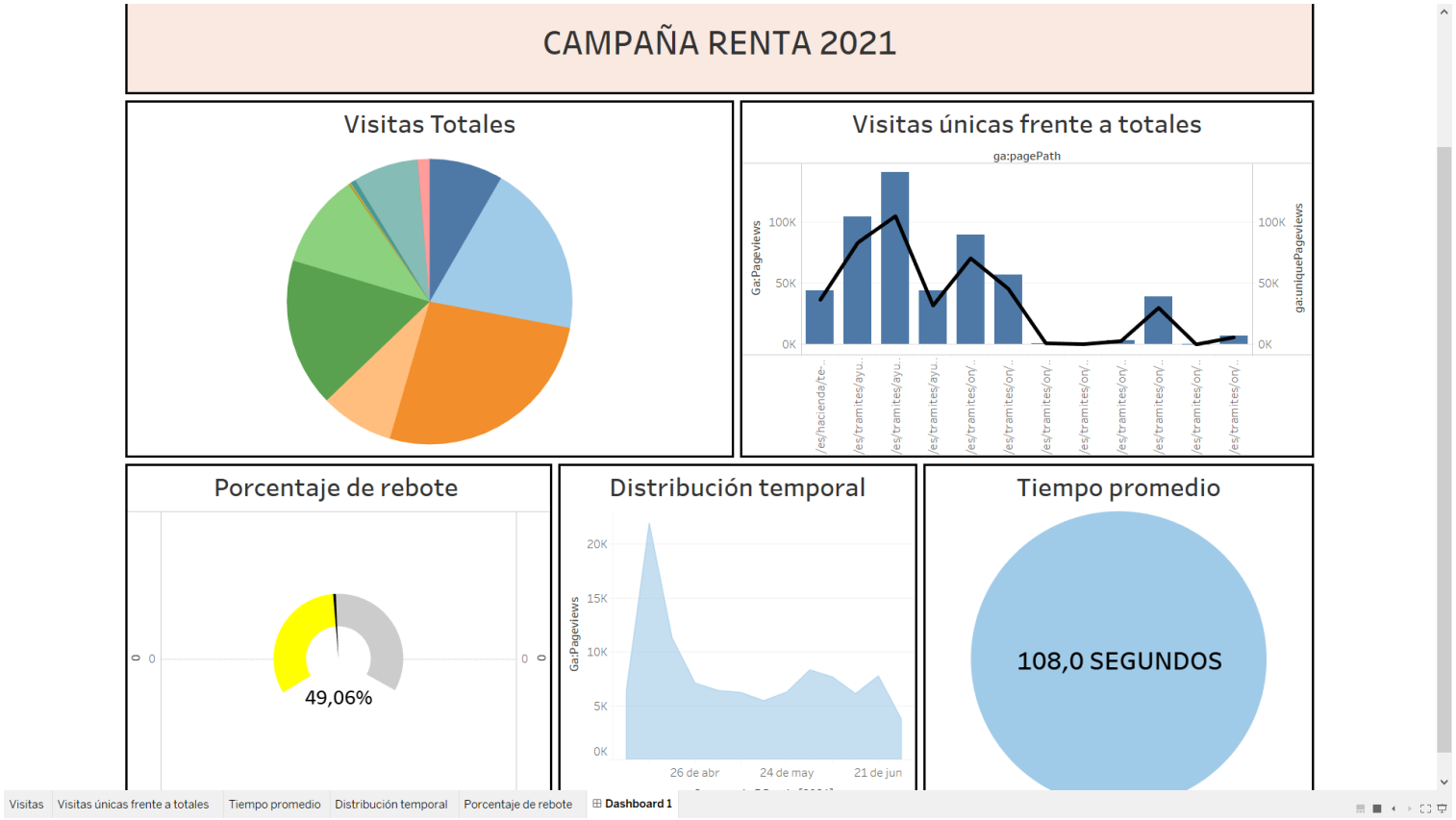


Figura 26: Claves cuenta GA

4.- PoC de Análisis Predictivo

El objetivo de este último apartado del TFG consiste en llevar a cabo un análisis predictivo en un ámbito relacionado con el Gobierno de Navarra, de modo que este les pueda servir en un futuro.

Se ha decidido emplear una regresión lineal relacionada con la calidad del aire en Navarra en los últimos años, mediante datos públicos del Gobierno de Navarra.

4.1- Tecnologías a utilizar

Python, MobaxTerm, Excel

4.2- Regresión Lineal

La regresión lineal es una técnica de modelado estadístico que se emplea para describir una variable de respuesta continua como una función de una o varias variables predictoras. Puede ayudar a comprender y predecir el comportamiento de sistemas complejos o a analizar datos experimentales, financieros y biológicos. [7]

La regresión lineal es una técnica paramétrica, de entre todas las utilizadas dentro del machine learning. El hecho de ser paramétrica indica al usuario que ya se conoce de antemano, previo al análisis de datos, el número de indicadores/parámetros o coeficientes que van a ser requeridos. Por ejemplo, con una sola variable, se sabe que una línea va a contar con dos parámetros. El modelo se expresa a través de la siguiente notación:

- Y es la variable dependiente o de respuesta.
- X representa las variables explicativas.
- B son los parámetros del modelo.

A pesar de que ya se conocen los indicadores necesarios, hay que saber elegir aquellos que minimicen alguna medida de error.

Por regla general, en la regresión lineal se utiliza el RMSE (error cuadrático medio). La fórmula que se usaría para una regresión lineal con una sola variable X es la siguiente:

$$y=wx+b$$

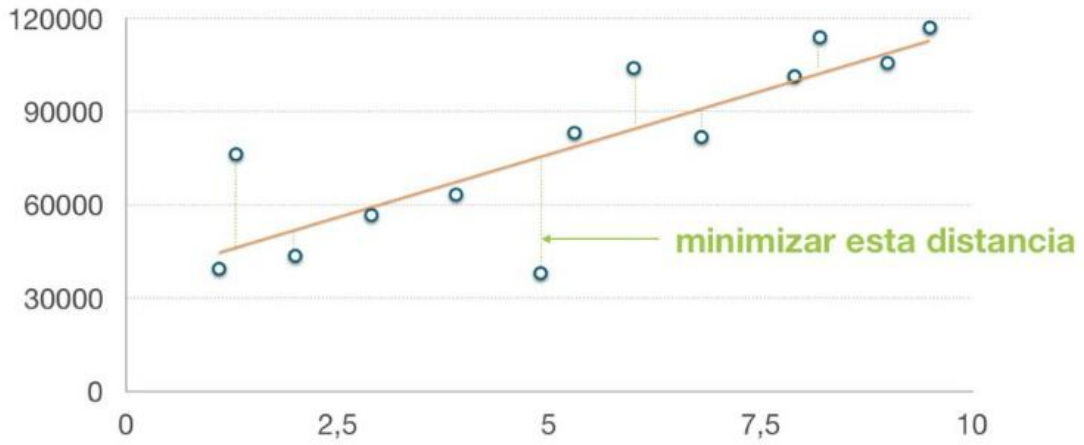


Figura 27: Regresión Lineal [8]

Dentro del campo de la inteligencia artificial, machine learning… Cabe destacar que la regresión lineal cumple la función de comprender y predecir el comportamiento de aquello que se esté estudiando. El modelo que se crea con esta regresión es lineal, es decir, asume una relación lineal entre la variable Y, que se conoce como respuesta, como función de una (análisis univariante) o varias (análisis multivariante) variables independientes X, denominadas predictores.

Las técnicas de regresión lineal están en múltiples ámbitos científicos tales como la medicina, química, electricidad, mecánica, física, construcción…

Estos son solo algunos de los ejemplos principales de campos de aplicación, aunque existen infinidad de sectores en los que se aplican las técnicas de. regresión lineal.

Dentro de este tipo de técnicas, se puede diferenciar entre 2 grandes tipos:

- Regresión lineal simple o univariante:

La regresión lineal simple trata de generar un modelo de regresión (ecuación de una recta) que permita explicar la relación lineal que existe entre dos variables X e Y. Un ejemplo podría ser el hecho de estudiar la relación entre la posesión y los goles de los 20 equipos de la liga, intentando explicar la relación (si es que la hay), entre la posesión del balón en un partido y los goles marcados.

- Regresión lineal múltiple o multivariante:

La regresión lineal múltiple trata de generar un modelo lineal en el que el valor de la variable dependiente o respuesta Y se determina a partir de un conjunto de variables independientes llamadas predictores (X1, X2, X3, X4…). Los modelos de regresión múltiple pueden emplearse para predecir el valor de la variable dependiente o bien para analizar la dependencia que tienen los predictores sobre ella.

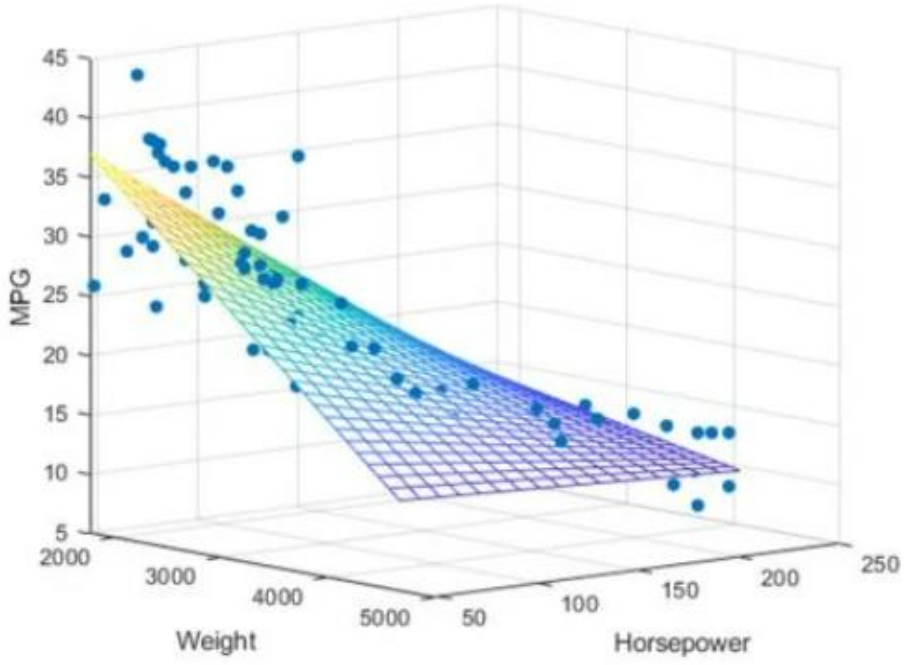


Figura 28: Regresión Lineal Multivariante [9]

Un ejemplo de análisis multivariante es el de analizar las ventas de una empresa en el año 2021 (esta sería la variable respuesta Y) en función de las variables X como el gasto en publicidad en televisión, el gasto en publicidad en las marquesinas, y el gasto en publicidad en la radio. Lo que permitiría esta regresión lineal multivariante sería ver cuáles de los predictores tiene más relación directa con la variable Y. De esta manera se podría saber qué publicidad es más rentable de cara a las ventas y qué publicidad es más residual.

Para que los resultados se consideren fiables en sentido estadístico, en función del tipo de análisis se deben cumplir una serie de hipótesis requeridas por el tipo de prueba, modelo o análisis estadístico, entre otras:

- No colinealidad
- Parsimonia
- Distribución normal de los residuos
- Que no haya autocorrelación
- Tamaño muestral significativo
- …

4.3- ARIMA

El estudio a realizar va a consistir en la predicción y evaluación de la calidad del aire en Navarra. Para ello, los datos a consultar los proporciona el propio Gobierno de Navarra de forma abierta en : https://datosabiertos.navarra.es/es/dataset/calidad-del-aire-en-navarra.

En este enlace se tiene acceso a varios indicadores de la calidad del aire, en varias estaciones localizadas en Navarra. Por ser la única situada en Pamplona, se van a coger los datos de la estación situada en Iturrama. De entre todos los indicadores de la calidad del aire proporcionados, se van a elegir el PM10 (partículas en suspensión cuyo diámetro aerodinámico no excede de las 10 micras), el O3 (ozono), y el NO2 (dióxido de nitrógeno), por ser las más representativas.

La idea consiste en hacer una regresión lineal simple (univariante) de cada uno de los 3 indicadores, con el objetivo de comprobar si el modelo funciona correctamente, poniéndolo a prueba.

Para ello, los datos a emplear son los comprendidos entre los años 2017 – 2020 (ambos incluidos), del PM10, O3 y NO2.

Antes de profundizar en el estudio, hay que elegir un modelo de regresión lineal que se adapte a las características del caso. Y es que, de entre todas las características de este caso, hay una bastante importante para el análisis temporal: la estacionariedad. Como se van a representar los datos de 4 años seguidos, en teoría debería haber un cierto comportamiento repetitivo cada año. Esto es una mera suposición, que se intentará corroborar posteriormente.

Es por todo esto argumentado anteriormente, que se ha decidido emplear un modelo ARIMA univariante. Se ha elegido emplear al modelo ARIMA como "baseline model" teniendo en cuenta que es un modelo para series temporales con un nivel de interpretabilidad mayor en comparación a otros modelos temporales.

En estadística y econometría, en particular en series temporales, un modelo autorregresivo integrado de promedio móvil o ARIMA (acrónimo del inglés autoregressive integrated moving average) es un modelo estadístico que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para una predicción hacia el futuro. Se trata de un modelo dinámico de series temporales, es decir, las estimaciones futuras vienen explicadas por los datos del pasado y no por variables independientes. [10]

ARIMA es un modelo estadístico empleado en ámbitos de predicción de series temporales. En la ecuación que rige este tipo de modelado, las variables independientes son retrasos de las variables dependientes y retrasos de los errores en la predicción. Dicha ecuación es la siguiente:

$$y'(t) = c + \varphi 1 * y'(t-1) + \cdots + \varphi p * y'(t-p) + \theta 1 * \varepsilon(t-1) + \cdots + \theta q * \varepsilon(t-q) + \varepsilon t$$

Hay 3 términos en la ecuación:

- **Auto Regresión:** Primero, la serie temporal es regresada mediante sus valores previos. y(t-1), y(t-2), y(t-p) ... El parámetro que mide el orden de retraso, o número de muestras de holgura es llamado p.

- **Integración:** Este apartado hace referencia al número de diferencias que necesita la serie temporal para hacer que sea estacionaria (posteriormente se explicará). El parámetro que mide este número de modificaciones se llama d.

- **Moving Average:** El parámetro que se encarga de medir el orden del retraso de los errores.

En la ecuación de arriba, **y't** es la serie temporal ya modificada (estacionaria), **φ1** es el coeficiente del primer término AR, **p** es su orden, **θ1** es el coeficiente del primer término MA, **q** es el orden de este y **εt** es el error.

Antes de llevar a cabo la predicción con ARIMA, hay que comprobar que la serie temporal cumple ciertas condiciones (estacionariedad en sentido estricto):

- Tiene una media constante

- Tiene una varianza constante

- Tiene una covarianza constante

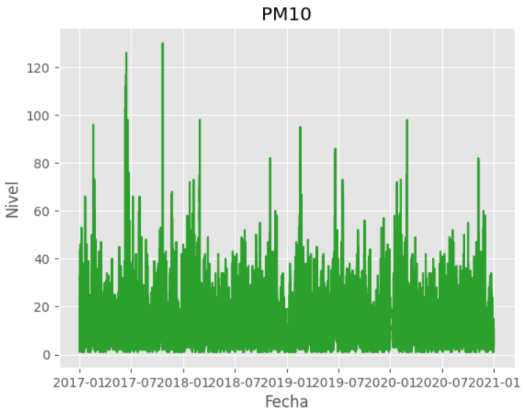


Figura 29: Serie temporal PM10

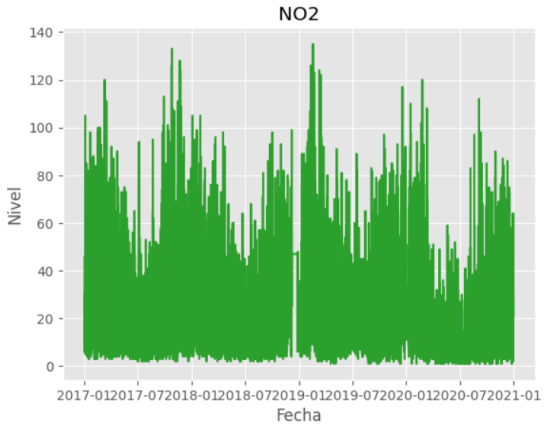


Figura 30: Serie temporal NO2

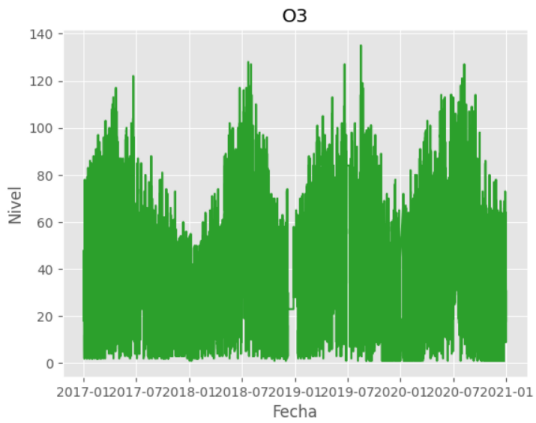


Figura 31: Serie temporal O3

Como se puede ver en las 3 figuras de arriba, estas series temporales tienen una estacionariedad cuanto menos dudosa. A simple vista es bastante difícil, por no decir imposible, corroborarlo a ciencia cierta. Python ofrece, entre muchas otras cosas, 2 test para corroborar al 100% si es estacionaria o no una serie temporal. Estos son:

- **Augmented Dickey Fuller Test:**

Si el p-valor no pasa de 0,05, se concluye que la serie temporal es estacionaria. De lo contrario, deduciremos que sigue sin ser estacionaria.

- **Kwiatkowski-Phillips-Schmidt-Shin Test:**

Si el p-valor excede de 0,05, se concluye que la serie temporal es estacionaria. De lo contrario, deduciremos que sigue sin ser estacionaria.

Para cada serie temporal, habrá que llevar a cabo ambas pruebas. Dicho esto, pueden darse 4 posibles escenarios:

1. Ambas pruebas niegan la estacionariedad: se concluye que la serie temporal sigue sin ser estacionaria.

2. Ambas pruebas concluyen estacionariedad: obviamente, significa que la serie temporal es estacionaria.

3. El resultado de una prueba contradice el del otro: la serie temporal sigue sin ser estacionaria.

Dicho todo lo anterior, se procede en primer lugar a comprobar la estacionariedad de cada una de las 3 series temporales.

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
ADF Statistic: -11.788510
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Results of KPSS Test:
Test Statistic            0.832006
p-value                   0.010000
Lags Used                50.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$
```

Figura 32: Resultado Test PM10

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
ADF Statistic: -14.755640
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Results of KPSS Test:
Test Statistic            1.630394
p-value                   0.010000
Lags Used                97.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$
```

Figura 33: Resultados Test NO2

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
ADF Statistic: -14.051240
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Results of KPSS Test:
Test Statistic            0.717601
p-value                   0.011945
Lags Used                97.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$
```

Figura 34: Resultados Test O3

Como se puede ver en las 3 figuras, se da el caso de que ninguna serie temporal es estacionaria. Es más, se está dando el tercer caso. La prueba ADF indica que las series temporales son estacionarias, mientras que la prueba KPSS contradice lo dicho por el primero. Como se ha dicho anteriormente, no se puede proceder con la regresión lineal sin ser la serie temporal estacionaria.

Entonces, hay que transformar estas de alguna manera para "convertirlas" en estacionarias. Dado que en estas series temporales hay una clara estacionalidad, se va a hacer un diferenciado estacional.

Así quedarían estas nuevas series temporales:

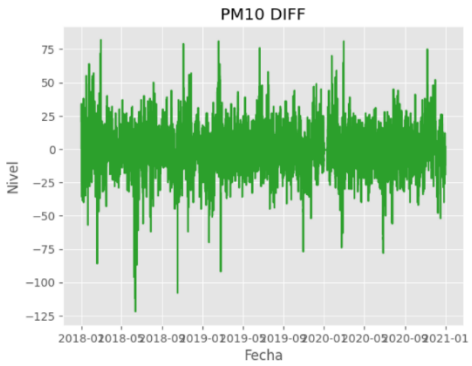


Figura 35: Serie temporal PM10 dif

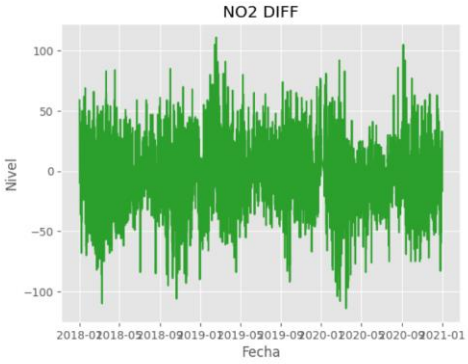


Figura 36: Serie temporal NO2 dif

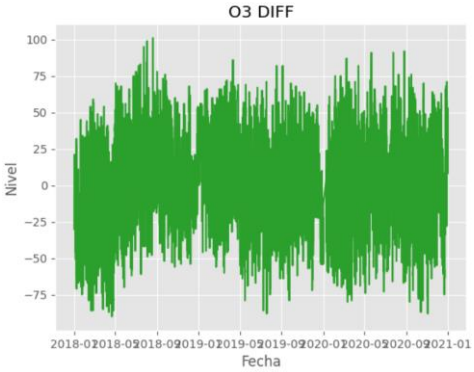


Figura 37: Serie temporal O3 dif

A priori, ya tienen un aspecto que parece bastante más estacionario, aunque la única manera de comprobarlo es mediante las dos pruebas:

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
ADF Statistic: -15.976083
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Results of KPSS Test:
Test Statistic            0.063036
p-value                   0.100000
Lags Used                86.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$
```

Figura 38: Resultados Test PM10 dif

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
ADF Statistic: -31.884675
p-value: 0.000000
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Results of KPSS Test:
Test Statistic            0.041227
p-value                   0.100000
Lags Used                677.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
dtype: float64
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$
```

Figura 39: Resultados Test NO2 dif

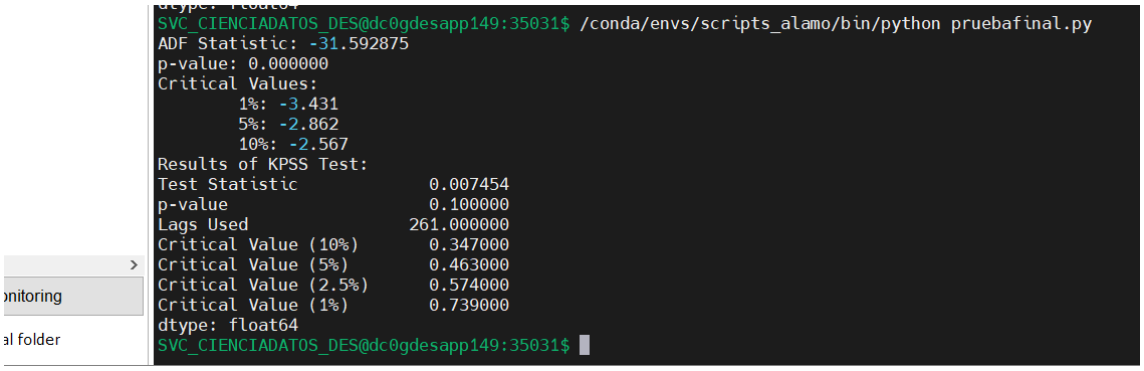


Figura 40: Resultados Test O3 dif

Cabe destacar, antes que nada, que tanto para el NO2 como para el O3 han hecho falta 2 transformaciones; la primera es la que ya se ha aclarado y la segunda ha consistido en volver a restarle su propia serie temporal, pero esta vez con un desfase de 1 muestra, y no de 8760 como antes, ya que, con solo la primera transformación, el resultado de las pruebas seguía siendo contradictorio.

Como se puede ver en estas series ya transformadas, ahora se ha pasado al escenario 2, en el que ambas pruebas corroboran lo que se ha podido deducir visualmente: las 3 series temporales ya se pueden considerar de carácter estacionario, y, por tanto, ya se puede proceder con la regresión.

El modelo ARIMA a emplear va a ser el ARIMA (1,1,1), cuyos parámetros se han elegido siguiendo las definiciones explicadas anteriormente en la teoría. Antes que nada, se va a ver que cumple los requisitos. Otra manera de comprobar que este era el modelo más adecuado, ha sido con un método prueba-error en el que siempre se calculaba el RSME al final de la predicción. Con ARIMA (1,1,1) siempre se ha logrado el mejor RMSE para los 3 casos. También es posible que el propio Python haga el trabajo de hallar el mejor ajuste, pero al tener series temporales de periodo 8760, el programa tardaría una eternidad en hallarla.

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
                               SARIMAX Results
==========================================================================
Dep. Variable:     Iturrama PM10_diff   No. Observations:              26279
Model:                 ARIMA(1, 1, 1)   Log Likelihood           -89502.353
Date:                Sun, 22 May 2022   AIC                     179010.706
Time:                        23:31:50   BIC                     179035.235
Sample:                             0   HQIC                    179018.626
                                - 26279
Covariance Type:                  opg
==========================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------
ar.L1          0.7449      0.004    169.277      0.000       0.736       0.754
ma.L1         -0.9505      0.002   -405.956      0.000      -0.955      -0.946
sigma2        53.1995      0.218    243.986      0.000      52.772      53.627
==========================================================================
Ljung-Box (L1) (Q):                  60.12   Jarque-Bera (JB):             57136.60
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.89   Skew:                            -0.42
Prob(H) (two-sided):                  0.00   Kurtosis:                        10.17
==========================================================================
```

Figura 41: Resumen ARIMA PM10

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
                               SARIMAX Results
==========================================================================
Dep. Variable:     Iturrama PM10_diff   No. Observations:              26279
Model:                 ARIMA(1, 1, 1)   Log Likelihood           -89502.353
Date:                Sun, 22 May 2022   AIC                     179010.706
Time:                        23:34:49   BIC                     179035.235
Sample:                             0   HQIC                    179018.626
                                - 26279
Covariance Type:                  opg
==========================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------
ar.L1          0.7449      0.004    169.277      0.000       0.736       0.754
ma.L1         -0.9505      0.002   -405.956      0.000      -0.955      -0.946
sigma2        53.1995      0.218    243.986      0.000      52.772      53.627
==========================================================================
Ljung-Box (L1) (Q):                  60.12   Jarque-Bera (JB):             57136.60
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.89   Skew:                            -0.42
Prob(H) (two-sided):                  0.00   Kurtosis:                        10.17
==========================================================================
```

Figura 42: Resumen ARIMA NO2

```
SVC_CIENCIADATOS_DES@dc0gdesapp149:35031$ /conda/envs/scripts_alamo/bin/python pruebafinal.py
                               SARIMAX Results
==========================================================================
Dep. Variable:     Iturrama PM10_diff   No. Observations:              26279
Model:                 ARIMA(1, 1, 1)   Log Likelihood           -89502.353
Date:                Sun, 22 May 2022   AIC                     179010.706
Time:                        23:47:46   BIC                     179035.235
Sample:                             0   HQIC                    179018.626
                                - 26279
Covariance Type:                  opg
==========================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------
ar.L1          0.7449      0.004    169.277      0.000       0.736       0.754
ma.L1         -0.9505      0.002   -405.956      0.000      -0.955      -0.946
sigma2        53.1995      0.218    243.986      0.000      52.772      53.627
==========================================================================
Ljung-Box (L1) (Q):                  60.12   Jarque-Bera (JB):             57136.60
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.89   Skew:                            -0.42
Prob(H) (two-sided):                  0.00   Kurtosis:                        10.17
==========================================================================
```

Figura 43: Resumen ARIMA O3

Como se puede observar en las figuras 41-43, los p-valores de los coeficientes AR y MA no exceden de 0,05, por lo que es de deducir que se puede seguir adelante con el modelo.

En las figuras de las predicciones, se puede ver en rojo la regresión lineal calculada mediante Python, y en azul la serie temporal original. El análisis de los resultados e interpretación de las figuras correspondientes se desarrollará en el siguiente apartado (4.3).

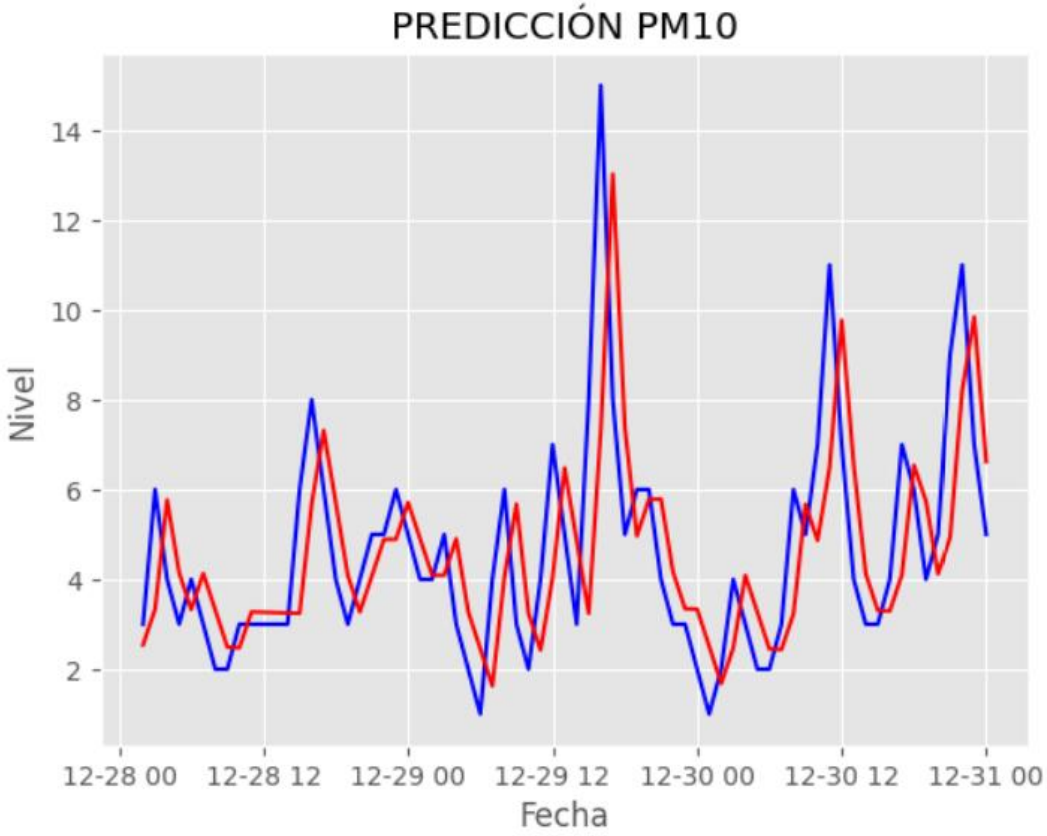


Figura 44: Predicción PM10 (Rojo: predicción, azul: original)

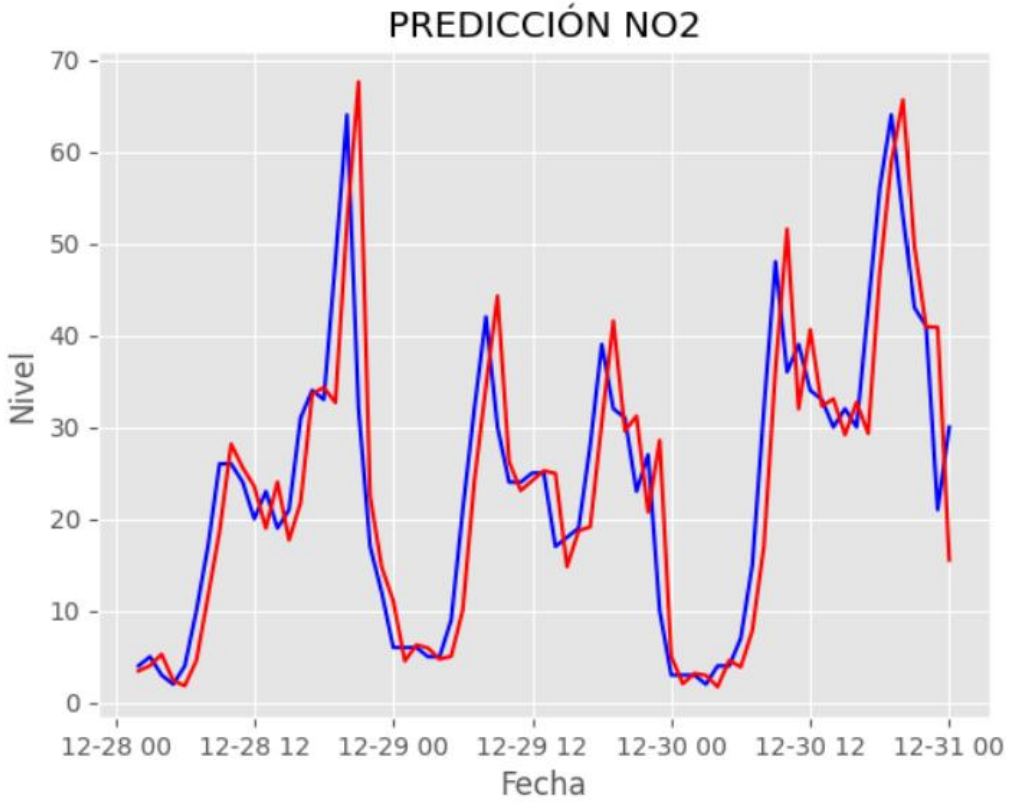


Figura 45: Predicción NO2 (Rojo: predicción, azul : original)

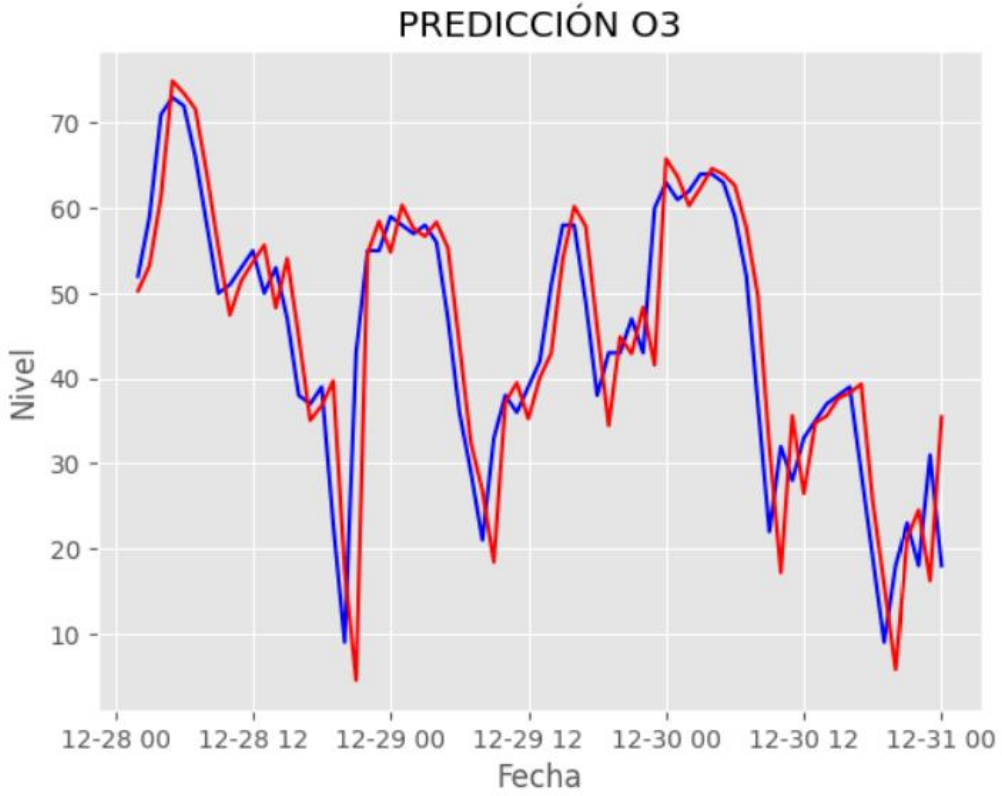


Figura 46: Predicción O3 (Rojo: predicción, azul : original)

4.4- Análisis de los resultados obtenidos

En las figuras de las predicciones, se puede ver en rojo la regresión lineal calculada mediante Python, y en azul la serie temporal original. Se han representado unas 70 muestras, que se corresponden con un periodo comprendido entre 3 y 4 días, elegidos de forma aleatoria por Python.

El RMSE proporcionado es del orden de entre 2 y 3 para las 3 predicciones, lo cual es un resultado bastante positivo, que indica que la diferencia entre la predicción y la serie temporal original es bastante baja. Visualmente, se puede apreciar como la predicción es bastante acertada (siendo el periodo temporal representado elegido de manera aleatoria). Sí que es cierto que le cuesta predecir bien los cambios bruscos, aunque también es de reconocer el hecho de que tampoco es que sea un método predictivo excesivamente rudimento y preciso.

En líneas generales, es muy sencillo apreciar como la predicción sigue de una manera bastante precisa el comportamiento de la serie temporal original, aunque como se ha dicho previamente, cuando hay cambios bruscos, sí que es verdad que le toma cierto tiempo, relativamente bajo, el retomar la tendencia de la serie temporal correspondiente.

En la figura 47 se ven ejemplos de las muestras exactas de la serie temporal como la de la predicción, siguiendo el modelo ARIMA (1,1,1), el mismo para los 3 indicadores ambientales.

Como conclusión final, teniendo en cuenta lo dicho anteriormente y todo el proceso explicado y argumentado capítulo, se podría afirmar con gran seguridad que se trata de un modelo de regresión lineal bastante idóneo si lo que se pretende es tener una idea de cómo puede comportarse cierta serie temporal en un futuro cercano de manera bastante acertada, aunque también es noble admitir que si lo que se pretende es una gran precisión (sobre todo en temas críticos como por ejemplo algo relacionado con la salud), no es del todo recomendable.

'ADATOS_DES)

```
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
predicted=3.450039, expected=4.000000
predicted=4.030875, expected=5.000000
predicted=5.269293, expected=3.000000
predicted=2.395998, expected=2.000000
predicted=1.843961, expected=4.000000
predicted=4.581623, expected=10.000000
predicted=11.537457, expected=17.000000
predicted=18.624069, expected=26.000000
predicted=28.158001, expected=26.000000
predicted=25.573646, expected=24.000000
predicted=23.533395, expected=20.000000
predicted=18.990477, expected=23.000000
predicted=24.025724, expected=19.000000
predicted=17.695717, expected=21.000000
predicted=21.808450, expected=31.000000
predicted=33.594415, expected=34.000000
predicted=34.313601, expected=33.000000
predicted=32.662614, expected=48.000000
predicted=52.197850, expected=64.000000
predicted=67.577962, expected=32.000000
predicted=22.482314, expected=17.000000
predicted=14.757374, expected=12.000000
predicted=11.067766, expected=6.000000
predicted=4.532373, expected=6.000000
predicted=6.290881, expected=6.000000
predicted=5.942348, expected=5.000000
predicted=4.736031, expected=5.000000
predicted=5.052319, expected=9.000000
predicted=10.091215, expected=21.000000
predicted=24.088696, expected=32.000000
predicted=34.417817, expected=42.000000
predicted=44.275802, expected=30.000000
```

Figura 47: Muestras en la predicción

5.- Conclusiones

A modo de cierre de este trabajo, podemos señalar que se han podido llevar a cabo los 3 puntos propuestos al inicio de las prácticas con el Gobierno de Navarra, a pesar de las dificultades que han ido surgiendo a lo largo de este periodo.

El resultado más importante de este estudio ha sido la regresión lineal ARIMA realizada sobre la calidad del aire en Navarra, debido a la gran capacidad de predicción presentada por un modelo "simple" y accesible sin excesiva complejidad. Bien es cierto que este modelo lo entendemos para casos que no requieran una gran exactitud en la predicción. Debe haber un cierto margen de error, el cual se ha podido ver en los resultados de dicho apartado.

Asimismo, cabe destacar aquí la relevancia que tiene el cuadro de mando creado para la declaración de la renta. Sobre todo, por la utilidad otorgada al Gobierno de Navarra de cara a hacer un análisis visual, rápido y eficaz. También tiene el punto a favor de ser un cuadro de mando con datos en tiempo real, de modo que, si se pretendiesen analizar campañas de otros años, simplemente habría que cambiar el intervalo temporal de los datos.

En términos comparativos, se observa una escala de menor a mayor complejidad a lo largo del trabajo, empezando por una mera guía a modo de introducción, hasta llegar a algo más técnico y relacionado con la formación obtenida en los 4 años del grado.

La proyección más evidente de este estudio está en el último apartado; el análisis predictivo. Es un principio que se puede extrapolar a otros ámbitos con los que trata el Gobierno de Navarra, siempre teniendo en cuenta las limitaciones que este modelo regresivo presenta.

Además, se puede seguir avanzando y mejorando el cuadro de mando relativo a la declaración de la renta, ya que presenta ciertas carencias. Lo importante es que ya se tiene una base (el desarrollo del script y la secuencia a seguir) a partir de la cual ir avanzando. Sobre todo, de cara a temas como pueden ser Hacienda, Policía Foral… en los que interesa ir teniendo información de lo que va sucediendo en tiempo real. Y precisamente para ello, los cuadros de mando son la mejor opción.

6.- Bibliografía

[1] https://es.wikipedia.org/wiki/Inteligencia_empresarial

[2] Cuadros de mando integral (Anova Consulting)

[3] https://universitylift.com/cursos/herramientas-para-el-business-intelligence/

[4] https://cmigestion.es/2012/10/23/el-cuadro-de-mando-integral-i-que-es-y-para-que-sirve/

[5] https://evotic.es/business-intelligence-bi/cuadrante-magico-de-gartner-en-el-ambito-del-business-intelligence/

[6] https://www.baoss.es/cuadros-de-mando-presentacion-y-visualizacion-de-datos/

[7] https://aprendeia.com/algoritmo-regresion-lineal-simple-machine-learning/

[8] https://blog.structuralia.com/ejemplos-de-regresion-lineal

[9] https://la.mathworks.com/discovery/linear-regression.html

[10] https://es.wikipedia.org/wiki/Modelo_autorregresivo_integrado_de_media_m%C3%B3vil#:~:text=En%20estad%C3%ADstica%20y%20econometr%C3%ADa%2C%20en,de%20encontrar%20patrones%20para%20una