

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE MORELIA

“José María Morelos y Pavón”

Análisis De Grandes Volúmenes De Datos

Profesor: Ferreira Escutia Rogelio

Proyecto Final

MACAVE STADISTICS

Equipo 5

Carlos Jahir Castro Cazares

Giovanni Hasid Martinez Resendiz

Jaime Isai Velazquez Aguilar

Semestre Marzo-Julio 2021

09 de Julio de 2021

Contenido

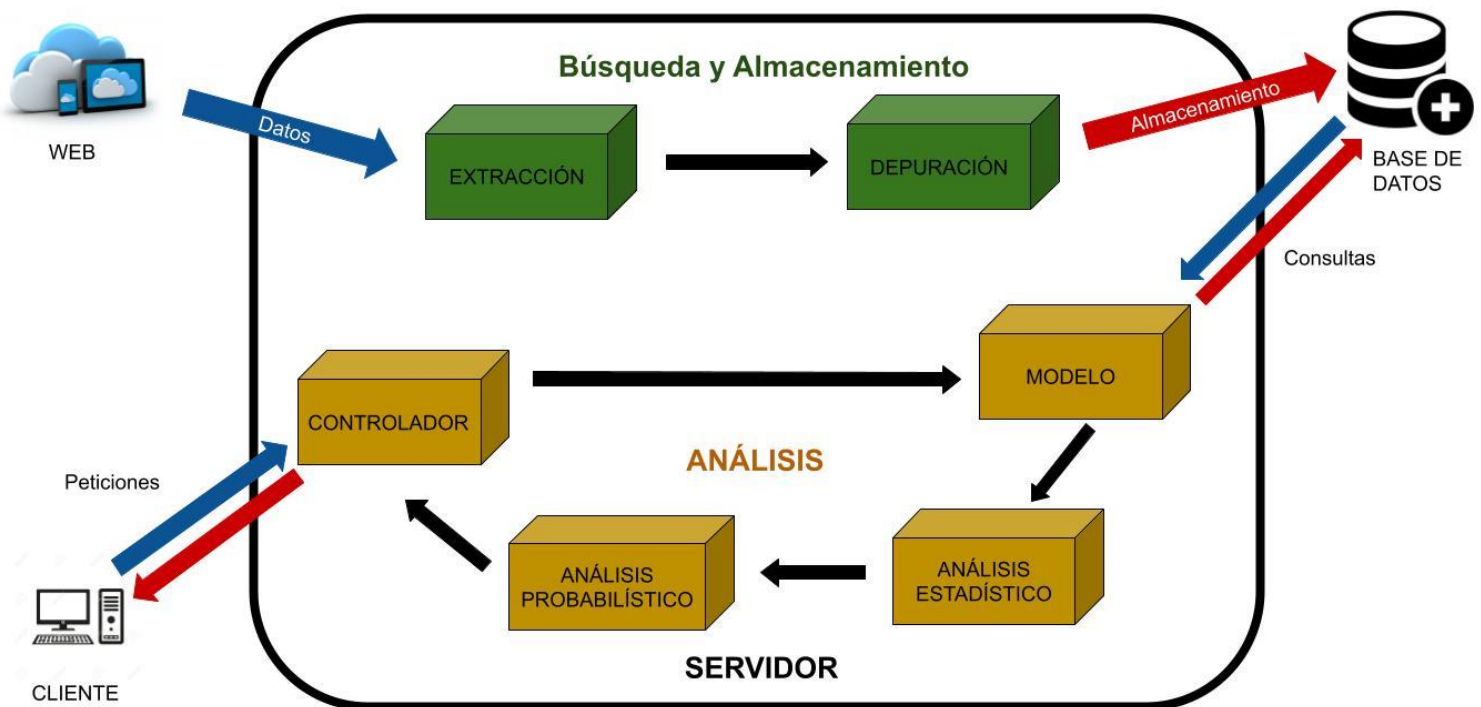
Descripción del proyecto.....	3
Diagrama a bloques del sistema.	3
Estructura de las bases de datos y tablas empleadas.	4
Punto 1: Adquisición de Datos	5
Punto 2: Almacenamiento.....	8
Punto 3: Algoritmo	13
Punto 4: Visualización	20
Punto 5: Complejidad e Impacto.....	26
Conclusiones y resultados del proyecto.....	27

Descripción del proyecto.

MACAVE STADISTICS, es un proyecto con el cual implementamos la búsqueda y recopilación de datos, para obtener un análisis estadístico y probabilístico de la información, además de implementar tecnologías web para mostrar los posibles resultados de los partidos de futbol de la Eurocopa. Esto da solución a un problema de falta de confiabilidad o no tener una base certera a la hora de hacer inversiones en partidos de futbol.

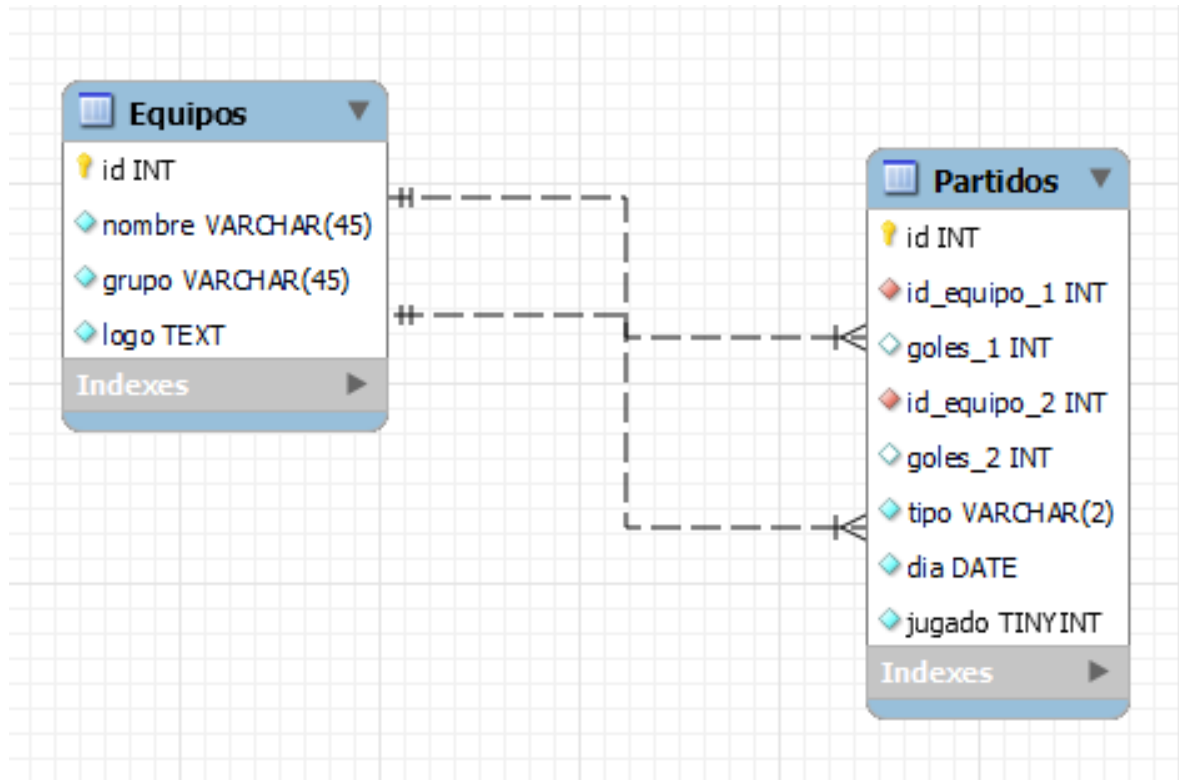
Diagrama a bloques del sistema.

En el siguiente diagrama de bloques se muestra la estructura básica e interna del sistema con el cual recopila, almacena y procesa información.



Estructura de las bases de datos y tablas empleadas.

Esta es la estructura de la base de datos que usamos para almacenar la información de los equipos y los resultados de los partidos. Estas tablas son de una base de datos relacional de mariaDB.

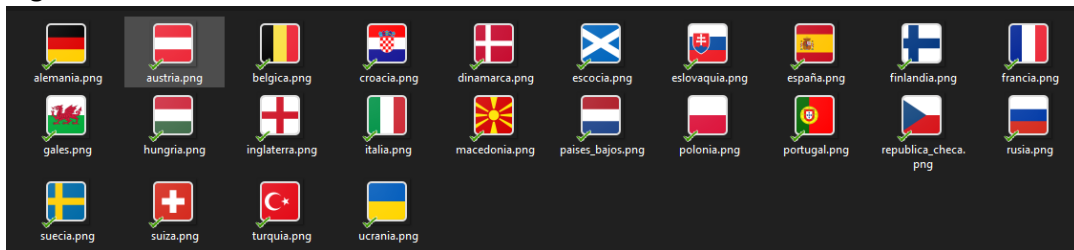


Punto 1: Adquisición de Datos

Para la adquisición de datos usamos, dos archivos csv, que contiene la información de los equipos participantes de la copa y los resultados de los partidos que se jugaron hasta el momento de descarga del dataset (06/07/2021); Estos archivos los alteramos para ingresar nuevos datos que nos ayudaran a diferenciar los datos. El dataset fue creado de la siguiente página:<https://www.google.com/search?q=Eurocopa+2020&oq=eurocopa+2020&ags=chrome.0.69i59.3664j0j1&sourceid=chrome&ie=UTF-8#sie=lg;/m/0p3p7v6;2;/m/01l10v;mt;fp;1;;>

Dataset

- logos



- equipos.csv

```
equipos.csv X
equipos.csv
1 id,nombre,grupo,logo
2 1,Italia,A,/logos/italia.png
3 2,Gales,A,/logos/gales.png
4 3,Suiza,A,/logos/suiza.png
5 4,Turquia,A,/logos/turquia.png
6 5,Belgica,B,/logos/belgica.png
7 6,Dinamarca,B,/logos/dinamarca.png
8 7,Finlandia,B,/logos/finlandia.png
9 8,Rusia,B,/logos/rusia.png
10 9,Países Bajos,C,/logos/paises_bajos.png
11 10,Austria,C,/logos/austria.png
12 11,Ucrania,C,/logos/ucrania.png
13 12,Macedonia del Norte,C,/logos/macedonia.png
14 13,Inglaterra,D,/logos/inglaterra.png
15 14,Croacia,D,/logos/croacia.png
16 15,República Checa,D,/logos/republica_checa.png
17 16,Escocia,D,/logos/escocia.png
18 17,Suecia,E,/logos/suecia.png
19 18,España,E,/logos/españa.png
20 19,Eslovaquia,E,/logos/eslovaquia.png
21 20,Polonia,E,/logos/polonia.png
22 21,Francia,F,/logos/francia.png
23 22,Alemania,F,/logos/alemania.png
24 23,Portugal,F,/logos/portugal.png
25 24,Hungria,F,/logos/hungria.png
26
```

- partidos.csv

	partidos.csv	X
	partidos.csv	
1	id,id_equipo_1,goles_1,id_equipo_2,goles_2,tipo,dia,jugado	
2	1,4,0,1,3,J1,2021-06-11,1	
3	2,2,1,3,1,J1,2021-06-12,1	
4	3,6,0,7,1,J1,2021-06-12,1	
5	4,5,3,8,0,J1,2021-06-12,1	
6	5,13,1,14,0,J1,2021-06-13,1	
7	6,10,3,12,1,J1,2021-06-13,1	
8	7,9,3,11,2,J1,2021-06-13,1	
9	8,16,0,15,2,J1,2021-06-14,1	
10	9,20,1,19,2,J1,2021-06-14,1	
11	10,18,0,17,0,J1,2021-06-14,1	
12	11,24,0,23,3,J1,2021-06-15,1	
13	12,21,1,22,0,J1,2021-06-15,1	
14	13,7,0,8,1,J2,2021-06-16,1	
15	14,4,0,2,2,J2,2021-06-16,1	
16	15,1,3,3,0,J2,2021-06-16,1	
17	16,11,2,12,1,J2,2021-06-17,1	
18	17,6,1,5,2,J2,2021-06-17,1	
19	18,9,2,10,0,J2,2021-06-17,1	
20	19,17,1,19,0,J2,2021-06-18,1	
21	20,14,1,15,1,J2,2021-06-18,1	
22	21,13,0,16,0,J2,2021-06-18,1	
23	22,24,1,21,1,J2,2021-06-19,1	
24	23,23,2,22,4,J2,2021-06-19,1	
25	24,18,1,20,1,J2,2021-06-19,1	
26	25,1,1,2,0,J3,2021-06-20,1	
27	26,3,3,4,1,J3,2021-06-20,1	
28	27,11,0,10,1,J3,2021-06-21,1	
29	28,12,0,9,3,J3,2021-06-21,1	
30	29,8,1,6,4,J3,2021-06-21,1	
31	30,7,0,5,2,J3,2021-06-21,1	
32	31,14,3,16,1,J3,2021-06-22,1	
33	32,15,0,13,1,J3,2021-06-22,1	
34	33,17,3,20,2,J3,2021-06-23,1	

35	34,19,0,18,5,J3,2021-06-23,1
36	35,23,2,21,2,J3,2021-06-23,1
37	36,22,2,24,2,J3,2021-06-23,1
38	37,2,0,6,4,OF,2021-06-26,1
39	38,1,2,10,1,OF,2021-06-26,1
40	39,9,0,15,2,OF,2021-06-27,1
41	40,5,1,23,0,OF,2021-06-27,1
42	41,14,3,18,5,OF,2021-06-28,1
43	42,21,4,3,5,OF,2021-06-28,1
44	43,13,2,22,0,OF,2021-06-29,1
45	44,17,1,11,2,OF,2021-06-29,1
46	45,3,1,18,3,CF,2021-07-02,1
47	46,5,1,1,2,CF,2021-07-02,1
48	47,15,1,6,2,CF,2021-07-03,1
49	48,11,0,13,4,CF,2021-07-03,1
50	49,1,4,18,2,SF,2021-07-06,1
51	50,13,2,6,1,SF,2021-07-07,1
52	51,1,0,13,0,F,2021-07-11,0

Punto 2: Almacenamiento

El siguiente programa de Python (**servidor\extraccion.py**), lee los archivos csv de los partidos y equipos, les da formato, limpia la base de datos para que no haya datos repetidos y finalmente inserta la información del dataset en la base de datos.

```
extraccion.py X
servidor > extraccion.py
1  # Proyecto Final AGVD (MACAVE STADISTIC)
2  # Equipo 5
3  # Castro Cazares Carlos Jahir (Responsable)
4  # Martinez Resendiz Giovanni Hasid
5  # Velazquez Aguilar Jaime Isai
6
7  # Almacenamiento de datos a partir del dataset
8
9  from conexionDB import cleanTable, insertEquipos, insertPartidos
10
11 #Extraccion y almacenamiento de los equipos
12 def equipos():
13     f = open("dataset/equipos.csv", "r")
14     cont=0
15     for linea in f:
16         if(cont>0):
17             arr = linea.split(",")
18             id = arr[0]
19             equipo = arr[1]
20             grupo = arr[2]
21             logo = "images"+(arr[3])[0:-1]
22
23             #Insertar datos a la base de datos
24             insertEquipos((id, equipo, grupo, logo))
25         cont+=1
26     f.close()
27
```



```

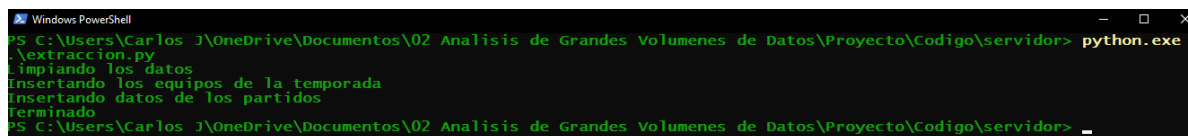
28  #Extraccion y almacenamiento de Los partidos
29  def partidos():
30      f = open("dataset/partidos.csv", "r")
31      cont=0
32      for linea in f:
33          if(cont>0):
34              arr = linea.split(",")
35              id = arr[0]
36              equipo1 = arr[1]
37              goles1 = arr[2]
38              equipo2 = arr[3]
39              goles2 = arr[4]
40              tipo = arr[5]
41              dia = arr[6]
42              jugado = arr[7]
43
44              #Insertar datos a la base de datos
45              insertPartidos((id, equipo1, goles1, equipo2, goles2, tipo, dia, jugado))
46          cont+=1
47      f.close()
48
49

```

```

50  if __name__=="__main__":
51      #Clean
52      print("Limpiando los datos")
53      cleanTable()
54
55      #Almacenamiento de Los equipos
56      print("Insertando los equipos de la temporada")
57      equipos()
58
59      #Almacenamiento de Los partidos
60      print("Insertando datos de los partidos")
61      partidos()
62
63      print("Terminado")
64

```



```

Windows PowerShell
PS C:\Users\Carlos J\OneDrive\Documentos\02 Analisis de Grandes Volúmenes de Datos\Proyecto\Codigo\servidor> python.exe
Limpiando los datos
Insertando los equipos de la temporada
Insertando datos de los partidos
Terminado
PS C:\Users\Carlos J\OneDrive\Documentos\02 Analisis de Grandes Volúmenes de Datos\Proyecto\Codigo\servidor>

```

Las consultas y conexión con la base de datos está en el archivo **servidor\conexionDB.py** aquí se configura la conexión con la base de datos.

```
1  # Conexiones con la base de datos para CRUD
2
3  import mysql.connector
4
5  conexion = {
6      'host': "localhost",
7      'usuario': "root",
8      'password': "",
9      'basedatos': "macave",
10     'puerto': 3307
11 }
```

```
13  #Limpieza database
14  def cleanTable():
15      mydb = mysql.connector.connect(
16          host=conexion['host'],
17          user=conexion['usuario'],
18          password=conexion['password'],
19          database=conexion['basedatos'],
20          port=conexion['puerto']
21      )
22      cursor = mydb.cursor()
23
24      sql = "DELETE FROM partidos WHERE id > 0"
25      cursor.execute(sql)
26      mydb.commit()
27
28      sql = "DELETE FROM equipos WHERE id > 0"
29      cursor.execute(sql)
30      mydb.commit()
31
32      cursor.close()
33
```

```
34  # Insertar equipos en la base de datos
35  def insertEquipos(input):
36      mydb = mysql.connector.connect(
37          host=conexion['host'],
38          user=conexion['usuario'],
39          password=conexion['password'],
40          database=conexion['basedatos'],
41          port=conexion['puerto']
42      )
43      cursor = mydb.cursor()
44      sql = "INSERT INTO equipos (id, nombre, grupo, logo) VALUES (%s, %s, %s, %s)"
45      cursor.execute(sql, input)
46      mydb.commit()
47      cursor.close()
48
```

```

49 # Insertar partidos
50 def insertPartidos(input):
51     mydb = mysql.connector.connect(
52         host=conexion['host'],
53         user=conexion['usuario'],
54         password=conexion['password'],
55         database=conexion['basedatos'],
56         port=conexion['puerto']
57     )
58     cursor = mydb.cursor()
59     sql = "INSERT INTO partidos (id, id_equipo_1, goles_1, id_equipo_2, goles_2, tipo,
60     cursor.execute(sql, input)
61     mydb.commit()
62     cursor.close()
63

```

Los datos se almacenarán en una base de datos relacional de mariaDB. La cual se crea con el siguiente script de SQL.

```

script_db_macave.sql X
db > script_db_macave.sql
1  /*
2  SQLyog Community v13.1.7 (64 bit)
3  MySQL - 10.5.4-MariaDB : Database - macave
4  ****
5  */
6  DROP DATABASE IF EXISTS `macave`;
7
8  /*!40101 SET NAMES utf8 */;
9
10 /*!40101 SET SQL_MODE='';
11
12 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
13 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
14 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
15 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
16 CREATE DATABASE /*!32312 IF NOT EXISTS*/ `macave` /*!40100 DEFAULT CHARACTER SET latin1 */;
17
18 USE `macave`;
19
20 /*Table structure for table `equipos` */
21
22 DROP TABLE IF EXISTS `equipos`;
23
24 CREATE TABLE `equipos` (
25   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
26   `nombre` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
27   `grupo` varchar(1) COLLATE utf8mb4_unicode_ci NOT NULL,
28   `logo` text COLLATE utf8mb4_unicode_ci NOT NULL,
29   PRIMARY KEY (`id`)
30 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
31

```

```

32  /*Table structure for table `partidos` */
33
34  DROP TABLE IF EXISTS `partidos`;
35
36  CREATE TABLE `partidos` (
37    `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
38    `id_equipo_1` bigint(20) unsigned NOT NULL,
39    `goles_1` smallint(6) DEFAULT NULL,
40    `id_equipo_2` bigint(20) unsigned NOT NULL,
41    `goles_2` smallint(6) DEFAULT NULL,
42    `tipo` varchar(2) COLLATE utf8mb4_unicode_ci NOT NULL,
43    `dia` date NOT NULL,
44    `jugado` tinyint(1) NOT NULL DEFAULT 0,
45    PRIMARY KEY (`id`),
46    KEY `partidos_id_equipo_1_foreign` (`id_equipo_1`),
47    KEY `partidos_id_equipo_2_foreign` (`id_equipo_2`),
48    CONSTRAINT `partidos_id_equipo_1_foreign` FOREIGN KEY (`id_equipo_1`) REFERENCES `equipos` (`id`) ON DELETE CASCADE,
49    CONSTRAINT `partidos_id_equipo_2_foreign` FOREIGN KEY (`id_equipo_2`) REFERENCES `equipos` (`id`) ON DELETE CASCADE
50  ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
51
52  /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
53  /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
54  /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
55  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
56

```

Punto 3: Algoritmo

El algoritmo que resuelve el problema de la predicción de marcadores de próximos partidos de futbol de la Eurocopa por probabilidad corresponde a los temas de **estadísticos y probabilísticos**; Por esto algoritmo se basa en los siguientes pasos que se documentaron en el código de Python (**servidor\main.py**), que se maneja en el método **consulta_info**:

1. Obtener la información de todos los partidos jugados en la copa (marcadores)

```
39 @route('/datos')
40 @enable_cors
41 def consulta_info():
42     response.headers['Content-type'] = 'application/json'
43     # Get todos los partidos
44     partidos = getPartidosJugados()
45
64 # Partidos Jugados
65 def getPartidosJugados():
66     mydb = mysql.connector.connect(
67         host=conexion['host'],
68         user=conexion['usuario'],
69         password=conexion['password'],
70         database=conexion['basedatos'],
71         port=conexion['puerto']
72     )
73     cursor = mydb.cursor()
74     cursor.execute("SELECT * FROM partidos WHERE jugado = 1")
75     return cursor.fetchall()
76
```

2. Obtener la información de los partidos próximos (equipos)

```
46 # Get los partidos que no han sido jugados
47 proximos = getPartidosProximos()
48
77 # Partidos Proximos
78 def getPartidosProximos():
79     mydb = mysql.connector.connect(
80         host=conexion['host'],
81         user=conexion['usuario'],
82         password=conexion['password'],
83         database=conexion['basedatos'],
84         port=conexion['puerto']
85     )
86     cursor = mydb.cursor()
87     cursor.execute("SELECT * FROM partidos WHERE jugado = 0")
88     return cursor.fetchall()
89
```

3. Por cada uno de los partidos próximos se deberán de calcular los siguientes datos estadísticos:

a. Media de goles en la copa

```
49 # Pronosticar (Estadística y Probabilidad)
50 data = {}
51 for proximo in proximos:
52     # Jugadores
53     equipo1 = {
54         'id': proximo[1], # Id del equipo
55         'ataque': 0, # Media de goles
56         'defensa': 0, # Media de encajes
57         'exito': 0, # Tasa promedio de Exito
58         'probabilidades': [0, 0, 0, 0, 0, 0] # Probabilidad de anotar de 0 - 5 Goles
59     }
60     equipo2 = {
61         'id': proximo[3],
62         'ataque': 0,
63         'defensa': 0,
64         'exito': 0,
65         'probabilidades': [0, 0, 0, 0, 0, 0]
66     }
67
68     # Calcular estadísticas de cada equipo (Medias)
69     totalGoles = 0
70
71     for partido in partidos:
72         # Total de goles de la copa
73         totalGoles += partido[2] + partido[4]
74
75         # Total de goles anotados y encajados por equipo 1
76         if partido[1] == equipo1['id']:
77             equipo1['ataque'] += partido[2]
78             equipo1['defensa'] += partido[4]
79         if partido[3] == equipo1['id']:
80             equipo1['ataque'] += partido[4]
81             equipo1['defensa'] += partido[2]
82
83         # Total de goles anotados y encajados por equipo 2
84         if partido[1] == equipo2['id']:
85             equipo2['ataque'] += partido[2]
86             equipo2['defensa'] += partido[4]
87         if partido[3] == equipo2['id']:
88             equipo2['ataque'] += partido[4]
89             equipo2['defensa'] += partido[2]
90
91     # Calculo de medias de anotaciones(ataque) y encajados(defensa)
92     totalPartidos = len(partidos)
93     mediaGoles = totalGoles / totalPartidos
```

b. Media de goles por cada equipo (Ataque)

```
94     equipo1['ataque'] /= totalPartidos
95     equipo1['defensa'] /= totalPartidos
96
```

c. Media de anotaciones por cada equipo (Defensa)

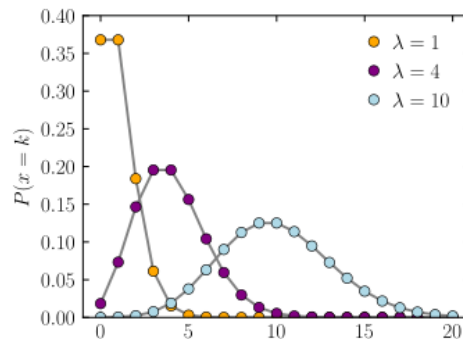
```
97     equipo2['ataque'] /= totalPartidos
98     equipo2['defensa'] /= totalPartidos
99
```

d. Tasa promedio de éxito por equipo (ataque propio x defensa del oponente x media de goles en la copa).

```
100     # Tasa promedio de éxito
101     equipo1['exito'] = equipo1['ataque'] * equipo2['defensa'] * mediaGoles
102     equipo2['exito'] = equipo2['ataque'] * equipo1['defensa'] * mediaGoles
103
```

4. Posteriormente por cada equipo teniendo su tasa promedio de éxito y usando el método de probabilidad de la distribución poisson se pude calcular la probabilidad de anotar de 0 a 5 goles por equipo.

$$P[X = k] = \frac{e^{-\lambda} \lambda^k}{k!}$$



K = Numero goles y λ tas de éxito

```
104     # Probabilidad de anotar de 0-5 Goles con Distribucion Poisson
105     for k in range(len(equipo1['probabilidades'])):
106         equipo1['probabilidades'][k] = probabilidadPoisson(k, equipo1['exito'])
107         equipo2['probabilidades'][k] = probabilidadPoisson(k, equipo2['exito'])
108
```

```
20
21 def probabilidadPoisson(k, lambd):
22     return (lambd ** k * np.exp(-lambd)) / np.math.factorial(k)
23
```

5. Finalmente, como cada probabilidad de anotaciones por equipo es independiente podemos multiplicar las probabilidades de cero goles del primer equipo con la de cero con el segundo equipo; así con cada una de las probabilidades para obtener la probabilidad de que el marcador sea 0-0, 0-1, 0-2, ..., 5-5.

```
109 # Convinacion de probabilidades
110 probabilidades = []
111 for probabilidadE1 in equipo1['probabilidades']:
112     probabilidadN = []
113     for probabilidadE2 in equipo2['probabilidades']:
114         probabilidadN.append(round((probabilidadE1 * probabilidadE2)*100, 2))
115     probabilidades.append(probabilidadN)
116
```

6. Finalmente se obtienen los nombres de los equipos, se formatean los datos a 2 decimales y el formato de los datos para enviarlos al cliente.

```
117 # Formato de datos para su envio
118 equipo1['ataque'] = round(equipo1['ataque'], 2)
119 equipo1['defensa'] = round(equipo1['defensa'], 2)
120 equipo1['exito'] = round(equipo1['exito'], 2)
121 equipo1['probabilidades'] = [round((p*100), 2) for p in equipo1['probabilidades']]
122
123 equipo2['ataque'] = round(equipo2['ataque'], 2)
124 equipo2['defensa'] = round(equipo2['defensa'], 2)
125 equipo2['exito'] = round(equipo2['exito'], 2)
126 equipo2['probabilidades'] = [round((p * 100), 2) for p in equipo2['probabilidades']]
127
128 infoEquipo1 = getInformacionEquipo(equipo1['id'])[0]
129 infoEquipo2 = getInformacionEquipo(equipo2['id'])[0]
130
```

```
90 # Get Informacion Equipo
91 def getInformacionEquipo(id):
92     mydb = mysql.connector.connect(
93         host=conexion['host'],
94         user=conexion['usuario'],
95         password=conexion['password'],
96         database=conexion['basedatos'],
97         port=conexion['puerto']
98     )
99     cursor = mydb.cursor()
100     cursor.execute(f"SELECT * FROM equipos WHERE id = {id}")
101     return cursor.fetchall()
```



```
131     tipo = ''
132     if (proximo[5] == 'J1'):
133         tipo = 'Jornada 1'
134     elif (proximo[5] == 'J2'):
135         tipo = 'Jornada 2'
136     elif (proximo[5] == 'J3'):
137         tipo = 'Jornada 3'
138     elif (proximo[5] == 'OF'):
139         tipo = 'Octavos de Final'
140     elif (proximo[5] == 'CF'):
141         tipo = 'Cuartos de Final'
142     elif (proximo[5] == 'SF'):
143         tipo = 'Semifinal'
144     elif (proximo[5] == 'F'):
145         tipo = 'Final'
146
```

```
147     partido = {
148         'id': proximo[0],
149         'equipo1': infoEquipo1[1],
150         'logo1': infoEquipo1[3],
151         'equipo2': infoEquipo2[1],
152         'logo2': infoEquipo2[3],
153         'tipo': tipo,
154         'fecha': proximo[6].strftime('%d/%m/%Y')
155     }
156
157     data={
158         'partido': partido,
159         'marcadores': probabilidades,
160         'equipo1': equipo1,
161         'equipo2': equipo2
162     }
163
164     print(data)
165     return data
```

Para hacer la conexión entre el servidor que obtiene, calcula y formatea los datos y el cliente una pagina web que muestra la información, se tubo que implementar una librería para crear un web services, que es consumido por el cliente su configuración es la siguiente.

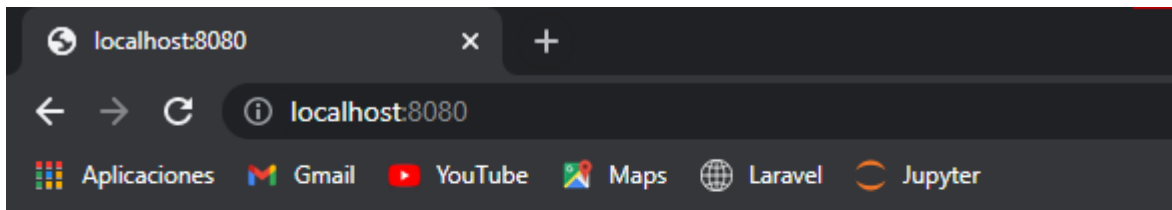
```
1  # Proyecto Final AGVD (MACAVE STADISTIC)
2  # Equipo 5
3  # Castro Cazares Carlos Jahir (Responsable)
4  # Martinez Resendiz Giovanni Hasid
5  # Velazquez Aguilar Jaime Isai
6
7  # Algoritmo para el procesamiento de datos y resultados estadísticos y probabilidades
8
9  # Librerías
10 import numpy as np
11 import bottle
12 from bottle import route, run, template
13 from bottle import response
14
15 # Conexion con base de datos
16 from conexionDB import getPartidosJugados, getPartidosProximos, getInformacionEquipo
17
18 # Variable global
19 app = bottle.app()
20
21 def probabilidadPoisson(k, lambd):
22     return (lambd ** k * np.exp(-lambd)) / np.math.factorial(k)
23
24 # CORS
25 def enable_cors(fn):
26     def _enable_cors(*args, **kwargs):
27         response.headers['Access-Control-Allow-Origin'] = '*'
28         response.headers['Access-Control-Allow-Methods'] = 'GET, POST, PUT, OPTIONS'
29         response.headers['Access-Control-Allow-Headers'] = 'Origin, Accept, Content-Type, X-Requested-With, X-CSRF-Token'
30
31         if bottle.request.method != 'OPTIONS':
32             return fn(*args, **kwargs)
33     return _enable_cors
34
```

```
35 @app.route('/', method=['GET'])
36 def index():
37     return '<h1>Bienvenido a MACAVE Stadistics</h1> <br>http://127.0.0.1:8080/datos => Este es la url con los da
38
```

```
39 @route('/datos')
40 @enable_cors
41 > def consulta_info(): ...
166
167 if __name__=="__main__":
168     app.run(port=8080)
169
```

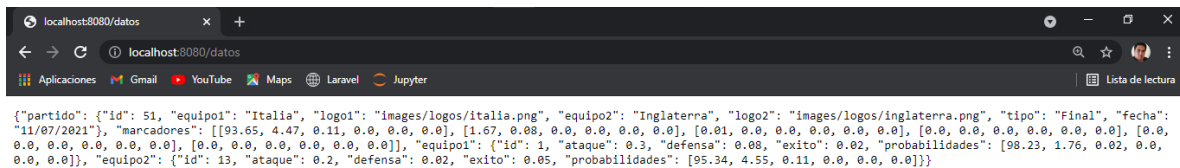
Ejecución:

```
Windows PowerShell
PS C:\Users\Carlos J\OneDrive\Documentos\02 Analisis de Grandes Volúmenes de Datos\Proyecto\Codigo\servidor> python.exe .\main.py
Bottle v0.12.19 server starting up (using WSGIRefServer())...
Listening on http://127.0.0.1:8080/
Hit Ctrl-C to quit.
```



Bienvenido a MACAVE Stadistics

`http://127.0.0.1:8080/datos` => Este es la url con los datos de prediccion

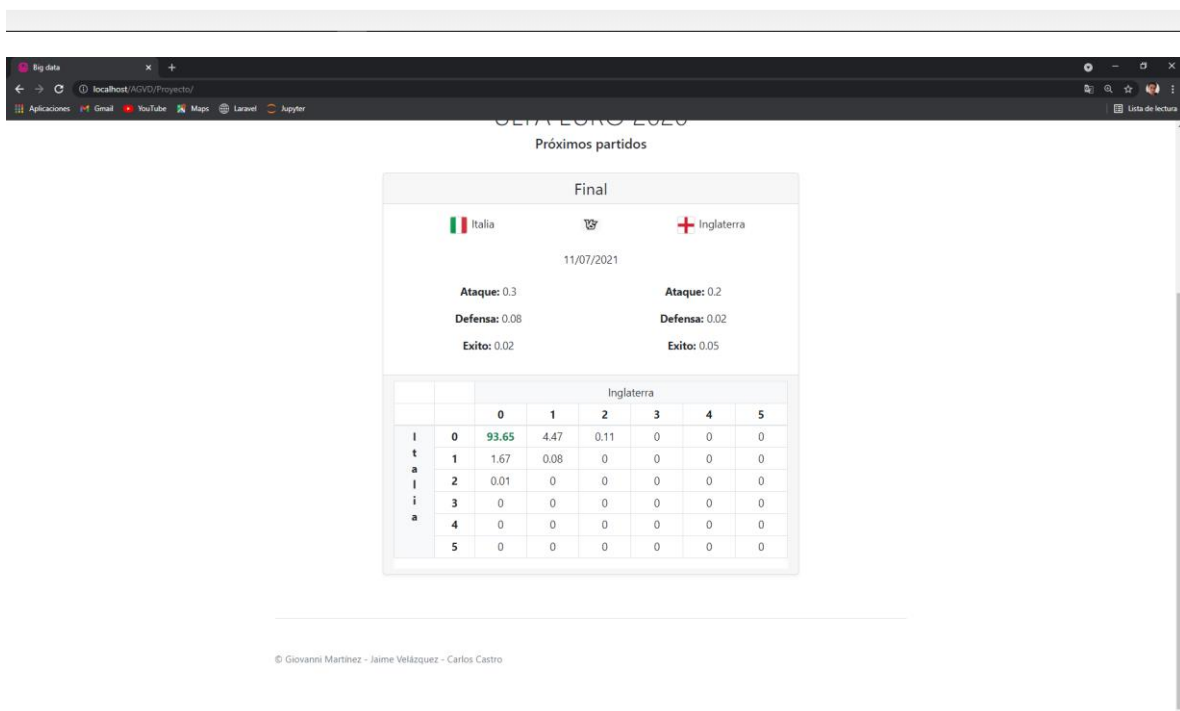
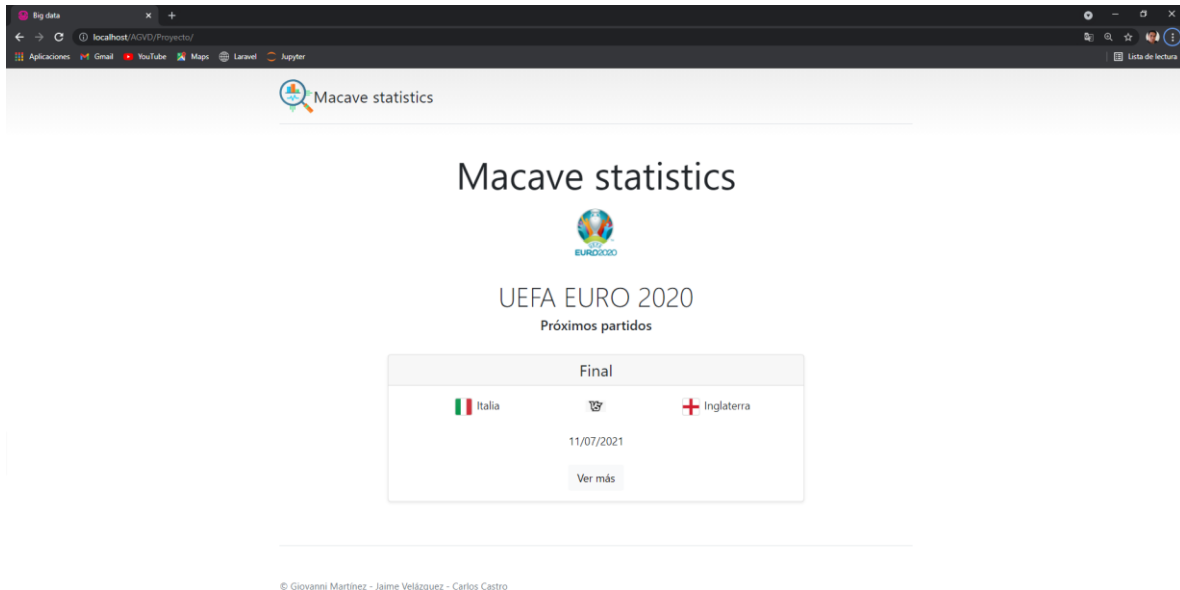


Las librerías que se usaron para los programas en Python son las siguientes:

- **MySQL-python 1.2.5**
pip install MySQL-python
- **numpy 1.21.0**
pip install numpy
- **bottle 0.12.19**
pip install bottle

Punto 4: Visualización

Los resultados estadísticos de cada equipo y las probabilidades de los marcadores por partido se muestran en una pagina web, donde se muestran los partidos pendientes y al dar clic en ellos se muestran las estadísticas de cada equipo y las probabilidades de los marcadores. En nuestro caso por los tiempos solo queda el partido final de la copa, pero esto no exenta el poder usarlo con otras copas o ligas.



Código

```
index.html ×
cliente > index.html > ...
1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <meta name="description" content="">
8      <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
9      <meta name="generator" content="Hugo 0.84.0">
10     <title>Big data</title>
11
12     <link rel="canonical" href="public/pricing.css">
13
14     <!-- Bootstrap core CSS -->
15     <link href="public/bootstrap.min.css" rel="stylesheet">
16
17     <style>
18         .bd-placeholder-img {
19             font-size: 1.125rem;
20             text-anchor: middle;
21             -webkit-user-select: none;
22             -moz-user-select: none;
23             user-select: none;
24         }
25
26         @media (min-width: 768px) {
27             .bd-placeholder-img-lg {
28                 font-size: 3.5rem;
29             }
30         }
31     </style>
32
```

```

33     <!-- Custom styles for this template -->
34     <link href="public/pricing.css" rel="stylesheet">
35 </head>
36
37 <body>
38
39     <div class="container py-3">
40         <header>
41             <div class="d-flex flex-column flex-md-row align-items-center pb-3 mb-4 border-bottom">
42                 <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
43                     
44                     <span class="fs-4">Macave statistics</span>
45                 </a>
46             </div>
47
48             <div class="pricing-header p-3 pb-md-4 mx-auto text-center">
49                 <h1 class="display-4 fw-normal">Macave statistics</h1>
50                 <p class="fs-5 text-muted">La mejor opción para tus apuestas deportivas</p>
51                 
52             </div>
53         </header>
54
55         <main>
56             <h2 class="col text-center display-6">UEFA EURO 2020</h2>
57             <h5 class="col text-center">Próximos partidos</h5>
58             <br>
59             <div id="partidos" class="row row-cols-1 row-cols-md-3 mb-3 text-center" id="partidos">
60             </div>
61         </main>
62

```

```

63     <footer class="pt-4 my-md-5 pt-md-5 border-top">
64         <div class="row">
65             <div class="col-12 col-md">
66                 <small class="d-block mb-3 text-muted">&copy; Giovanni Martínez - Jaime Velázquez - Carlos Castro</small>
67             </div>
68         </div>
69     </footer>
70 </div>
71 <script src="https://code.jquery.com/jquery-3.6.0.min.js"
72     integrity="sha256-/xUj+30JU5yExlq66SYGSHk7tPXikyn57ogEvDej/m4=" crossorigin="anonymous"></script>
73 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js"
74     integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
75     crossorigin="anonymous"></script>
76 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
77     integrity="sha384-JZR65Pej4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
78     crossorigin="anonymous"></script>
79 <script src="public/index.js" type="text/javascript"></script>
80
81 </body>
82
83 </html>

```

index.js

cliente > public > index.js

```
1 function info(id) {
2   $('#footer-' + id).removeClass("d-none");
3   $('#col-' + id+'-1').removeClass("d-none");
4   $('#col-' + id+'-2').removeClass("d-none");
5
6   $('#btn-'+id).addClass("d-none");
7 }
8
```

```
9 $(document).ready(function () {
10   $.ajax({
11     type: "get",
12     url: "http://127.0.0.1:8080/datos",
13     dataType: "json",
14     success: function (data) {
15       console.log(data);
16       prob=0;
17       data.marcadores.forEach(task => {
18         for(var i=0; i<task.length; i++){
19           if(task[i]>prob){
20             prob=task[i];
21           }
22         }
23       });
24       var cont=0;
25       var word="";
26       for(var i=0; i<data.partido.equipo1.length; i++){
27         word += data.partido.equipo1[i]+'<br>';
28       }
29       var tbd=`<tr><th rowspan="7" class="bg-light">`+word+`</th></tr>`;
30       data.marcadores.forEach(task => {
31         tbd+=`<tr><th>`+cont+`</th>`;
32         for(var i=0; i<task.length; i++){
33           if(task[i]==prob){
34             tbd+=`<th class="text-success">`+task[i]+`</th>`;
35           }
36           else{
37             tbd+=`<td>`+task[i]+`</td>`;
38           }
39         }

```

```

40         cont++;
41     });
42     tbd+="|";
43
44
45     cad="
|  |

```



```
102         <tbody>
103     ✓         `+tbd+`
104         </tbody>
105     </table>
106     </div>
107 </div>
108 </div>
109 </div>
110 </div>`;
111     $('#partidos').append(cad)
112 }
113 });
114 });
```

Punto 5: Complejidad e Impacto

Una de las mayores complejidades del proyecto fue buscar la fórmula adecuada que nos pudiera determinar la probabilidad de los marcadores de los partidos. Una vez encontrada, fue un poco difícil acoplarla a nuestra información, pero después de ver varios documentos, tutoriales y ejemplos, supimos cómo obtener los resultados esperados. Otra de las complejidades fue crear un micro web-framework para poder mandar la información desde python, el cual fue resuelto con la librería de **bottle**, el cual está diseñado para ser rápido y ligero, este se pudo resolver debido a la documentación que existe.

El impacto esperado del proyecto es que pueda ser utilizado para predecir los posibles marcadores de los partidos, ayudando a las personas a realizar inversiones con una base sólida.

Conclusiones y resultados del proyecto.

El análisis de grandes volúmenes de datos puede impactar en nosotros de manera positiva y ayudarnos a identificar nuevas oportunidades. Ahora es importante conocer los datos que tenemos almacenados para poder manipularlos y ver que relaciones tienen unos a otros o inclusive predecir un posible resultado en base a algún patrón que se vaya teniendo.

Durante las practicas que tuvimos durante el semestre vimos cómo es que con mucha información podemos determinar una conclusión en base a su análisis estadístico y obtener conclusiones en base a ello, y que inclusive si tuviéramos más la estadística sería más precisa. Todos estos temas nos sirvieron para desarrollar nuestro tema de MACAVE Statistics de manera exitosa y creemos que no es solamente un proyecto si no que este se podría llegar a utilizar o inclusive a comercializar.

Los resultados que alcanzamos en nuestro proyecto, fueron los esperados ya que estuvimos haciendo pruebas con las semifinales de la Eurocopa en la que la mayoría acertó y esperemos que tengamos el mismo éxito con la final de la próxima semana. Igualmente, este sistema se puede usar para otras ligas solo alimentando la base de datos.

