

Universidad Internacional de La Rioja (UNIR)

**Escuela Superior de Ingeniería y
Tecnología**

**Máster en Análisis y Visualización de Datos
Masivos**

Visualización interactiva en resultados de ajedrez

Trabajo Fin de Máster

presentado por: Martín Amores, Cristian

Directora: Fuentes Lorenzo, Damaris

Ciudad: Salamanca

Fecha: 18 de septiembre de 2018

ÍNDICE

1. INTRODUCCIÓN	7
1.1. MOTIVACIÓN	7
1.2. OBJETIVO PRINCIPAL	8
1.3. ESTRUCTURA DEL DOCUMENTO	9
2. CONTEXTO Y ESTADO DEL ARTE	10
2.1. CONTEXTO	10
2.2. ESTADO DEL ARTE	10
3. OBJETIVOS Y METODOLOGÍA DE TRABAJO	12
3.1. OBJETIVO GENERAL	12
3.2. OBJETIVOS ESPECÍFICOS	12
3.3. METODOLOGÍA DE TRABAJO	12
4. IDENTIFICACIÓN, TECNOLOGÍAS Y PROCESAMIENTO DE DATOS	14
4.1. IDENTIFICACIÓN DE LAS FUENTES DE DATOS	15
4.2. TECNOLOGÍAS UTILIZADAS	17
SCRIPT CAISSATOOLS.JAR	17
MONGODB	18
NoSQLBOOSTER FOR MONGODB	20
TABLEAU SOFTWARE	21
MICROSTRATEGY	24
4.3. CAPTURA DE LOS DATOS	25
4.4. ESTUDIO Y PROCESAMIENTO DE LOS DATOS	27
CONVERSIÓN DE PGN A JSON	27
ARRANQUE SERVICIO MONGOD	29
IMPORTACIÓN DE LAS PARTIDAS A MONGODB	30
PROCESAMIENTO DE LOS DATOS EN MONGODB	32
EXTRACCIÓN DE LA INFORMACIÓN A UN FICHERO JSON	37
5. DESARROLLO DE LAS VISUALIZACIONES	39
5.1. IMPORTACIÓN DE LOS DATOS	39
5.2. CREACIÓN DE LAS VISUALIZACIONES INDIVIDUALES	41
5.3. CREACIÓN DEL CUADRO DE MANDO	52
5.4. CREACIÓN DEL GRAFO	55
5.5. EJEMPLO DE USO	58
6. CONCLUSIONES Y TRABAJO FUTURO	60
7. BIBLIOGRAFÍA	61

ÍNDICE DE FIGURAS

FIGURA 1. ESQUEMA DEL DESARROLLO	14
FIGURA 2. EJEMPLO DE LA ESTRUCTURA DE UNA PARTIDA DE AJEDREZ EN FORMATO PGN	16
FIGURA 3. LOGO DE LA PÁGINA WEB DE MARCO DE BOOIJ.....	17
FIGURA 4. DESCARGA DE CAISSATOOLS.....	17
FIGURA 5. EXPLICACIÓN DE LAS OPCIONES DE CAISSATOOLS	18
FIGURA 6. LOGO DE MONGODB	19
FIGURA 7. SITIO WEB DE DESCARGA DE MONGODB.....	19
FIGURA 8. LOGO DE NoSQLBOOSTER	20
FIGURA 9. SITIO WEB DE DESCARGA DE NoSQLBOOSTER FOR MONGODB	20
FIGURA 10. INTERFAZ DE NoSQLBOOSTER	21
FIGURA 11. LOGO DE TABLEAU SOFTWARE.....	21
FIGURA 12. SITIO WEB DESCARGA DE TABLEAU DESKTOP.....	22
FIGURA 13. TUTORIALES DE TABLEAU OFICIALES.....	22
FIGURA 14. EJEMPLO DE LOS TEMAS PARA LA CREACIÓN DE VISUALIZACIONES CON TABLEAU	23
FIGURA 15. INTERFAZ DE TABLEAU DESKTOP.....	24
FIGURA 16. LOGO DE MICROSTRATEGY.....	24
FIGURA 17. INTERFAZ DE CREACIÓN DE DOSSIER EN MICROSTRATEGY	25
FIGURA 18. PÁGINA DE SELECCIÓN DE PARTIDAS POR JUGADOR.....	25
FIGURA 19. EJEMPLO DE DESCARGA DE LAS PARTIDAS EN PGN MENTOR	27
FIGURA 20. CONVERSIÓN DE PGN A JSON CON CAISSATOOLS.JAR	28
FIGURA 21. COMANDO DE ARRANQUE PARA MONGOD.....	29
FIGURA 22. SERVIDOR MONGOD CORRECTAMENTE ARRANCADO	30
FIGURA 23. CONEXIÓN A LA BASE DE DATOS DESDE NoSQLBOOSTER FOR MONGODB	30
FIGURA 24. IMPORTACIÓN DEL ARCHIVO JSON.....	31
FIGURA 25. DOCUMENTOS IMPORTADOS EN MONGODB	31
FIGURA 26. SCRIPT EN AGGREGATION FRAMEWORK DE MONGODB	33
FIGURA 27. NÚMERO DE DOCUMENTOS DUPLICADOS ELIMINADOS	33
FIGURA 28. ELIMINACIÓN DE CAMPOS.....	35
FIGURA 29. CAMBIO DE NOMBRE DE _MOVES A MOVES	35
FIGURA 30. ELIMINACIÓN DE LOS CAMPOS IRRELEVANTES PARA EL ESTUDIO.....	35
FIGURA 31. CAMBIO DE VALORES EN EL CAMPO RESULT.....	36
FIGURA 32. ACTUALIZACIÓN DE LOS NOMBRES DE LOS JUGADORES.....	37
FIGURA 33. EXTRACCIÓN DE LA INFORMACIÓN A UN FICHERO JSON	38
FIGURA 34. EJEMPLO DEL FICHERO FINAL.JSON	38
FIGURA 35. MENÚ DE CONEXIÓN DE TABLEAU DESKTOP.....	39
FIGURA 36. ESQUEMA DE LOS CAMPOS A IMPORTAR EN TABLEAU.....	40
FIGURA 37. TABLA DE DATOS IMPORTADOS EN TABLEAU.....	41
FIGURA 38. GRÁFICA GENERAL, PRIMER PASO	42

FIGURA 39.GRÁFICA GENERAL, SEGUNDO PASO.....	42
FIGURA 40.GRÁFICA GENERAL, TERCER PASO.....	43
FIGURA 41.GRÁFICA GENERAL SIMPLE.....	43
FIGURA 42.ORDENADO DE COLUMNAS.....	44
FIGURA 43.GRÁFICA GENERAL COMPLETA	45
FIGURA 44.MODIFICACIÓN DE TOOLTIP.....	46
FIGURA 45.TOOLTIP MODIFICADO	46
FIGURA 46.ORDENACIÓN DE APERTURAS.....	47
FIGURA 47.COLUMNAS Y FILAS PARA CONTABILIZAR APERTURAS.....	47
FIGURA 48.GRÁFICO DE LARGA COLA PARA LAS APERTURAS	48
FIGURA 49.DESPLEGABLE MARKS.....	48
FIGURA 50.GRÁFICO DE APERTURAS MÁS JUGADAS.....	49
FIGURA 51.GRÁFICO DE BURBUJAS EMPAQUETADAS, PRIMER PASO.....	50
FIGURA 52.GRADIENTE DE COLOR SELECCIONADO	51
FIGURA 53.GRÁFICO DE BURBUJAS EMPAQUETADAS	51
FIGURA 54.CUADRO DE MANDO.....	53
FIGURA 55.VENTANA DE ACTIONS.....	54
FIGURA 56.HIGHLIGHT ACTION	54
FIGURA 57.EJEMPLO DE HIGHLIGHT CON APERTURA B90	55
FIGURA 58.EDITOR PARA NETWORK.....	56
FIGURA 59.GRAFO DE LOS JUGADORES	57
FIGURA 60.EJEMPLO DE USO, BURBUJAS	58
FIGURA 61.INFORMACIÓN DE LOS MOVIMIENTOS REALIZADOS.....	59

Resumen

El trabajo consiste en la creación de un cuadro de mando interactivo con la herramienta de visualización Tableau para el análisis de los resultados obtenidos en ajedrez basándose en las aperturas utilizadas de doce Grandes Maestros en un gran número de partidas. El principal problema que se encuentra al estudiar aperturas en ajedrez es la elección de éstas, ya que existen un gran número de ellas. La visualización está enfocada para que la utilice un jugador de ajedrez aficionado o de club para ayudar a la decisión de qué aperturas estudiar y aprender en base a las veces que se utilizan y el éxito de éstas.

Palabras Clave: Ajedrez, aperturas, análisis, visualización interactiva, Tableau

Abstract

The work involved consists of the creation of an interactive dashboard using the visualization tool called Tableau for the analysis of the results obtained in chess based on the openings used by twelve Grand Masters in a large number of games. The main problem when studying chess openings is which one should be chosen, since there are a large number of them. The visualization is aimed to be used by an amateur or club chess player to help them to decide which openings they have to study and learn based on how many times they have been used and how many times they have been successful.

Keywords: Chess, openings, analysis, interactive visualization, Tableau

1. Introducción

El estudio del ajedrez, para su comprensión y mejora como jugador competitivo o de club, siempre ha sido una tarea complicada de gestionar ya que existe una gran cantidad de información para prepararse desde que eres un jugador aficionado hasta que se llega a nivel de torneo profesional.

En este trabajo se pretende crear un cuadro de mando con la información de partidas del más alto nivel para un análisis visual e interactivo de los resultados obtenidos por los Grandes Maestros, con el uso de distintos tipos de aperturas y defensas para que así el usuario sea capaz de filtrar qué aperturas debería de prepararse con mayor dedicación de cara a sus futuras partidas. Se explicará detalladamente cómo se realizan las visualizaciones y la potencia que tienen para el usuario.

La apertura en ajedrez se conoce como los primeros movimientos realizados para el desarrollo de las piezas desde su posición inicial. A lo largo de los años se ha teorizado sobre dichos movimientos y se les ha ido dando nombres, por ejemplo: apertura española, defensa francesa, defensa india de rey, ataque indio de rey, etc.

Cuando la apertura se ve desde el punto de vista de las negras, se puede llamar defensa, ya que es este jugador el que va respondiendo a los movimientos de las blancas y por eso se conoce con este nombre. En definitiva, definir las como apertura o defensa es básicamente lo mismo, ya que como se ha explicado antes, son los movimientos de desarrollo. Una vez terminada la apertura se llega al medio juego y por último al final. Siendo estas, las tres fases del ajedrez: apertura, medio juego y final.

Para que todo esto sea posible se van a descargar una serie de partidas del más alto nivel de la página web de PGN Mentor (PGN Mentor, 2018). Una vez se tengan todas las partidas que se necesitan, se tendrá que estudiar el formato de los ficheros para su futuro procesamiento e inclusión en la base de datos que se utilizará para realizar las visualizaciones.

1.1. Motivación

Hoy en día, con el uso de la tecnología y la gran oferta que hay en los distintos medios de difusión como YouTube o Twitch, existe demasiado material sobre el ajedrez y se hace muy

complicada la selección del material a estudiar ya que se explica prácticamente de todo sin un plan de estudio determinado.

El tipo de vídeos que se pueden encontrar son explicaciones de los movimientos que hay que aprenderse de la teoría de una cierta apertura y el porqué de estos. Existen demasiadas aperturas como para aprenderlas todas al nivel de aficionado o de club, por eso se hace complicado el tema de elegir qué vídeo ver sin saber la efectividad que suele tener una u otra. Un ejemplo de vídeo es el realizado por Luís Fernández Siles en YouTube¹ para Chess.com (Chess.com, 2007).

Para poder lidiar con todo el contenido público que existe se han creado plataformas online en las que se puede, tanto jugar, como aprender teoría sobre el ajedrez de una manera más compacta y distribuida por niveles, como lo son, por ejemplo, Chess.com o Chess24 (Chess24, 2014).

El problema que surge con todo esto es saber elegir qué material estudiar de todo lo que se proporciona y por ello se realiza este trabajo.

1.2. Objetivo principal

El objetivo principal del trabajo es crear las visualizaciones que van a mostrar de una forma interactiva y visual los resultados que han obtenido con filtros personalizados por el usuario para obtener la información que sea relevante para él. De esta forma se podría solucionar de una manera sencilla, para un jugador aficionado o de club, en qué focalizar su aprendizaje.

Se decide utilizar una técnica visual porque es más intuitiva al uso y los resultados que se ven en ella son fácilmente comprensibles, comparando por ejemplo las alturas entre varias barras, se ve a simple vista cuál es más utilizado.

Es mucho más sencillo para los humanos la comparación horizontal de las cosas ya que enseguida distinguimos cuáles son más altas. Sería más complicado si pusiéramos simplemente el porcentaje de muchos datos. Con la visualización no se necesitaría apenas más información. También es más sencillo la comparación de tamaños a simple vista.

¹ https://www.youtube.com/watch?v=k5JVfKTsQ_k

1.3. Estructura del documento

En la estructura del documento aparecerá el contexto y el estado del arte, apartado en el que se explicará la situación actual del aprendizaje de ajedrez y las posibles soluciones que se han llevado a cabo hasta ahora con el uso de herramientas y tutoriales.

Se detallarán también los objetivos, tanto general como específicamente. Se continuará con el desarrollo de la contribución explicando detalladamente los pasos a seguir para realizar las visualizaciones.

Se empezará con la identificación de las fuentes de datos en el que se explica de dónde se capturan los datos, el formato que tienen y el formato que se necesitará para poder procesarlos.

Se expondrán las herramientas tecnológicas que se van a utilizar y cómo obtenerlas. En el siguiente paso se capturarán los datos y se explicará el proceso de la elección de las partidas, cómo y dónde conseguirlas.

En el estudio y procesamiento de los datos se estudiará el formato de las partidas descargadas y se hará la conversión a otro formato para su procesamiento. Se hablará del arranque de los servicios y de la importación a la base de datos, así como el procesado de los datos para la obtención de los campos necesarios y útiles para las visualizaciones. Se explicará paso a paso cómo se crean las visualizaciones a modo de tutorial.

Se realizará un ejemplo de uso del cuadro de mando obtenido en las visualizaciones para comprobar que responde a posibles preguntas que se puedan plantear y demostrar su utilidad.

Por último, se hablará de las conclusiones, se verá si se han cumplido los objetivos marcados y se planteará el posible trabajo futuro.

2. Contexto y estado del arte

En este apartado se hablará del contexto del aprendizaje de las aperturas en ajedrez y de los software y páginas web que más se utilizan para analizar las partidas y qué información nos muestran.

2.1. Contexto

El estudio de las aperturas en ajedrez consiste en la memorización y entendimiento de los primeros movimientos que realizan en una partida. Hoy en día existen muchísimos vídeos y libros explicando la mayoría de las aperturas con muchas de sus variantes, así como poniendo ejemplos de partidas completas en las se utilizó cierta apertura.

El problema que hay con este tema es que son demasiadas aperturas y variantes para aprendérselas todas. Los profesores y entrenadores de ajedrez de jugadores novatos o aficionados filtran las aperturas que tienen que enseñar por su dificultad para que se vayan aprendiendo las más comunes.

El ajedrez es un juego con muchísimas configuraciones posibles. Un paso muy importante para conseguir el éxito en una partida es empezarla bien, realizar una buena apertura con bastantes posibilidades para la victoria. Por este motivo, el trabajo se va a enfocar en los resultados obtenidos tras la utilización de cada apertura.

2.2. Estado del arte

Hoy en día las plataformas más utilizadas para el estudio del ajedrez son Chess.com, Chess24 y ChessBase (ChessBase, 2018). Tanto Chess.com como Chess24 son dos plataformas online que constan de un sistema de juego en línea para los jugadores registrados en sus respectivas páginas. Estas plataformas, además, disponen de aplicaciones para dispositivos móviles.

En estas dos plataformas existe una zona de acceso Premium para usuarios de pago en la que se puede acceder a contenido educativo. El contenido en ambas páginas consta de vídeos con explicaciones de aperturas, partidas actuales, partidas históricas, etc.

ChessBase es un software que cuenta con una enorme base de datos con prácticamente todas las partidas que se han documentado en la historia. También dispone de un tablero para analizar posiciones específicas. Consta de muchos filtros para navegar entre las partidas y con un software de inteligencia artificial para calcular el mejor movimiento en cada posición en base a todas las posibles respuestas.

Las tres plataformas tienen un árbol de aperturas. El árbol de aperturas se compone de los movimientos realizados uno a uno y calcula cuántas partidas existen, dependiendo del siguiente de movimiento. Es decir, la partida empieza con el primer movimiento del blanco y el negro tiene muchos movimientos posibles para responder. El árbol de aperturas mostrará cuántas partidas existen con cada respuesta posible del negro. Al seleccionar una respuesta, calculará de nuevo lo mismo, pero para el siguiente movimiento que ahora es del blanco.

El problema que se encuentra en las bases de datos y en los softwares de ajedrez es que no son visuales. La información estadística siempre se muestra mediante números, ya que todas las plataformas están más orientadas al juego en sí y a los movimientos calculados por la inteligencia artificial. Aunque también cuentan con un gran número de vídeos educativos, el usuario se encuentra en el problema de la elección.

Una experiencia visual e interactiva podría ser bastante productiva para un jugador de ajedrez aficionado ya que son fáciles de interpretar. Esta visualización podría guiar al jugador en su estudio del ajedrez dentro de las otras plataformas y utilizarla de manera complementaria.

3. Objetivos y metodología de trabajo

En este apartado se van a explicar tanto el objetivo general como todos los objetivos específicos necesarios para la realización del trabajo. También se explicará la metodología empleada para conseguir dichos objetivos. Es muy importante tener una buena metodología para que todo esté bien estructurado y se vayan superando por partes todos los objetivos.

3.1. Objetivo general

El objetivo general de este trabajo es conseguir que un usuario sea autónomo para realizar filtrados personalizados de la información de los resultados de ajedrez de forma visual e interactiva de una manera sencilla y de esta forma plantear su futuro estudio del material ajedrecístico que hay disponible en diferentes medios como libros, internet, vídeos, etc.

3.2. Objetivos específicos

Para que sea posible el alcance del objetivo general, se tendrá que dividir el trabajo en una serie de objetivos específicos de manera que se vayan cumpliendo y así lograr el propósito del trabajo utilizando las técnicas y herramientas estudiadas en el máster.

Se han establecido dos objetivos específicos necesarios para la consecución del objetivo principal.

- El primer objetivo específico es conseguir un gran número de partidas procesadas y almacenadas de la manera oportuna para poder realizar las visualizaciones con ellas.
- El segundo objetivo específico es la creación de las visualizaciones interactivas y todos los filtros necesarios para que el usuario tenga la mayor libertad y así consiga sacarle el mayor partido a la herramienta.

3.3. Metodología de trabajo

El trabajo empieza con el estudio de los distintos sitios web en los que hay información pública sobre las partidas y que estén correctamente almacenadas, ya que existen sitios en los que

la información de las partidas puede aparecer con un formato distinto al que se busca en este trabajo. Una vez que se tiene el sitio web encontrado, se procede a la captura de los archivos con la información de las partidas. Cuando se tienen todos los archivos se unen en un solo para su procesamiento.

El siguiente paso que se realiza es el procesamiento de los datos. Para esto, se utiliza la base de datos no relacional MongoDB (MongoDB, 2008) que ha sido la utilizada en la asignatura de Métodos de Captura y Almacenamiento de la Información.

El tercer paso consiste en la inclusión de los datos capturados y procesados en la herramienta de visualización elegida. Se utilizará Tableau (Tableau, 2003) para la creación de las visualizaciones, herramienta utilizada en la asignatura de Herramientas de Visualización. También se utilizará MicroStrategy (MicroStrategy, 1989) para la creación de un grafo.

Para terminar, se explicará un ejemplo de uso, se obtendrán las conclusiones del trabajo y se planteará la posibilidad de trabajo futuro y el plan que se puede seguir.

4. Identificación, tecnologías y procesamiento de datos

En esta parte del trabajo se explicará la contribución que se va a realizar con las visualizaciones. No obstante, para realizarlas se va a necesitar, primero identificar las fuentes de datos que son necesarias para que esto sea posible. Una vez se tienen las fuentes bien identificadas, se procede a la captura de la información que se va a representar, a la utilización de una serie de tecnologías para el almacenamiento de la información, su procesamiento posterior con el objetivo de limpiar el *dataset* y obtener el necesario para crear las visualizaciones.

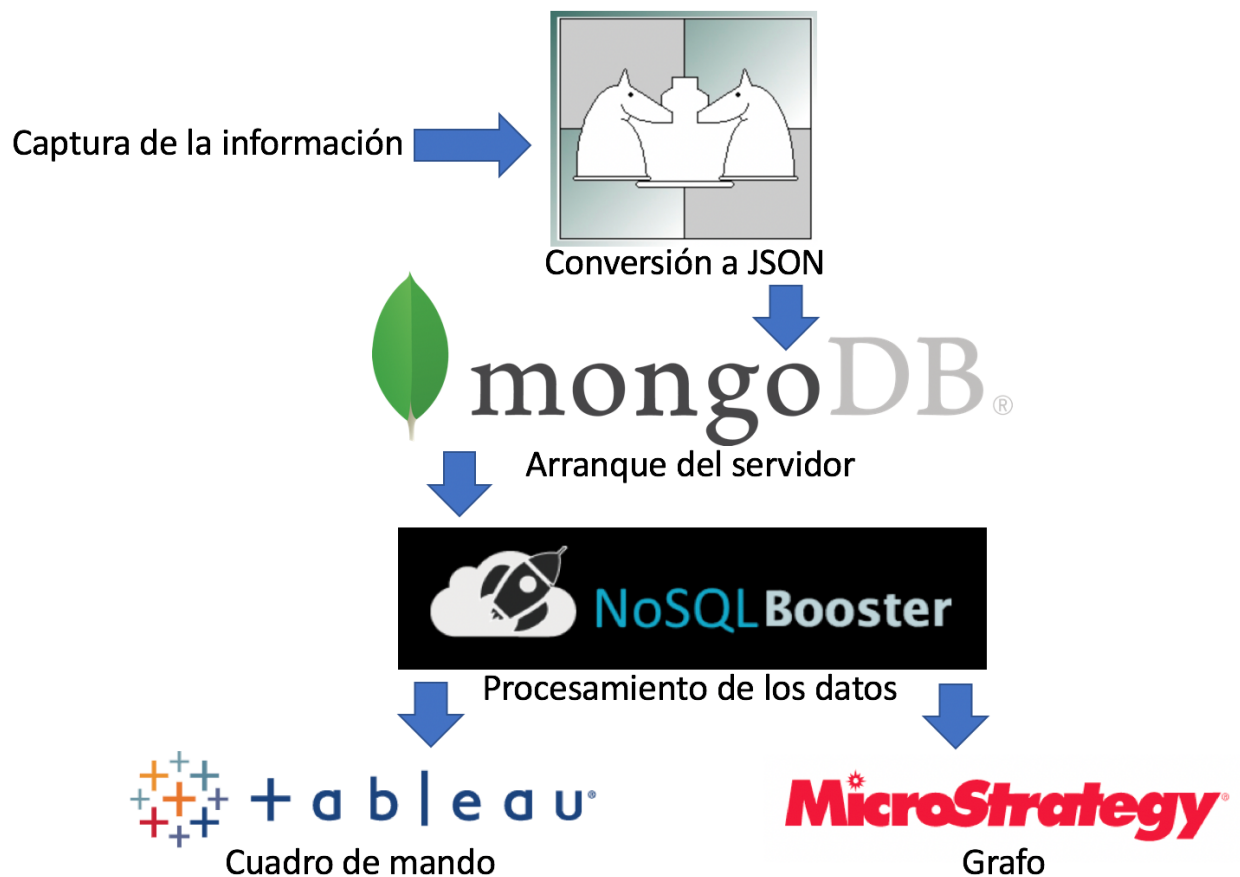


Figura 1. Esquema del desarrollo

4.1. Identificación de las fuentes de datos

Para empezar, se quiere encontrar un lugar del cual obtener la información. Para esto, se investiga en las diferentes plataformas de las que podemos descargar las partidas sobre las que se va a hacer el análisis. Se quiere encontrar las partidas en formato PGN² que es el formato en el que se almacenan.

Todas las partidas se descargan de la página web de PGN Mentor en la sección de *Players*, lugar en el que están almacenadas un gran número de partidas por cada jugador del más alto nivel.

Las partidas tienen la siguiente estructura (ver *Figura 2*):

Event: nombre del torneo en el que se ha disputado la partida.

Site: lugar donde se ha disputado la partida.

Date: fecha de la partida dentro del torneo.

Round: número de la partida de la ronda en el torneo.

White: jugador de las piezas blancas.

Black: jugador de las piezas negras.

Result: resultado obtenido en la partida, siendo éste 1-0, 0-1 o 1/2-1/2.

WhiteElo: puntuación ELO³ del jugador de piezas blancas.

BlackElo: puntuación ELO del jugador de piezas negras.

ECO: código específico de la apertura de la partida.

Después del código ECO aparecen los movimientos efectuados por los jugadores. Empezando el movimiento del blanco seguido del número con un punto y después la respuesta del negro y de nuevo el número de movimiento con un punto hasta finalizar con el resultado obtenido en la partida. Por ejemplo (1. e4 e5) sería el primer movimiento completo de la partida.

² Extensión que se proporciona a los archivos en los que se almacenan las partidas para ser leído por las plataformas informáticas de ajedrez.

³ Sistema de puntuación basado en el cálculo estadístico para valorar la habilidad relativa de los jugadores.

```
[Event "GCT Blitz YourNextMove"]
[Site "Leuven BEL"]
[Date "2017.07.01"]
[Round "9.1"]
[White "Aronian,L"]
[Black "Kramnik,V"]
[Result "1/2-1/2"]
[WhiteElo "2793"]
[BlackElo "2808"]
[ECO "A14"]

1.c4 e6 2.g3 d5 3.Bg2 Nf6 4.Nf3 Be7 5.0-0 0-0 6.d4 dxc4 7.Qc2 a6 8.a4 Bd7
9.Qxc4 Bc6 10.Bf4 Nbd7 11.Nc3 Bd6 12.a5 Bxf4 13.gxf4 Nd5 14.e3 Nxc3 15.Qxc3 Bxf3
16.Bxf3 c6 17.Kh1 Qe7 18.Rg1 g6 19.Rg5 Rad8 20.Rag1 Nf6 21.b4 Nd5 22.Qc5 Qf6
23.Be4 Kh8 24.R1g3 Rfe8 25.Re5 Ne7 26.Reg5 h6 27.Rh3 Kg7 28.Rg1 Nf5 29.Qe5 Qxe5
30.fxe5 Nxd4 31.exd4 Rxd4 32.Re3 Red8 33.Rb1 Rc4 34.Rd3 Rxd3 35.Bxd3 Rd4
36.Rb3 Rd5 37.Kg2 Rxe5 38.Kf3 f5 39.Bc2 Kf6 40.Rd3 Rb5 41.Rd4 c5 42.bxc5 Rxc5
43.Ba4 Rxa5 44.Be8 g5 45.Rd7 Ra3+ 46.Kg2 Rb3 47.Rh7 b5 48.Bd7 Rd3 49.Bc8 Rd6
50.Ra7 b4 51.Rxa6 Rxa6 52.Bxa6 b3 53.Kf3 g4+ 54.Ke3 b2 55.Bd3 Kg5 56.f4+ gxf3
57.Kxf3 e5 58.Ke3 e4 59.Bc2 Kg4 60.Kf2 Kf4 61.Ke2 Ke5 62.Kd2 Kd4 63.Bb1 f4
64.Ke2 Ke5 65.Bc2 Kd4 66.Bb1 e3 67.Bh7 Kc3 68.Bb1 Kb3 69.Kf3 Kc4 70.Ke2 Kc3
71.Bh7 Kd4 72.Bb1 Ke5 73.Kf3 Kf6 74.h4 Ke5 75.Bc2 Kd4 76.Bb1 Kc4 77.Ke2 Kc3
78.Bh7 Kb4 79.Bb1 Kb3 80.Kf3 Kc3 1/2-1/2
```

Figura 2. Ejemplo de la estructura de una partida de ajedrez en formato PGN

Habiendo identificado la estructura de la partida se tendrá que realizar una conversión de los datos a un formato que se pueda utilizar en la base de datos.

Durante el máster se ha aprendido a utilizar la base de datos no relacional MongoDB y será ésta la que se utilice para almacenar y procesar los datos.

Se van a convertir todos los ficheros almacenados en formato PGN en el de JSON⁴ para así poder cargar los archivos en MongoDB.

Una vez que ya se tiene importada toda la información en la base de datos, se procederá a procesarla y quedarse únicamente con los campos que son relevantes para los análisis. Cuando ya se tiene todo listo, se utilizará el software de visualización interactiva Tableau, en el que se desarrollarán las visualizaciones. Se utilizará también MicroStrategy para la construcción de grafos.

⁴ Es el subconjunto de notación de objetos de JavaScript

4.2. Tecnologías utilizadas

Las tecnologías que se han utilizado en este trabajo son: CaissaTools.jar (Booij, 2018) para la conversión de PGN a JSON, la base de datos no relacional MongoDB, el software de visualización interactiva Tableau y MicroStrategy. Se explicarán detalladamente cada uno a continuación.

Script CaissaTools.jar

CaissaTools.jar es un archivo JAVA creado por Marco de Booij, un jugador neerlandés aficionado del ajedrez, para la conversión de archivos con el formato PGN a diversos otros formatos como pueden ser: JSON o LaTeX.

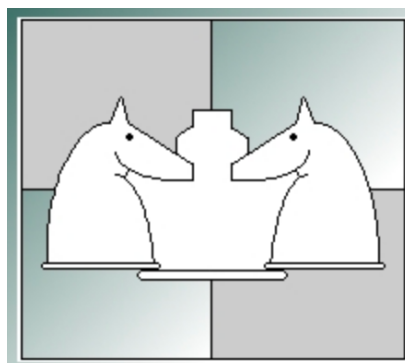


Figura 3. Logo de la página web de Marco de Booij

En la página web de Marco⁵ se pueden encontrar tanto el enlace de descarga del script como la explicación de su uso.

Applicatie	package	GitHub
Caissa		
Caissa (Core)	0.1.13	CaissaCore
Caissa Tools	1.2.0	CaissaTools

Figura 4. Descarga de CaissaTools

⁵ <https://www.debooy.eu/Java/caissatools.html>

Se descarga el archivo alojado en el servidor de GitHub. En la web se tiene la explicación detallada de todas las opciones disponibles en el script de CaissaTools.

```

+-----+
|      | |      | |      | |      | |      | |      |
|  | \   *   _/ | | PGN to JSON |
|  /   .\ *** / . \ |
|  | ( _ \ * / _ ) | |
+---+ | \ ) ( / | +-----+
|  |      | |      | |      | |      |
|  / _ \ \ / _ \ |
| [ ] [ ] | E-Mail : noreply@debooy.eu
| [ ] | Website: https://www.debooy.eu
+-----+
v1.2.0 | 2017-12-28 19:45:25

java -jar CaissaTools.jar PgnToJson [OPTION...] --bestand=<PGN bestand>

--bestand    The file with games in PGN format.
--charsetin  The charset of if it is different from UTF-8.
--charsetuit The charset of the output if it is different from UTF-8.
--invoerdir  Directory from which the inputfiles are read.
--json       The file with the games in JSON format.
--pgnviewer  Add information for tee PGNViewer .
--uitvoerdir Directory in which the output files are written.

Only parameter bestand is mandatory.

```

Figura 5. Explicación de las opciones de CaissaTools

Se puede ver la explicación detallada de cada opción (ver *Figura 5*) en la que se muestran los parámetros que se deben de introducir para convertir un archivo en formato PGN a JSON siendo el parámetro *bestand* obligatorio para su conversión.

El script consiste en una serie de archivos programados en el lenguaje JAVA comprimidos todos ellos en un archivo JAR para su utilización desde la terminal. Se explicarán detalladamente los pasos dados para su conversión en el apartado de captura de los datos.

MongoDB

Se utilizará la base de datos no relacional MongoDB para el almacenamiento y el procesamiento de la información ya que es la que se ha utilizado en el máster para estos propósitos.

MongoDB es una base de datos NoSQL de código abierto que utiliza una estructura similar a JSON llamada BSON para el almacenamiento de la información.

Se empezó su desarrollo en 2007 pero hasta el año 2009 no se lanzó como producto independiente bajo la licencia de código abierto de AGPL.



Figura 6. Logo de MongoDB

Se utiliza esta base de datos porque es no relacional, ya que las partidas descargadas son de diferentes jugadores y puede haber campos sin definir en alguna de ellas. Utilizando MongoDB se evita el problema con los campos de una base de datos relacional y se podrá procesar la información sin problemas.

En la página web de MongoDB se puede descargar gratuitamente la versión compatible con el sistema operativo que se esté utilizando (ver Figura 7). Se accede a su sitio web⁶.

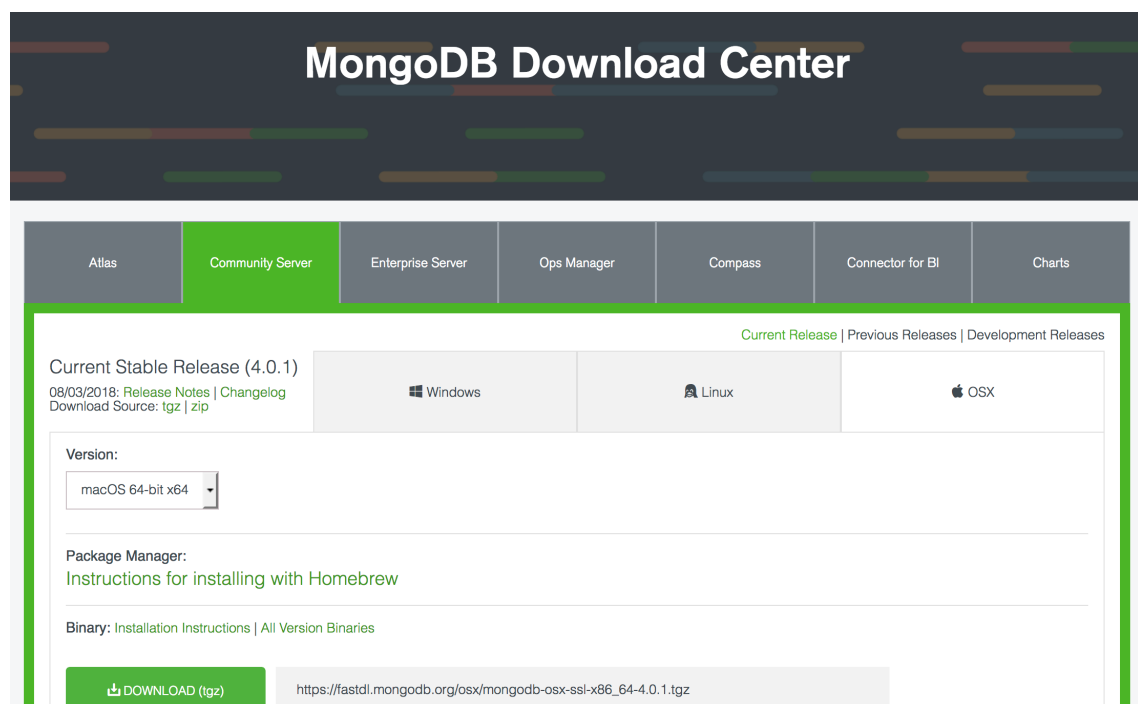


Figura 7. Sitio web de descarga de MongoDB

⁶ <https://www.mongodb.com/download-center?jmp=nav#community>

NoSQLBooster for MongoDB

Se utiliza el GUI NoSQLBooster for MongoDB (NoSQLBooster, 2018) para la inclusión de la información y su procesamiento.

Es una aplicación GUI que se conecta al servidor que se ha montado de MongoDB para acceder a la información y utilizarla de manera interactiva con un sistema de peticiones *query* fluido e intuitivo para realizar el procesamiento de una forma más sencilla



Figura 8. Logo de NoSQLBooster

Para descargarse el programa NoSQLBooster for MongoDB se accede a su sitio web⁷ (ver Figura 9).

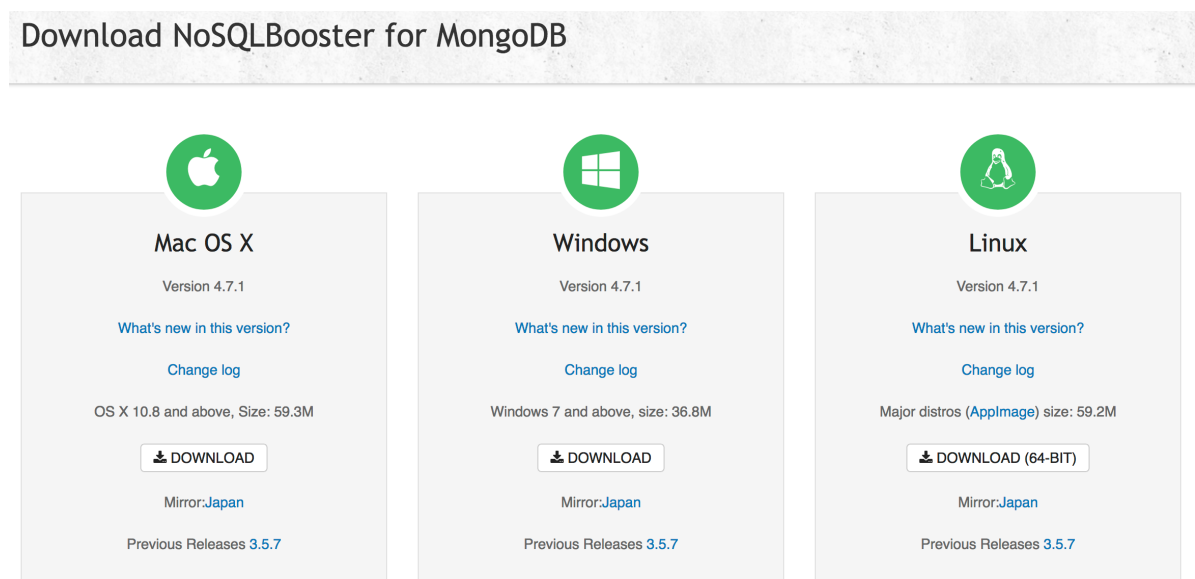


Figura 9. Sitio web de descarga de NoSQLBooster for MongoDB

Se puede ver en la imagen que este software también es compatible con los sistemas operativos principales por lo que se puede instalar en cualquier ordenador.

⁷ <https://nosqlbooster.com/downloads>

Se elige este software porque tiene una versión limitada pero gratuita que servirá perfectamente para la realización del procesamiento de los datos que se quiere conseguir. Ésta es la interfaz de NoSQLBooster (ver *Figura 10*).

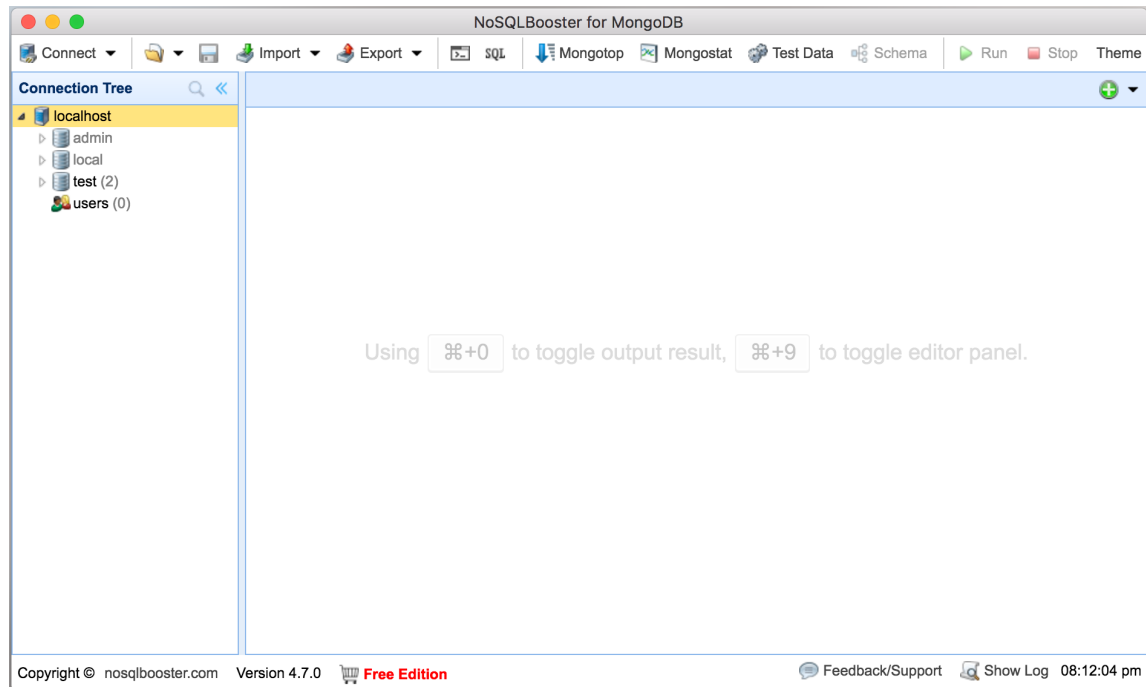


Figura 10. Interfaz de NoSQLBooster

Tableau Software

Para el desarrollo principal de las visualizaciones se va a utilizar el software de visualización interactiva Tableau.

Tableau fue creado por tres personas de la universidad de Stanford en California con el objetivo de crear una aplicación que consiguiera hacer los datos y la información entendible para cualquier persona.



Figura 11. Logo de Tableau Software

Se elige Tableau porque es una herramienta sencilla de utilizar, con mucha potencia y resultados muy buenos como se explica en el libro “Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software” (Murray, 2016).

En la asignatura “Herramientas de Visualización” es el principal software utilizado y por ese motivo, también, se elige esta aplicación.

Para descargar Tableau, se accede a su sitio web⁸ y se pulsa el botón de *TRY NOW*. Una vez allí se pide un email corporativo para acceder a un *trial* de 14 días (ver *Figura 12*).

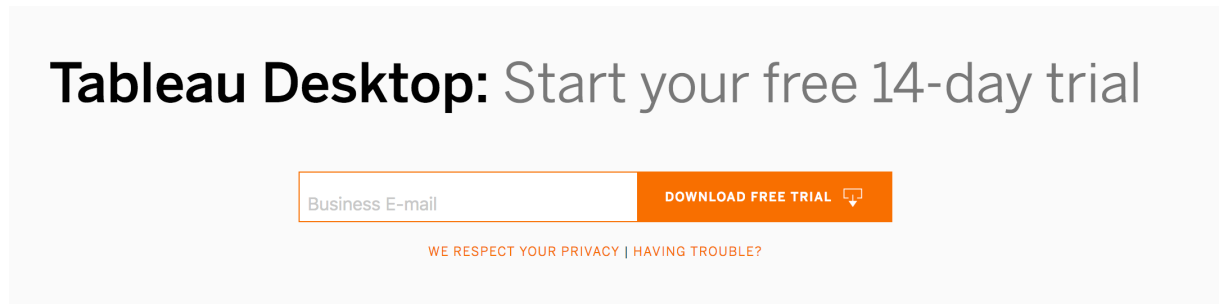


Figura 12. Sitio web descarga de Tableau Desktop

Existe una versión llamada Tableau Public para acceder y utilizar las visualizaciones creadas con Tableau Desktop, pero sin poderlas modificar, lo que quiere decir que cualquier usuario puede utilizar gratuitamente todo el contenido desarrollado en el trabajo. Gracias al máster se tiene una versión completa durante un año.

Tableau también tiene numerosos tutoriales para aprender a utilizar sus herramientas de forma gratuita en su página web. Simplemente entrando en *Learning* en la parte superior de la página principal se accede a una serie de vídeos que están disponibles para los usuarios registrados. La cuenta de acceso se puede crear en sencillos pasos y de forma gratuita.

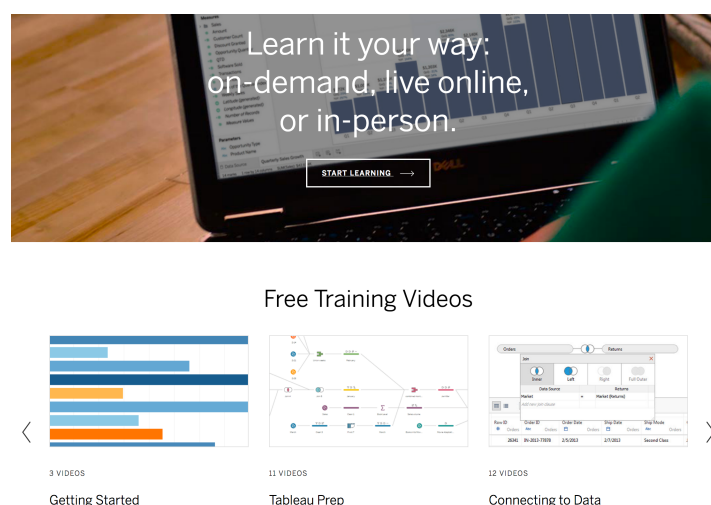


Figura 13. Tutoriales de Tableau oficiales

⁸ <https://www.tableau.com/>

Una vez dentro encontramos varios apartados con todos los temas y capítulos de la información que se incluye en cada tutorial (ver *Figura 14*). Existen un total de más de seis horas de vídeos solamente para la parte de creador, también se tienen más secciones como administrador, visualizador, explorador, etc. Para el trabajo se utilizarán los tutoriales de la sección de creador.

Creator

Are you doing deep data prep and analysis? Responsible for creating content for others? If you have Tableau Prep and Tableau Desktop, these videos are for you. Learn how to prepare, analyze, and share your data.

3 VIDEOS	Getting Started	34 MIN
11 VIDEOS	Tableau Prep	62 MIN
11 VIDEOS	Connecting to Data	62 MIN

Figura 14. Ejemplo de los temas para la creación de visualizaciones con Tableau

En el libro de “Tableau 10 for Beginners: Step by Step guide to developing visualizations in Tableau 10” (Sinha, 2017) se explica perfectamente cada parte del entorno de Tableau. (ver *Figura 15*).

La interfaz de trabajo de Tableau para el desarrollo de visualizaciones está dividida principalmente en tres espacios:

- Lateral izquierdo: donde aparecen todas las variables separadas en dimensiones y medidas dependiente de qué tipo de dato sean.
- Parte central: donde se ven las visualizaciones que se están desarrollando junto con los campos a los que se pueden arrastrar las variables a utilizar.
- Parte inferior: aparecen las pestañas de la fuente de datos utilizada y todas las hojas de las visualizaciones que se van haciendo junto a los cuadros de mando.

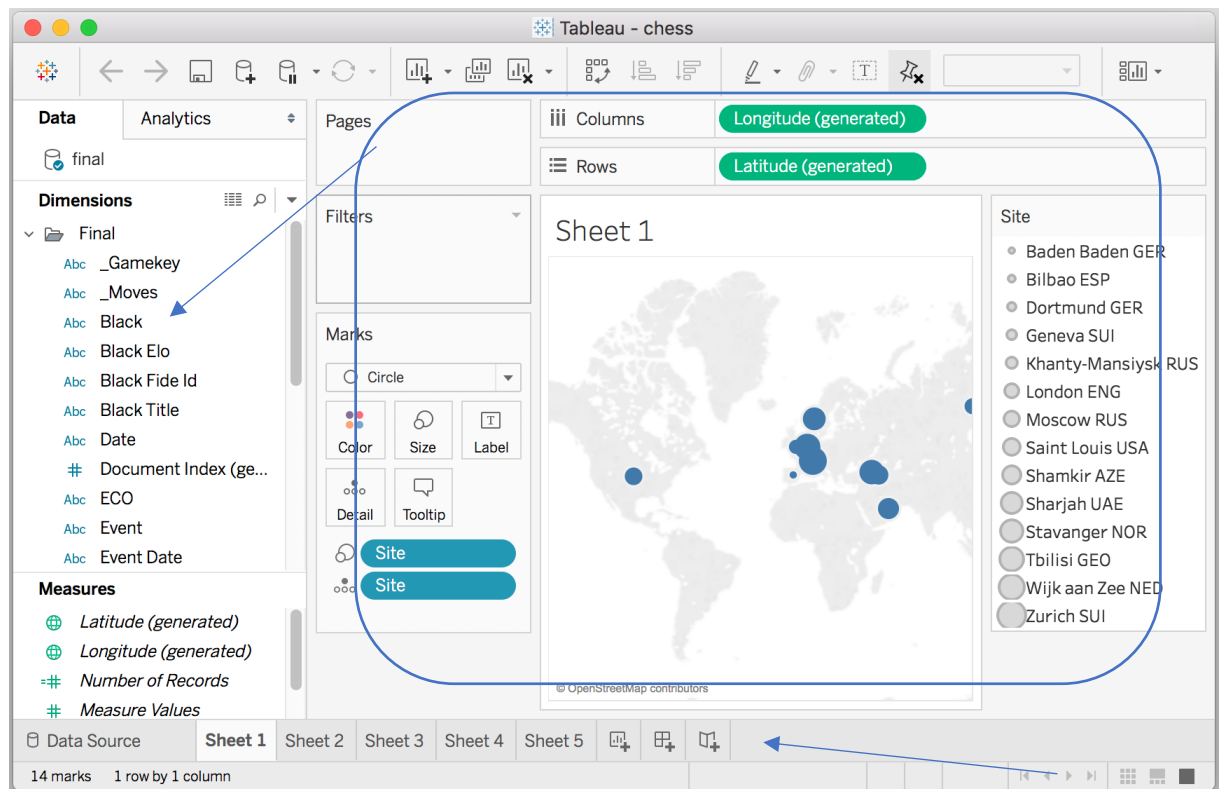


Figura 15. Interfaz de Tableau Desktop

MicroStrategy

El software de MicroStrategy se va a utilizar para el desarrollo de algunos grafos que mostrarán la relación que existe en el número de partidas que juegan los jugadores entre ellos.

MicroStrategy es una herramienta de inteligencia de negocio gratuita y fácil de utilizar como nos explica Cristián Gómez en su libro “Data Discovery con MicroStrategy 10: Una Guía Práctica” (Gómez, 2015)



Figura 16. Logo de MicroStrategy

Los grafos se crearán desde la aplicación de MicroStrategy Desktop, aunque también existe la posibilidad de utilizar MicroStrategy en su versión online.

Se muestra un ejemplo de la interfaz principal de la creación de un *Dossier* (ver Figura 17) en MicroStrategy.

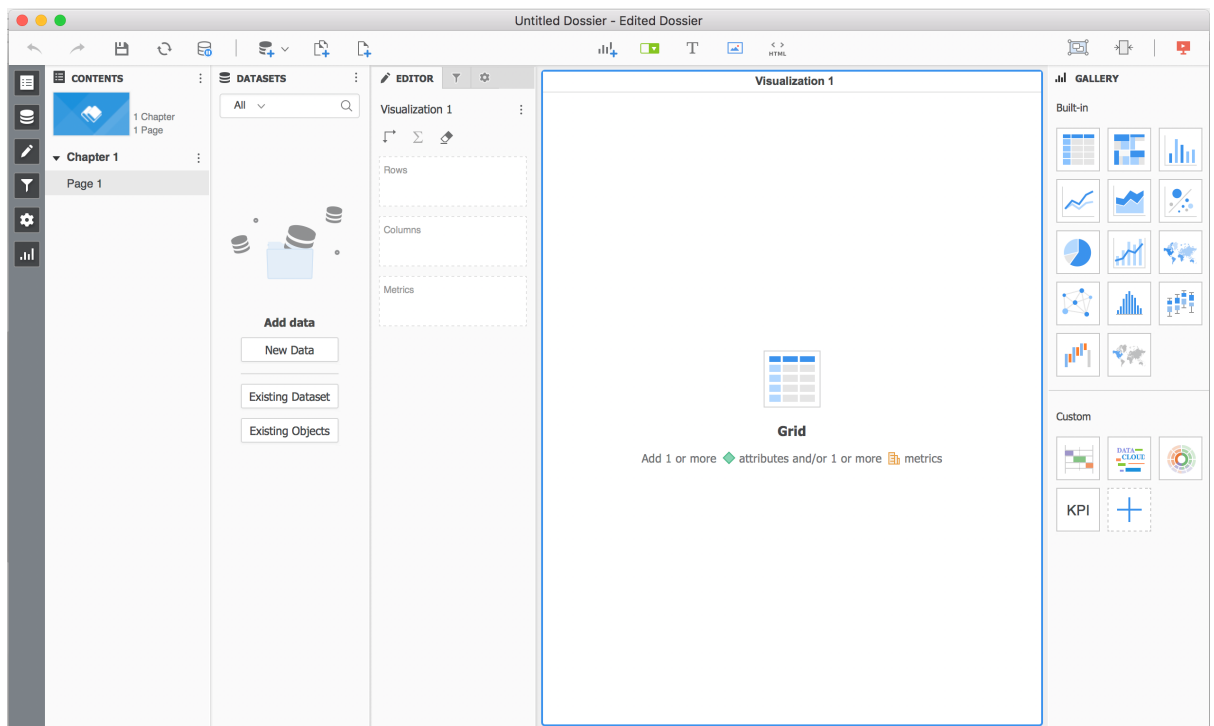


Figura 17. Interfaz de creación de Dossier en MicroStrategy

4.3. Captura de los datos

La fase de la captura de datos se realiza desde el sitio web de PGN Mentor en la sección de *Players*⁹.

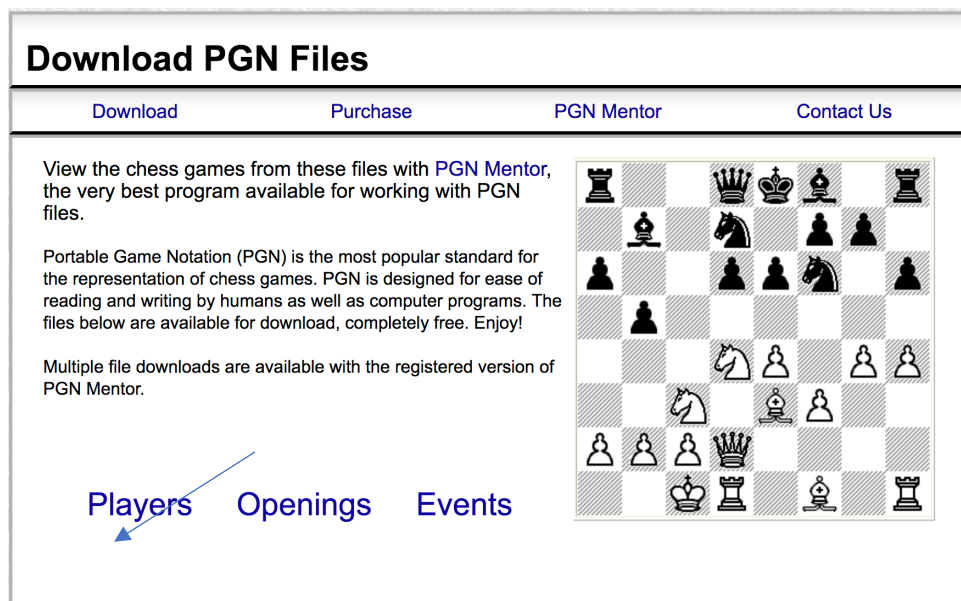


Figura 18. Página de selección de partidas por jugador

⁹ <https://www.pgnmentor.com/files.html#players>.

Se descarga un archivo PGN por cada jugador, seleccionando aquellos que están más activos y que tienen el mejor nivel según la página de 2700chess (2700Chess, 2018). Para ello, se accede al botón de *Download* que hay debajo del nombre de cada jugador. También se indica el número de partidas que contiene el archivo (ver *Figura 19*).

No se utiliza ninguna técnica de *Web Scrapping* para el acceso a la información ya que de manera sencilla con tan sólo acceder al botón de *Download* de cada jugador ya se tendrán las partidas en el formato requerido para su futura conversión a JSON.

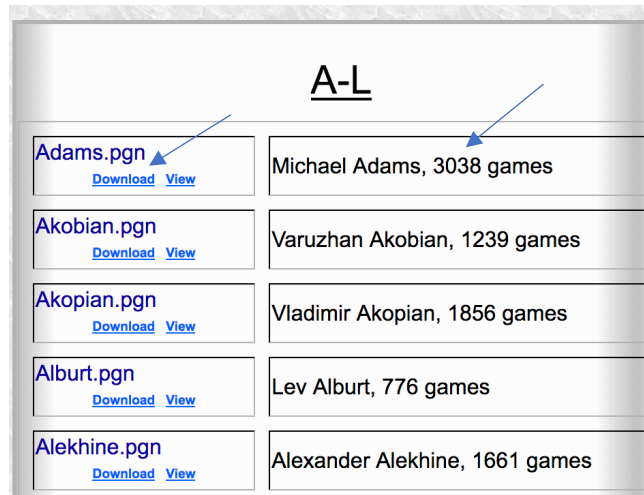
Los jugadores de ajedrez seleccionados, para la descarga de sus partidas, son once pertenecientes al top 15 de los jugadores con más nivel y son ajedrecistas muy activos en la escena de hoy en día. A la lista de los once se le añade también el mejor jugador de ajedrez español, Francisco Vallejo Pons, para así tener también una representación española por si algún usuario desea conocer qué tipo de aperturas juega.

La lista de los jugadores que componen las 28287 partidas de las que se va a sacar la información ordenados por nivel hoy en día es:

- *Magnus Carlsen*
- *Fabiano Caruana*
- *Shakhriyar Mamedyarov*
- *Anish Giri*
- *Levon Aronian*
- *Maxime Vachier-Lagrave*
- *Vladimir Kramnik*
- *Wesley So*
- *Viswanathan Anand*
- *Hikaru Nakamura*
- *Sergey Karjakin*
- *Francisco Vallejo Pons*

Cuando ya se tienen descargados todos los archivos con las partidas que pertenecen a los jugadores de los cuales se quiere hacer el análisis, se procede a juntarlas todas en un único archivo PGN, para que así, en la conversión a JSON, se obtenga solamente otro archivo con toda la información necesaria para añadirlo a la base de datos desde la cual se hará el procesamiento de la información.

Se utiliza la página de PGN Mentor porque tiene acceso gratuito a las partidas archivadas y se puede hacer pleno uso de ellas ya que son de uso público.



A-L	
Adams.pgn Download View	Michael Adams, 3038 games
Akopian.pgn Download View	Varuzhan Akopian, 1239 games
Akopian.pgn Download View	Vladimir Akopian, 1856 games
Albur.pgn Download View	Lev Albur, 776 games
Alekhine.pgn Download View	Alexander Alekhine, 1661 games

Figura 19. Ejemplo de descarga de las partidas en PGN Mentor

4.4. Estudio y procesamiento de los datos

Para esta parte, es necesario que se estudie bien el formato que tenemos de la información e identificar qué campos son los que se van a utilizar y prescindir de todos los demás.

Es muy importante entender bien qué es necesario y qué no para que las visualizaciones muestren la información correctamente.

Como se ha explicado en el paso anterior se procede a unir todos los archivos en uno solo con formato PGN. En este archivo se incluyen todas las partidas de todos los jugadores propuestos para el estudio y para el desarrollo de las visualizaciones pertinentes.

Para unir todos los archivos en uno, se realiza la copia de la información de cada fichero en uno nuevo en el que van a ir secuencialmente uno detrás de otro hasta terminar con todos ellos.

Conversión de PGN a JSON

Una vez que se tiene el archivo PGN con toda la información que se va a procesar hay que abrir la terminal para la conversión del formato PGN a JSON.

Se utiliza la herramienta de Marco de Booyj, CaissaTools.jar en la que se pondrán los comandos necesarios según la explicación que se ofrece en el sitio web de Marco¹⁰.

El comando que se va a utilizar es el siguiente:

```
java -jar CaissaTools.jar PgnToJson [OPTION...] --bestand=<PGN bestand>
```

Para que todo funcione correctamente se va a colocar el script CaissaTools.jar en la misma carpeta en la que se tiene el archivo PGN con todas las partidas a convertir a JSON. A dicho archivo se le llamará partidas.pgn. Es necesario acceder desde la terminal a la carpeta en la que se tienen estos dos archivos y ejecutar el siguiente comando en la terminal: “java -jar CaissaTools.jar PgnToJson --bestand = partidas.pgn”.

Nótese que se ha cambiado <PGN bestand> por el nombre del fichero que se va a convertir. Una vez ejecutado el comando aparece un mensaje de verificación (ver *Figura 20*) en el que se puede ver el nombre de fichero de origen, el número de partidas que había en él (28287 partidas) y el nombre de fichero de destino seguido de un *Ready* que indica que todo el proceso ha ido correctamente. Como no se utiliza ninguna otra opción, el nombre del nuevo fichero tiene que coincidir con el nombre del fichero original, pero con la extensión JSON.

```

cma:Chess games cristianmartinamores$ java -jar CaissaTools.jar PgnToJson --bestand=partidas.pgn
+-----+
|      |      |      |      |      |      |      |      |      |      |
|  | \_  |  *  |  _/_/ |      |      |      |      |      |
|  /  .  | *** | /_  |      |      |      |      |      |
|  (  \_ | * /  |      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+
|  | \_  |  (  |  /_  |      |      |      |      |      |
|  /  .  |  /_  |      |      |      |      |      |
|  [  _  ]  [  _  ]      |      |      |      |      |
|  [  _  ]  [  _  ]      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+
|      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+
v1.2.0 | 2017-12-28 19:45:25

File : ./partidas.pgn
Games : 28287
Output: ./partidas.json
Ready.
cma:Chess games cristianmartinamores$

```

Figura 20. Conversión de PGN a JSON con CaissaTools.jar

¹⁰ https://www.debooy.eu/Java/caissatools_en.html#pgnjson

El nuevo archivo se encuentra en la misma carpeta en la que estaba el archivo original ya que no se ha indicado otra ruta distinta y por defecto utiliza ésta.

El formato JSON es necesario para añadirlo a la base de datos en MongoDB. Para realizar este proceso primero tenemos que iniciar el servidor de MongoDB.

En el apartado 4.2 Tecnologías utilizadas se explica cómo descargarse la versión necesaria para realizar el trabajo.

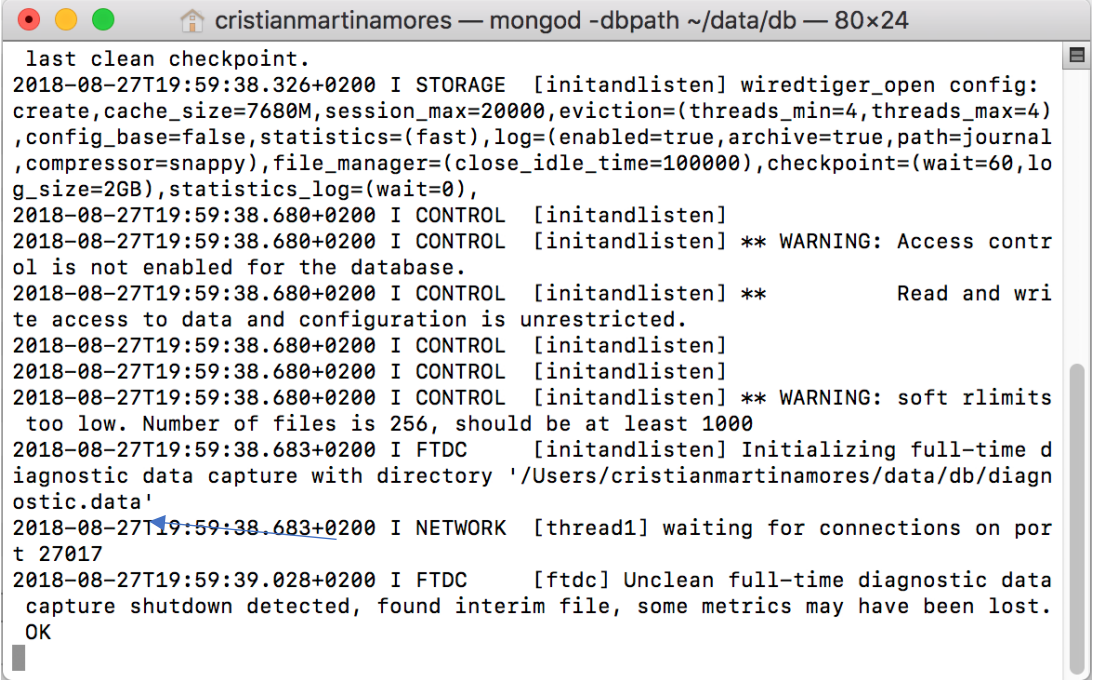
Arranque servicio mongod

Una vez que se tiene todo descargado, se abre la terminal y se lanza el servicio *mongod* como se explica en la figura (ver *Figura 20*) en la que se puede ver que se ha añadido el *-dbpath* con la ruta en la cual se tiene la base de datos. Por defecto, cuando se instala MongoDB la carpeta *data/db* está situada en el directorio raíz.



Figura 21. Comando de arranque para mongod

Cuando se tiene arrancado el servidor de mongod (ver *Figura 22*) ya se puede acceder desde otra terminal con el comando *mongo* y utilizar los comandos propios de MongoDB para crear la base de datos, las colecciones, etc. En este trabajo ya se ha hablado de que se va a utilizar el GUI de NoSQLBooster for MongoDB y la consola de *mongo* se usará únicamente para exportar la información.

A terminal window titled 'cristianmartinamores — mongod -dbpath ~/data/db — 80x24'. It displays the output of the 'mongod' command. The logs show the server starting, including messages about the wiredtiger engine configuration, control plane status, warnings about access control and soft limits, and network initialization. The server successfully starts and begins listening on port 27017.

```
last clean checkpoint.
2018-08-27T19:59:38.326+0200 I STORAGE  [initandlisten] wiredtiger_open config:
create,cache_size=7680M,session_max=20000,eviction=(threads_min=4,threads_max=4)
,config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal
,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,lo
g_size=2GB),statistics_log=(wait=0),
2018-08-27T19:59:38.680+0200 I CONTROL  [initandlisten]
2018-08-27T19:59:38.680+0200 I CONTROL  [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2018-08-27T19:59:38.680+0200 I CONTROL  [initandlisten] **           Read and wri
te access to data and configuration is unrestricted.
2018-08-27T19:59:38.680+0200 I CONTROL  [initandlisten]
2018-08-27T19:59:38.680+0200 I CONTROL  [initandlisten]
2018-08-27T19:59:38.680+0200 I CONTROL  [initandlisten] ** WARNING: soft rlimits
too low. Number of files is 256, should be at least 1000
2018-08-27T19:59:38.683+0200 I FTDC    [initandlisten] Initializing full-time d
iagnostic data capture with directory '/Users/cristianmartinamores/data/db/diagn
ostic.data'
2018-08-27T19:59:38.683+0200 I NETWORK  [thread1] waiting for connections on por
t 27017
2018-08-27T19:59:39.028+0200 I FTDC    [ftdc] Unclean full-time diagnostic data
capture shutdown detected, found interim file, some metrics may have been lost.
OK
```

Figura 22. Servidor mongod correctamente arrancado

Importación de las partidas a MongoDB

Ahora que ya está arrancado el servidor, se abre la aplicación de NoSQLBooster for MongoDB y se pulsa Connect (ver Figura 23) para conectarse al servidor. El puerto por defecto es el 27017 y éste se puede comprobar en la terminal en la que hemos ejecutado el comando *mongod*.

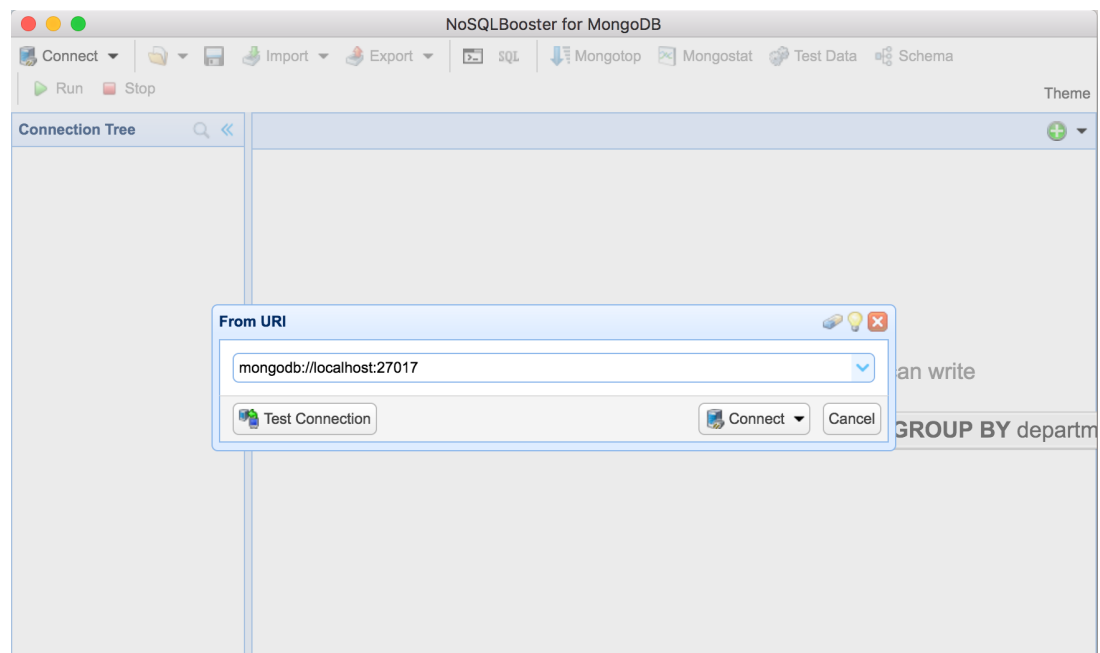


Figura 23. Conexión a la base de datos desde NoSQLBooster for MongoDB

Pulsando de nuevo a Connect se conecta a la base de datos y se utilizará la que se llama *test* ya que es una versión gratuita y está limitada al uso exclusivo con ese nombre. Una vez en ella se importará el archivo de *partidas.JSON* que se ha generado en la conversión con el script CaissaTools.jar.

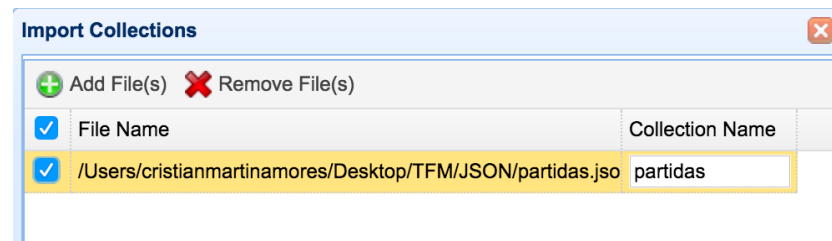


Figura 24.Importación del archivo JSON

Para esto, se accede a *Import* en la parte superior de la aplicación y se elige la opción de *Import from JSON* en la que se seleccionará el archivo que se va a importar, pulsando a *Add file* (ver Figura 24). Cuando está seleccionado el archivo, se pulsa en *Import* y aparecerá la información de cuántos archivos se han importado y cuánto tiempo ha tardado en ejecutar la acción (ver Figura 25).

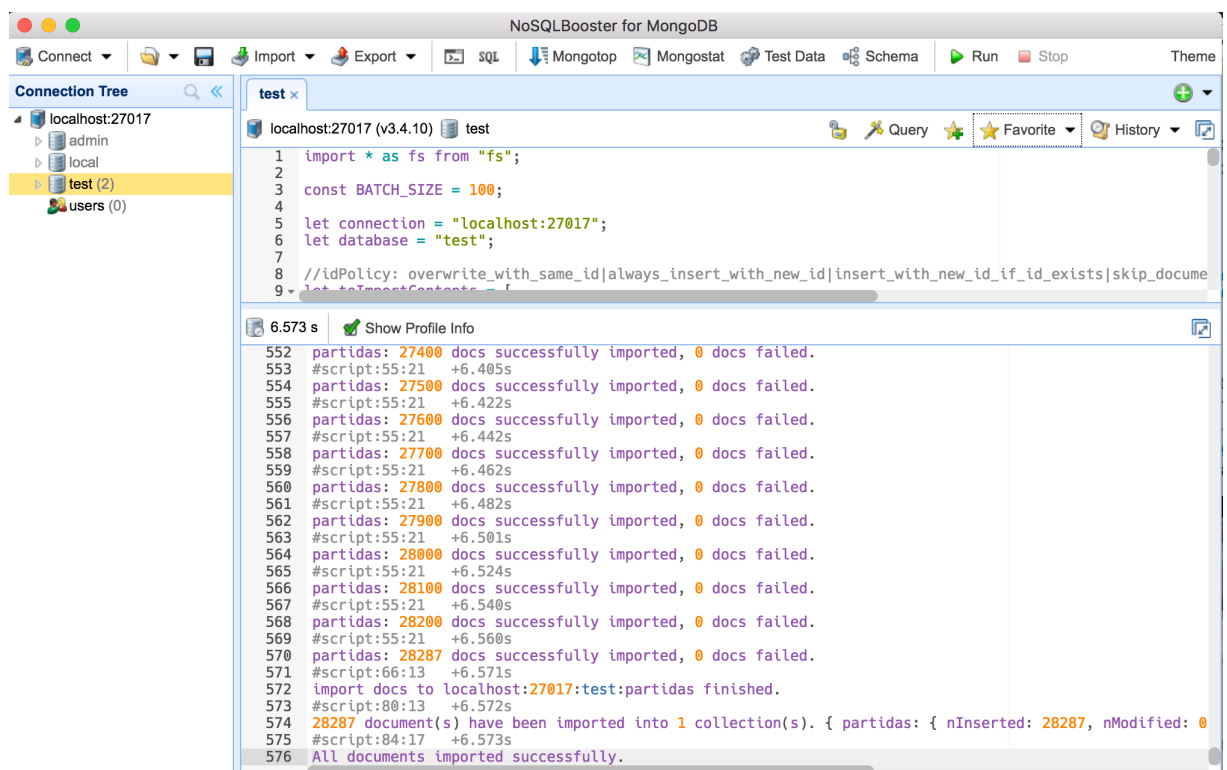


Figura 25.Documentos importados en MongoDB

Procesamiento de los datos en MongoDB

Con todo esto terminado, ya se puede empezar el procesamiento de los datos de acuerdo con el estudio que se va a realizar de los mismos para determinar con qué campos quedarse y el proceso de limpieza la información.

Para realizar el procesamiento de datos se van a utilizar los manuales que se tienen disponibles en el sitio de web de MongoDB en el apartado de documentación (MongoDB Documentation, 2008). En la página se utilizará el buscador para encontrar aquellos comandos que nos sean útiles en el procesamiento de los datos.

Antes de comenzar, hay que darse cuenta de que las partidas han sido descargadas por jugador de ajedrez, por lo tanto, es posible que las partidas que se juegan entre los doce candidatos elegidos estén duplicadas, por ejemplo, una partida entre Magnus Carlsen y Sergey Karjakin aparecerá una vez en cada archivo, una en el de Magnus Carlsen y otra vez en el archivo de Sergey Karjakin.

Para solucionar este problema en las versiones anteriores de MongoDB bastaba con asegurar un índice único por el campo que no se quisiera tener repetido y de esta manera eliminar todos los documentos en los que coincidiera la clave elegida en la asignación del índice. El problema que se plantea es que en las nuevas versiones ya no se puede utilizar el comando *dropDups: true*.

Por lo tanto, se necesita buscar una nueva alternativa para el borrado de documentos duplicados. La manera de identificar si hay documentos duplicados no es simplemente porque aparezca el mismo jugador con negras y el mismo con blancas en varios documentos, ya que estos jugadores pueden haberse enfrentado en numerosas ocasiones.

La forma que se utiliza para identificar si los documentos son iguales es el uso del campo *_moves* ya que no existen dos partidas completas iguales en la base de datos, por lo tanto, este campo tiene que ser único.

```

1  var duplicates = [];
2
3  db.partidas.aggregate([
4    { $group: {
5      _id: { moves: "$_moves" }, // Elegimos el campo _moves
6      duplis: { "$addToSet": "$_id" },
7      count: { "$sum": 1 }
8    }},
9    { $match: {
10     count: { "$gt": 1 } // Si count > 1 serán documentos duplicados
11   }}
12 ])
13 .forEach(function(doc) {
14   doc.duplis.shift(); // El primer elemento no se elimina
15   doc.duplis.forEach(function(duplisId){
16     duplicates.push(duplisId);
17   })
18 })
19 })
20 db.partidas.remove({_id:{$in:duplicates}}) // Se eliminan todos los duplicados

```

Figura 26. Script en Aggregation framework de MongoDB

Para realizar este proceso en las nuevas versiones se utilizará el aggregation framework de MongoDB con el que se encontrará cuándo aparece más de una vez el mismo contenido en el campo `_moves` (ver Figura 25).

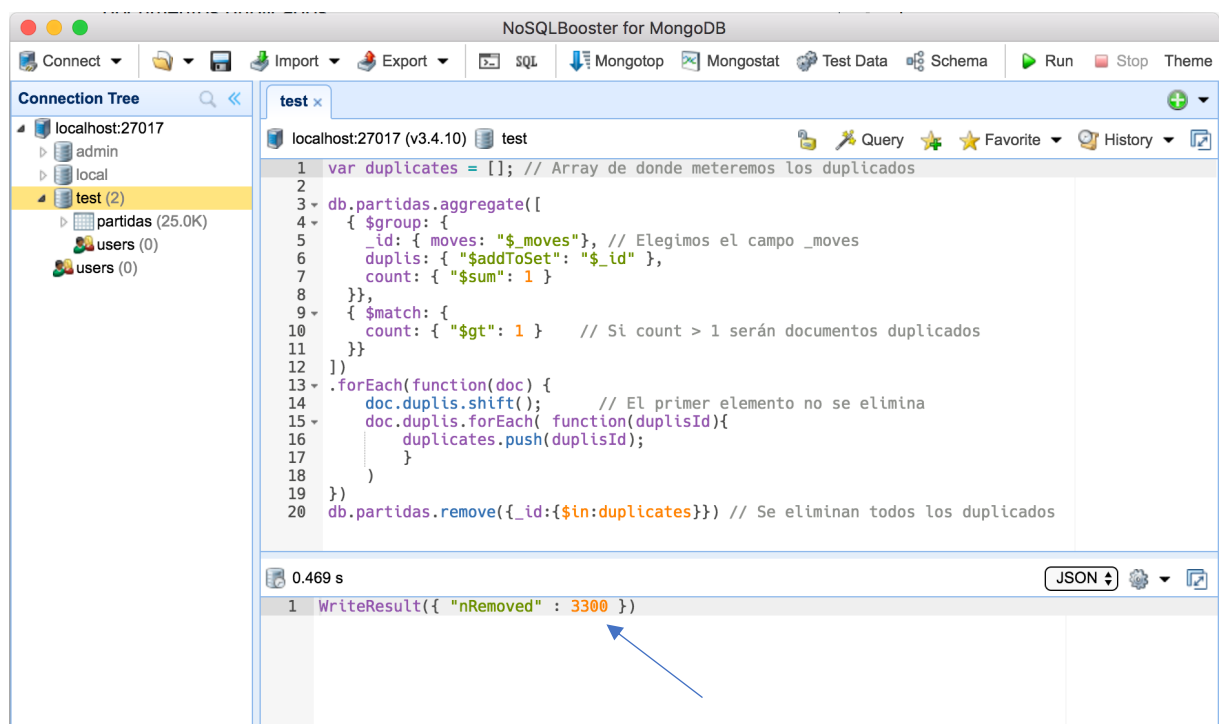


Figura 27. Número de documentos duplicados eliminados

Una vez ejecutado el código en NoSQLBooster se puede ver que 3300 documentos (ver *Figura 27*) han sido borrados obteniendo un total de 24987 partidas para los análisis posteriores.

Una vez que se tienen todos los duplicados eliminados hay que decidir qué campos van a ser interesantes para el desarrollo de las visualizaciones. Los campos que hay en la base de datos de MongoDB por orden de inclusión son:

- `_id`: es el identificador por defecto en MongoDB creado de forma aleatoria al importar los archivos.
- `_gamekey`: es el identificador por defecto que crea CaissaTools.jar al convertir el formato PGN a JSON.
- `Black`: nombre del jugador de piezas Negras.
- `BlackElo`: ELO del jugador de piezas Negras.
- `Date`: fecha de la partida.
- `ECO`: código específico de la apertura.
- `Event`: nombre del evento.
- `Result`: resultado obtenido.
- `Round`: ronda dentro del evento.
- `Site`: lugar de disputa de la partida.
- `White`: nombre del jugador de piezas blancas.
- `WhiteElo`: ELO del jugador de piezas blancas.
- `_moves`: lista de los movimientos realizados en la partida.

Los campos que no interesan para el desarrollo de las visualizaciones serán eliminados de la base de datos. Los campos `_id` y `_gamekey` son irrelevantes para el desarrollo ya que son simplemente un identificador del documento. El campo `_id` es propio del sistema de MongoDB por lo tanto no puede ser eliminado. Tanto *BlackElo* como *WhiteElo* no aportan información relevante para las visualizaciones ya que todos los jugadores son del más alto nivel y la diferencia entre ellos es muy pequeña, por lo tanto, los dos campos serán eliminados. *Date*, que es la fecha de la partida tampoco aporta información para conseguir el objetivo del trabajo. *Round*, es el número de la ronda dentro del evento y tampoco proporciona información relevante.

Por lo tanto, repasando todos los campos, los que son útiles para el desarrollo de las visualizaciones son los siguientes:

- Black
- ECO
- Event
- Result
- White
- _moves

Los campos que no son relevantes para las visualizaciones, se procede a eliminarlos con el siguiente comando:

```
db.partidas.update({},{$unset: {_gamekey:1, BlackElo:1, Date:1, Round:1, WhiteElo:1 }},{multi: true})
```

Figura 28. Eliminación de campos

Se puede apreciar que la ejecución afecta al total de 24987 documentos que hay en la base de datos (ver Figura 30). También se cambia el nombre del campo _moves a *Moves* utilizando el comando *update* con *\$rename*.

```
db.partidas.update({},{$rename: {_moves:"Moves"}},{multi: true})
```

Figura 29. Cambio de nombre de _moves a *Moves*

Ahora ya se tienen todos los campos necesarios con los nombres correspondientes

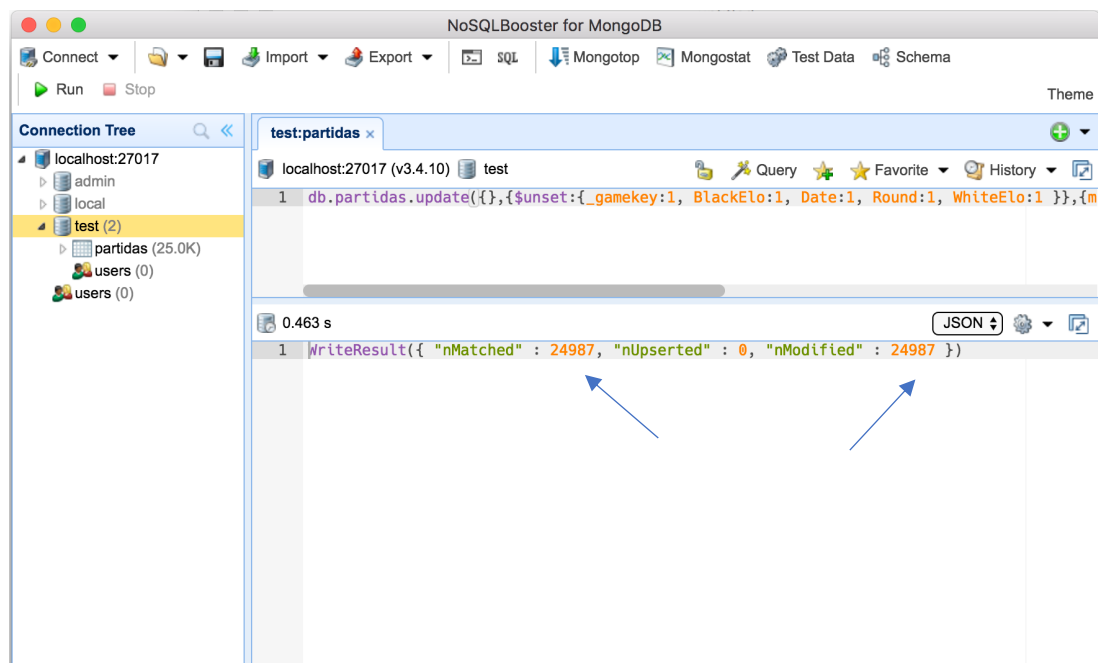


Figura 30. Eliminación de los campos irrelevantes para el estudio

Hay que realizar también una modificación en el campo *Result* ya que los resultados aparecen como

- 1-0: Victoria para el jugador de piezas blancas
- 0-1: Victoria para el jugador de piezas negras
- 1/2-1/2: Tablas, empate.

Los nuevos resultados serán White para cuando el resultado sea 1-0, Black cuando el resultado sea 0-1 y Draw para el resultado que falta de 1/2-1/2. Se realiza el cambio de la información en el campo *Result* utilizando tres comandos *update* distintos y ejecutados por separado, cambiando primero el resultado de 1-0, luego de 0-1 y finalmente el resultado de 1/2-1/2.

```
db.partidas.update({Result:"1-0"},{$set:{Result:"White"}},{multi: true})
db.partidas.update({Result:"0-1"},{$set:{Result:"Black"}},{multi: true})
db.partidas.update({Result:"1/2-1/2"},{$set:{Result:"Draw"}},{multi: true})
```

Figura 31. Cambio de valores en el campo Result

Después de esto ya se tiene el campo *Result* modificado para el uso como filtro en el desarrollo de las visualizaciones.

Otra cosa a tener cuenta, es que los nombres de los jugadores seleccionados pueden variar de una partida a otra ligeramente, esto es, puede aparecer Magnus Carlsen como Carlsen,M o Carlsen, Magnus y una larga lista de variantes sobre el apellido seguido de una coma y el nombre codificado por inicial, completo o incluso las tres primeras letras.

Para solucionar este problema, que no se consideren valores diferentes y todos tengan el nombre adecuado, se procede a una búsqueda con comodines para encontrar todas las variantes juntas y cambiarlas por el nombre completo del jugador.

```

db.partidas.update({"White": {$regex:".*Carlsen"}},{$set:{White:"Magnus Carlsen"}},{multi: true})
db.partidas.update({"White": {$regex:".*Caruana"}},{$set:{White:" Fabiano Caruana"}},{multi: true})
db.partidas.update({"White": {$regex:".*Mamedya"}},{$set:{White:"Shakhriyar Mamedyarov"}},{multi: true})
db.partidas.update({"White": {$regex:".*Giri"}},{$set:{White:"Anish Giri"}},{multi: true})
db.partidas.update({"White": {$regex:".*Aronian"}},{$set:{White:"Levon Aronian"}},{multi: true})
db.partidas.update({"White": {$regex:".*Vachier"}},{$set:{White:"Maxime Vachier-Lagrave"}},{multi: true})
db.partidas.update({"White": {$regex:".*Kramnik"}},{$set:{White:"Vladimir Kramnik"}},{multi: true})
db.partidas.update({"White": {$regex:".*So"}},{$set:{White:"Wesley So"}},{multi: true})
db.partidas.update({"White": {$regex:".*Anand"}},{$set:{White:"Viswanathan Anand"}},{multi: true})
db.partidas.update({"White": {$regex:".*Nakamura"}},{$set:{White:"Hikaru Nakamura"}},{multi: true})
db.partidas.update({"White": {$regex:".*Karjakin"}},{$set:{White:"Sergey Karjakin"}},{multi: true})
db.partidas.update({"White": {$regex:".*Vallejo"}},{$set:{White:"Francisco Vallejo Pons"}},{multi: true})

db.partidas.update({"Black": {$regex:".*Carlsen"}},{$set:{Black:"Magnus Carlsen"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Caruana"}},{$set:{Black:" Fabiano Caruana"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Mamedya"}},{$set:{Black:"Shakhriyar Mamedyarov"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Giri"}},{$set:{Black:"Anish Giri"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Aronian"}},{$set:{Black:"Levon Aronian"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Vachier"}},{$set:{Black:"Maxime Vachier-Lagrave"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Kramnik"}},{$set:{Black:"Vladimir Kramnik"}},{multi: true})
db.partidas.update({"Black": {$regex:".*So"}},{$set:{Black:"Wesley So"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Anand"}},{$set:{Black:"Viswanathan Anand"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Nakamura"}},{$set:{Black:"Hikaru Nakamura"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Karjakin"}},{$set:{Black:"Sergey Karjakin"}},{multi: true})
db.partidas.update({"Black": {$regex:".*Vallejo"}},{$set:{Black:"Francisco Vallejo Pons"}},{multi: true})

```

Figura 32. Actualización de los nombres de los jugadores

Se realiza todo esto utilizando el comando *update* con una particularidad en la búsqueda que se llama *\$regex* (ver Figura 32) para usar comodines en los valores de los campos Black y White, que es el campo en el que están los nombres de los jugadores.

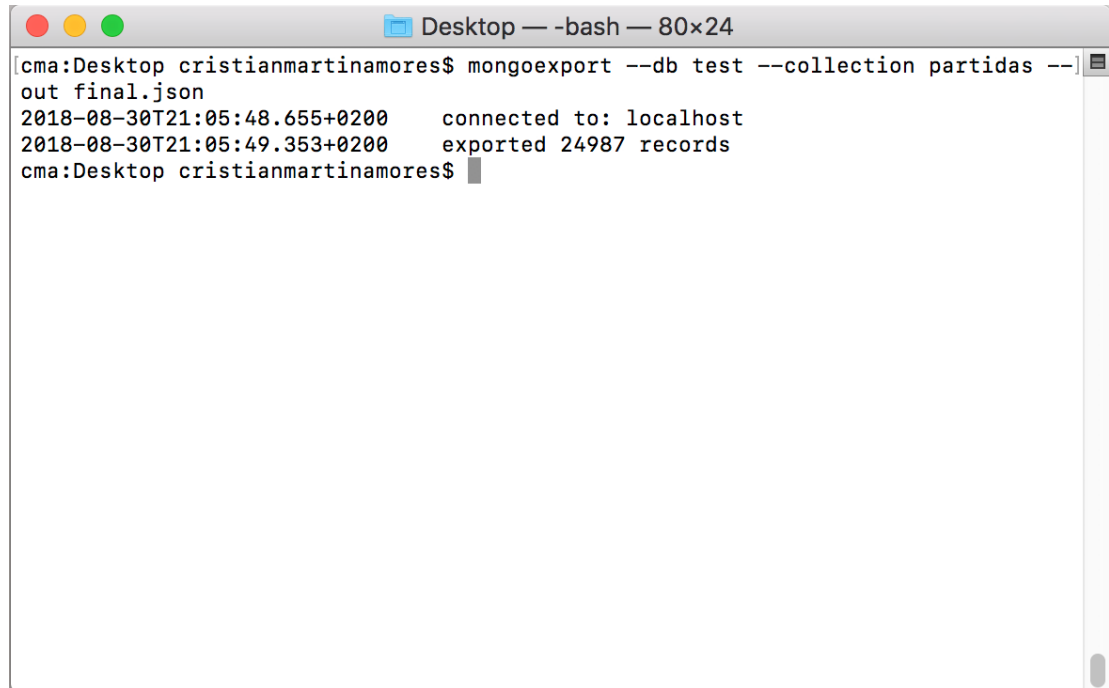
Se realiza la búsqueda con un asterisco delante del nombre para que así cualquier campo que empiece de la misma forma será reemplazado por el que aparece en el operador *\$set*.

Extracción de la información a un fichero JSON

Una vez concluida la parte anterior del procesamiento de los datos, ya se tiene completamente procesada toda la información con los campos necesarios y la información que hay en ellos corregida para que así no haya conflictos a la hora de desarrollar las visualizaciones y utilizar los filtros.

El siguiente paso es la extracción de la información obtenida después del procesamiento a un fichero JSON para que sea importado en Tableau para realizar el desarrollo de las visualizaciones. Para realizar esta parte es necesario utilizar *mongoexport*, que es una herramienta que proporciona MongoDB para extraer a un fichero JSON los documentos que se tienen en la base de datos. Desde la terminal se tiene que elegir la ruta en la que se quiere extraer la información y ejecutar el comando *mongoexport* (ver Figura 33) seguido de las opciones de *--db* para seleccionar el nombre de la base de datos, en este caso "test". Después

hay que identificar el nombre de la colección con la opción `--collection` eligiendo la colección “partidas” y por último el uso de la opción `--out` para elegir el nombre del fichero de salida.



```

cma:Desktop cristianmartinamores$ mongoexport --db test --collection partidas --out final.json
2018-08-30T21:05:48.655+0200    connected to: localhost
2018-08-30T21:05:49.353+0200    exported 24987 records
cma:Desktop cristianmartinamores$
  
```

Figura 33.Extracción de la información a un fichero JSON

Con este último paso ya se obtiene el fichero *final.JSON* que será utilizado en el siguiente punto para empezar a realizar el desarrollo de las visualizaciones.

```

1  ▾ [{
2  ▾   "_id": {
3    "$oid": "5b892c566494b110acc9046a"
4    },
5    "Black": "Wolff, Patrick G",
6    "ECO": "B09",
7    "Event": "Wch U20",
8    "Result": "Black",
9    "Site": "Kiljava",
10   "White": "Viswanathan Anand",
11   "Moves": "1.e4 d6 2.d4 Nf6 3.Nc3 g6 4.f4 Bg7 5.Nf3 O-O 6.e5 Nfd7 7.h4 c5 8.e6 fxe6 9.h5 gxh5 10.Rxh5
    Nf6 11.Rh4 Nc6 12.Be3 Qa5 13.Bd3 cxd4 14.Nxd4 Nxd4 15.Bxd4 e5 16.fxe5 dxe5 17.Be3 e4 18.Bc4+ Kh8 19.Qd4
    Bg4 20.Bb3 Rac8 21.Bd2 Qa6 22.Qe3 Nd5"
12  },
13  ▾ {
14  ▾   "_id": {
15    "$oid": "5b892c566494b110acc9046b"
16    },
17    "Black": "Viswanathan Anand",
18    "ECO": "B66",
19    "Event": "Lloyds Bank op",
20    "Result": "White",
21    "Site": "London",
22    "White": "Chandler, Murray G",
23    "Moves": "1.e4 c5 2.Nf3 Nc6 3.d4 cxd4 4.Nxd4 Nf6 5.Nc3 d6 6.Bg5 e6 7.Qd2 a6 8.0-0 h6 9.Be3 Be7 10.Be2
    Bd7 11.f4 b5 12.Bf3 Rc8 13.Nb3 Na5 14.Nxa5 Qxa5 15.Kb1 b4 16.Ne2 Bc6 17.Ng3 O-O 18.Bd4 d5 19.e5 Ne4
    20.Bxe4 dxe4 21.Qe2 Bd5 22.b3 f5 23.exf6 Bxf6 24.Bxf6 Rxf6 25.Rhf1 Rcf8 26.Qe3 Qc7 27.f5 Qb7 28.fxe6
    Rxf1 29.Nxf1 Bxe6 30.Ng3 Bd5 31.Qc5 Bf7 32.Nf5 Qb5 33.Qxb5 axb5 34.Rd4 Re8 35.Ne3 Bg6 36.Rxb4 Rf8
    37.Rxb5 Rf2 38.Rc5"
24  },
25  ]
  
```

Figura 34.Ejemplo del fichero final.JSON

5. Desarrollo de las visualizaciones

Para empezar el desarrollo de las visualizaciones, se tiene que seguir una estructura definida por la importación de los datos, la creación de las visualizaciones en ventanas simples, esto es, crear las visualizaciones individualmente en cada página para luego crear el cuadro de mando en el que aparezcan simultáneamente varias de ellas.

5.1. Importación de los datos

Para importar los datos que contiene el fichero *final.JSON* primero hay que abrir la aplicación del Tableau Desktop. Una vez abierto aparece la ventana principal con un menú a la izquierda para hacer la conexión a un archivo o a un servidor. En este caso es desde un archivo. Hay que seleccionar el botón *JSON file* (ver *Figura 35*) y se abrirá el explorador de archivos en el que hay que buscar el archivo *final.JSON* generado desde *mongoexport* para conectarse a él.

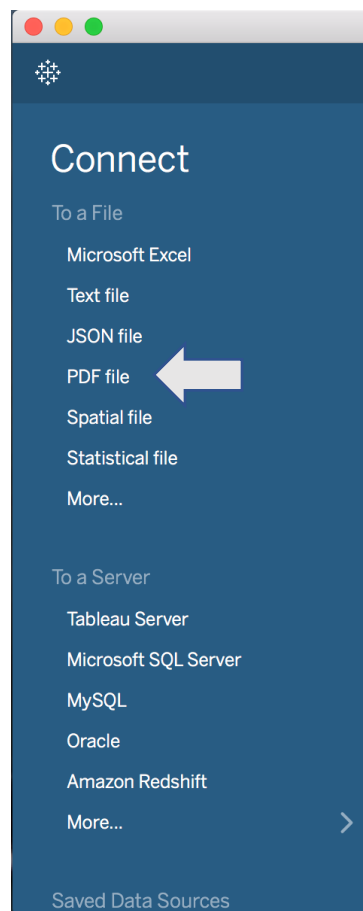


Figura 35. Menú de conexión de Tableau Desktop

Una vez se ha seleccionado el archivo que se va a introducir y se acepta, aparece un menú con las opciones de los campos que se quieren importar a la fuente de datos de la herramienta para su posterior uso en las visualizaciones. Aparece el esquema de los campos y un ejemplo de información para cada campo (ver *Figura 36*). El campo `_id` aparece separado y da la opción de añadirlo. Es un campo obligatorio para MongoDB por ese motivo no se eliminó, pero para las visualizaciones no se necesitará, por lo tanto, no se selecciona ese campo y se comprueba que estén los siete campos que se van a utilizar. En el caso de Tableau aparecen en orden alfabético: Black, ECO, Event, Moves, Result, Site y White.

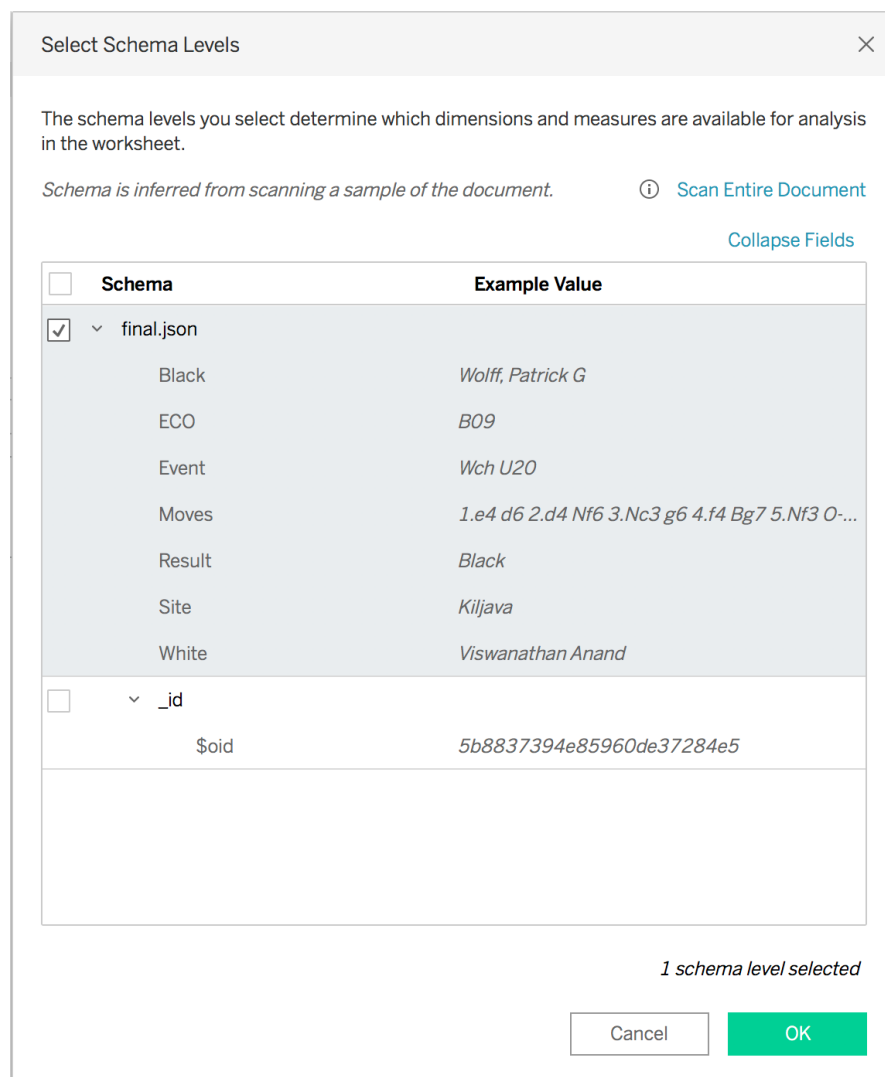


Figura 36. Esquema de los campos a importar en Tableau

Una vez se acepta el esquema aparecerán creados los campos en formato de tabla en la que se puede comprobar que todo haya salido correctamente (ver *Figura 37*).

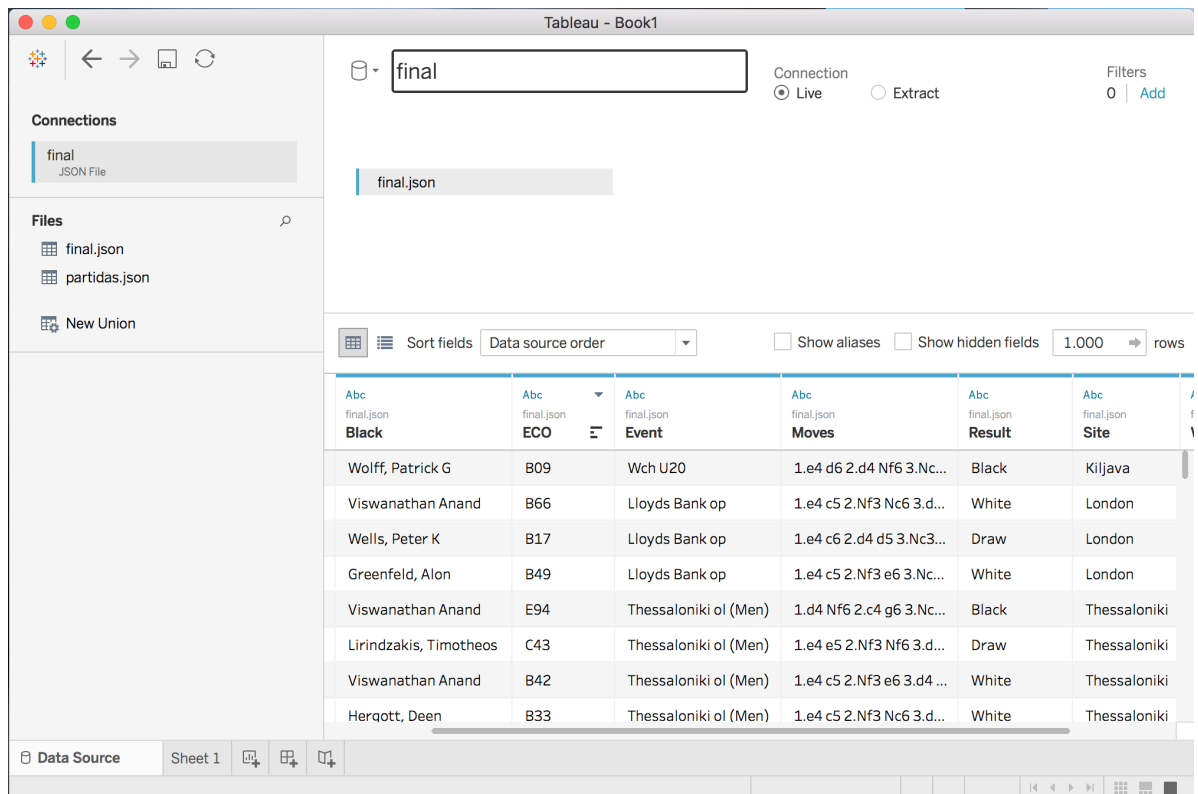


Tableau - Book1

Connections: final (JSON File)

Files: final.json, partidas.json, New Union

Sort fields: Data source order

Show aliases: ☐ Show hidden fields: ☐ 1.000 rows

Player	ECO	Event	Moves	Result	Site
Wolff, Patrick G	B09	Wch U20	1.e4 d6 2.d4 Nf6 3.Nc...	Black	Kiljava
Viswanathan Anand	B66	Lloyds Bank op	1.e4 c5 2.Nf3 Nc6 3.d...	White	London
Wells, Peter K	B17	Lloyds Bank op	1.e4 c6 2.d4 d5 3.Nc...	Draw	London
Greenfeld, Alon	B49	Lloyds Bank op	1.e4 c5 2.Nf3 e6 3.Nc...	White	London
Viswanathan Anand	E94	Thessaloniki ol (Men)	1.d4 Nf6 2.c4 g6 3.Nc...	Black	Thessaloniki
Lirindzakis, Timotheos	C43	Thessaloniki ol (Men)	1.e4 e5 2.Nf3 Nf6 3.d...	Draw	Thessaloniki
Viswanathan Anand	B42	Thessaloniki ol (Men)	1.e4 c5 2.Nf3 e6 3.d...	White	Thessaloniki
Herqott, Deen	B33	Thessaloniki ol (Men)	1.e4 c5 2.Nf3 Nc6 3.d...	White	Thessaloniki

Figura 37. Tabla de datos importados en Tableau

Una vez comprobados todos los campos, se puede ver que todo son dimensiones ya que no hay valores numéricos en la información de los campos. Si los hubiera tendrían que llamarse medidas.

5.2. Creación de las visualizaciones individuales

Se procede ahora a crear las visualizaciones individualmente para luego juntarlas en el cuadro de mando o *dashboard*.

Se accede a *Sheet 1* que es la primera página en la que se va a crear la primera visualización. Una vez dentro, aparecen los campos a la izquierda en el apartado de *Dimensions*. Estos valores serán los que se arrastrarán a las distintas opciones que ofrece Tableau para ir realizando las visualizaciones.

Se empezará por una visualización que muestre de manera general el porcentaje de victorias de las blancas (White), de las negras (Black) y de tablas (Draw). Para ello se arrastra el campo *Result* desde la columna de la izquierda al campo de las columnas (Columns) que aparece en la parte superior central de la aplicación.

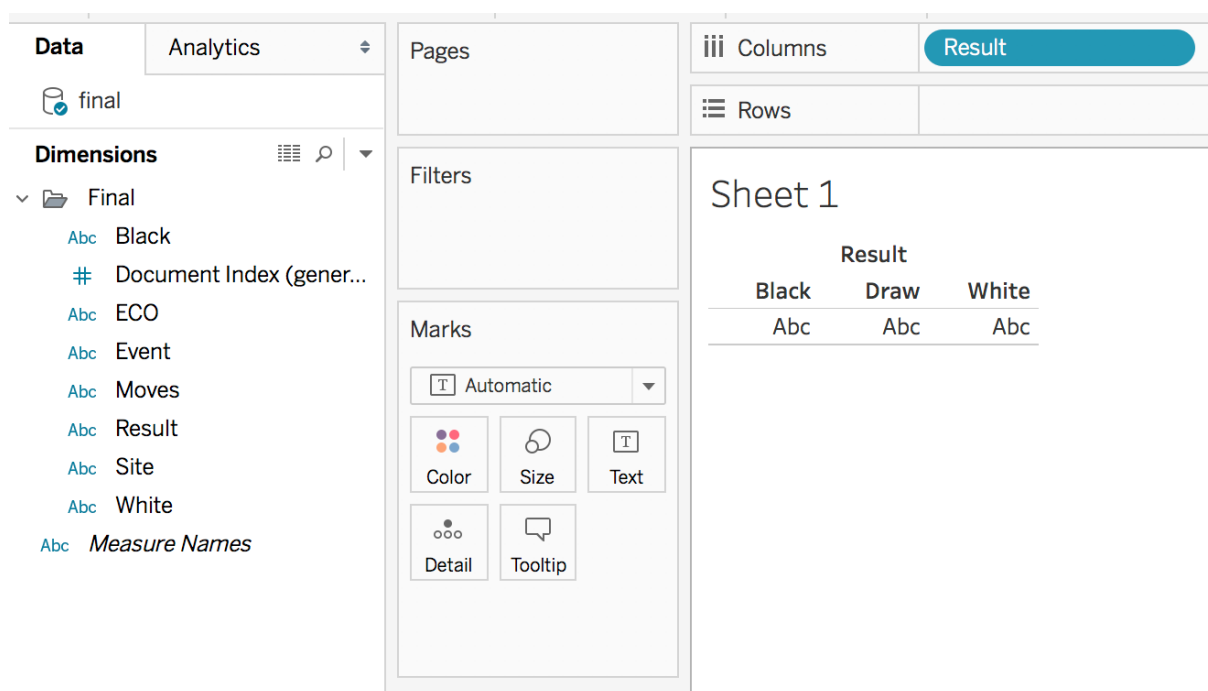


Figura 38. Gráfica general, primer paso

Aparece en el centro el nombre por columnas de las distintas variables que aparecen en el campo *Result* que son en este caso: Black, Draw y White. Como lo que se quiere ver es cuántos de los registros que hay en la fuente de datos tienen Black, Draw o White en el campo *Result* hay que arrastrar nuevamente dicho campo a las filas (Rows) y ahora aparecerá una tabla cruzada de datos.

Sheet 1

	Result		
Result	Black	Draw	White
Black	Abc		
Draw		Abc	
White			Abc

Figura 39. Gráfica general, segundo paso

Esta tabla que aparece no tiene ninguna información porque se están cruzando dos dimensiones. Para que aparezca la medida del número de veces que aparece cada variable hay que cambiar el tipo de dato del campo *Result* de las filas. Para esto, se selecciona en la flecha que aparece a la derecha del campo *Result* que hay en las filas y en el desplegable se selecciona medidas (Measures) y dentro de ellas a *Count* para que así cuente las veces que aparece cada variable.

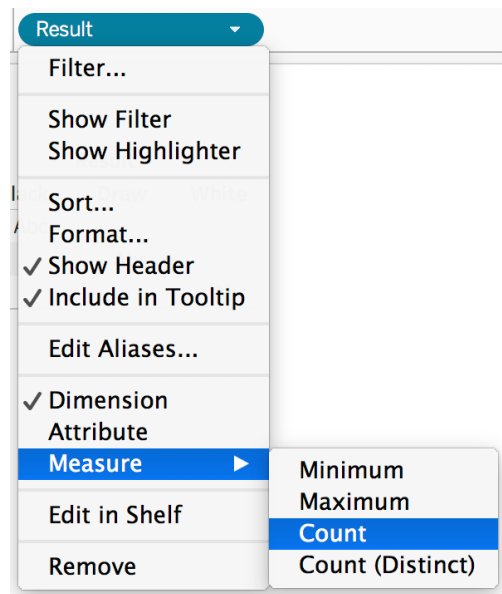


Figura 40. Gráfica general, tercer paso

Ahora ya aparece un gráfico de barras simple, ordenado por defecto por orden alfabético y con el título de la página *Sheet 1*. También los nombres de los ejes son por defecto el nombre de los campos utilizados.

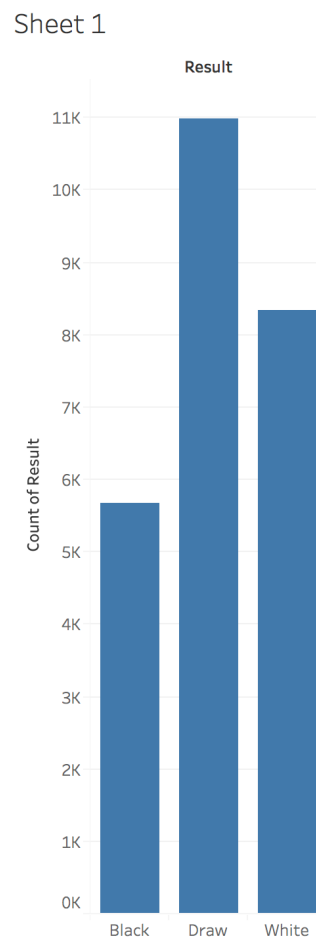


Figura 41. Gráfica general simple

Para cambiar el nombre del título basta con hacer doble clic en él y aparecerá una ventana con el cambio de nombre. El eje de las abscisas o columnas tiene el nombre del campo que se está utilizando y se puede ocultar si se desea haciendo clic derecho sobre él y pulsando en *Hide Field*. Para cambiar el eje de las ordenadas basta con hacer clic derecho en el nombre *Count of Result* y elegir la opción de *Edit Axis*, aparecerá una ventana en la que se puede cambiar el campo *Title* y ahí se elige el nombre que se desee. Lo mismo para las variables Black, Draw y White de las columnas eligiendo *Edit Alias*.

El orden lógico en el ajedrez para contabilizar los resultados de las partidas es: Blancas, Tablas y Negras. Por lo tanto, hay que cambiar el orden que aparece por defecto al establecido. Ahora hay que seleccionar la flecha del campo *Result* que aparece en el campo columnas y elegir la opción para ordenar *Sort*, una vez seleccionada aparece una ventana (ver *Figura 42*) en la que se elige la opción manual y se coloca el orden que se quiera, en este caso: White, Draw y Black, y se acepta en OK. Después de esto aparecerán las barras en dicho orden.

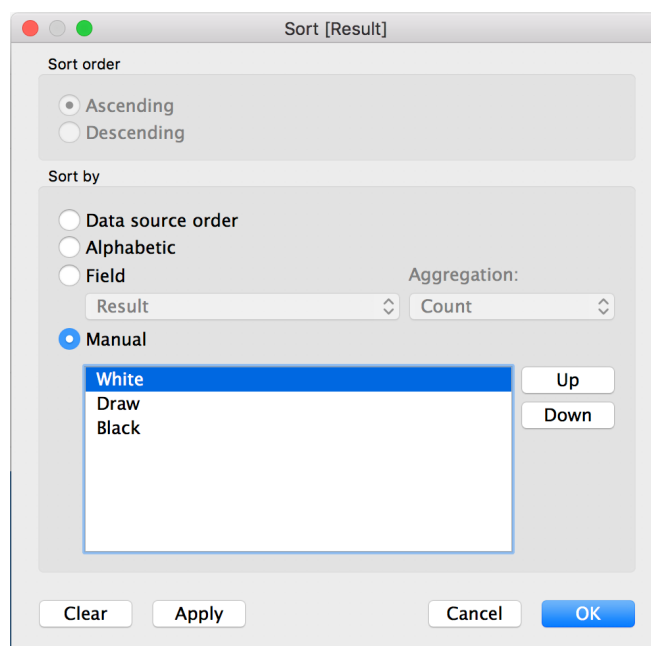


Figura 42. Ordenado de columnas

Otro cambio que se quiere realizar es el del color de las barras. Se utilizarán los colores típicos del ajedrez para el color de éstas.

Arrastrando nuevamente el campo *Result*, pero esta vez al cuadrado en el que pone *Color* utilizará un matiz distinto para cada valor encontrado en *Result*, como son tres los valores, pondrá cada columna de un color. Ahora hay que cambiar el color que aparece por defecto

por el que se vaya a elegir, pulsando en *Color* → *Edit Colors...* En este caso: para White se elegirá gris claro, para Draw, gris oscuro y negro para Black.

Como lo que interesa ver realmente es cuánto porcentaje del total obtiene cada resultado, en vez del número total de partidas, se da la orden a Tableau de que calcule el porcentaje del total. Para esto, se ce clic en la flecha de la que se habló anteriormente para el *Count*, y esta vez se selecciona *Quick Table Calculation* → *Percent of Total* y recalculará el conteo de cada resultado dividido por el total de partidas en la base de datos.

También se seleccionan las tres barras con clic derecho y en la opción de *Mark Label* se elige *Always Show*, para que aparezca el porcentaje encima de cada barra y así eliminar la cabecera de las filas.

Se cambia el nombre de *Sheet 1* por *Results* para que aparezca este título en el gráfico.

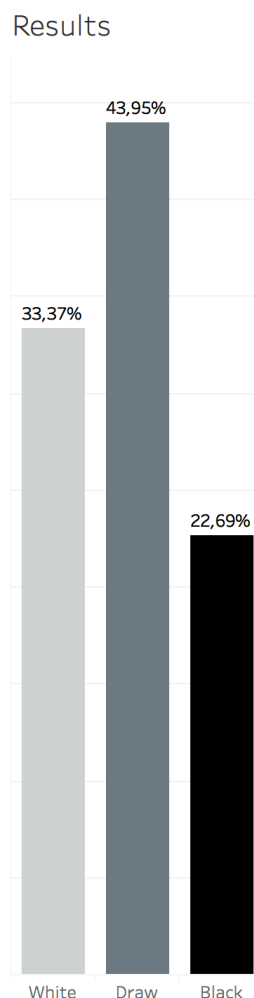


Figura 43. Gráfica general completa

Por último, hay que modificar el *Tooltip*, que es la ventana que aparece cuando se pasa el puntero del ratón por encima y que nos muestra información.

Se accede a él en el menú que hay a la izquierda del gráfico. Haciendo clic en la palabra *Tooltip* aparece una ventana para modificarlo. En la ventana se cambia el mensaje “% of Total Count of Result along Table (Across)” por “Percent of Total” ya que ese mensaje es el que aparece por defecto al hacer el cálculo del porcentaje total.

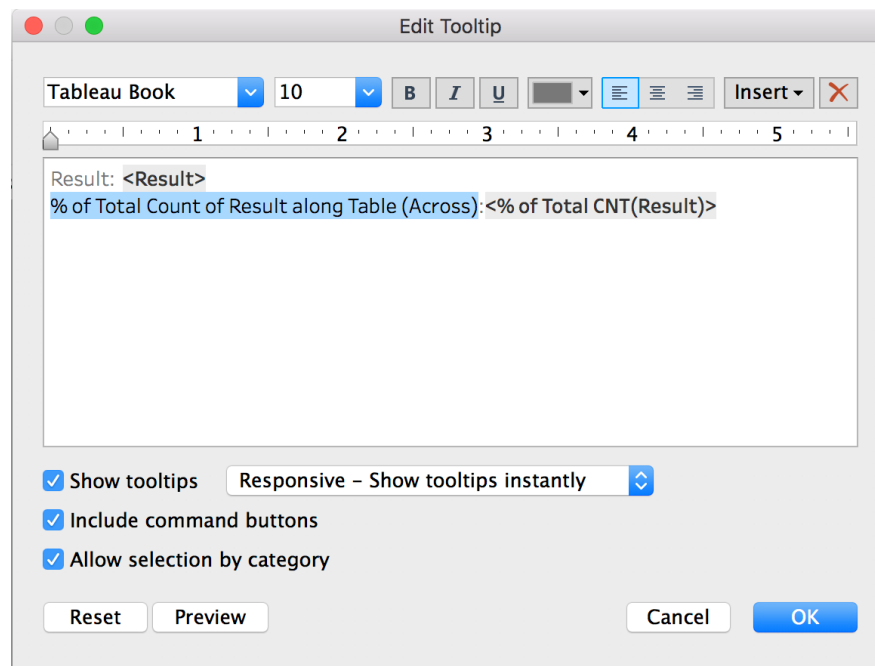


Figura 44. Modificación de Tooltip

Ahora el mensaje que aparece cuando se pasa el ratón por encima de cualquier barra aparecerá de la siguiente forma: (ver Figura 45).

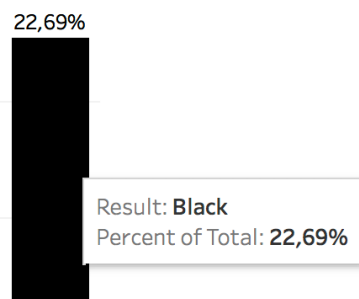


Figura 45. Tooltip modificado

Para la siguiente visualización se quiere representar el número de veces que se ha jugado cada apertura. Para ello, primero se crea una página nueva y se coloca en las columnas y en

las filas el campo *ECO*. En las filas, como en el paso anterior, se cambia la dimensión por medida y se elige también *Count*.

También se van a ordenar las columnas por orden descendente, es decir, a la izquierda del todo aparecerá la apertura más jugada mientras que al extremo contrario aparecerá la menos utilizada. Para esto, se hace clic en la flecha del campo *ECO* de las columnas y se selecciona *Sort*. En la parte de *Sort order* (ver *Figura 46*) se elige la opción *Descending* y en *Sort by* se elige *ECO* en *Field* y *Aggregation* por *Count*.

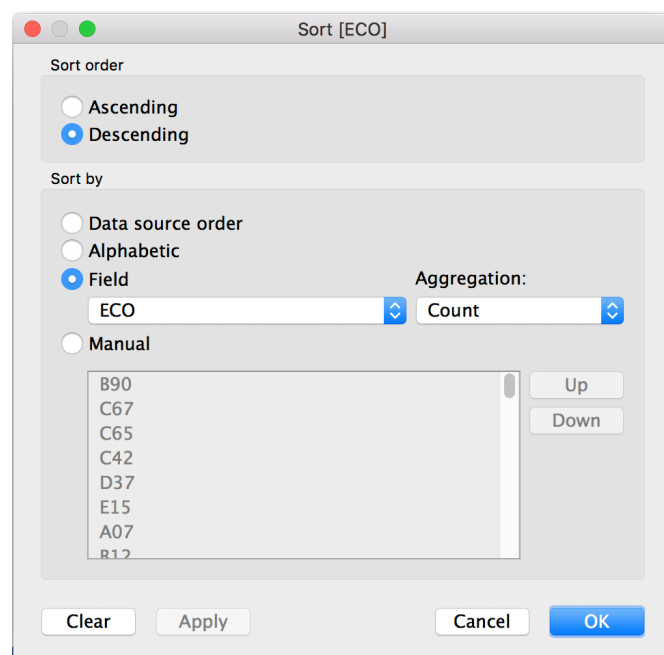


Figura 46. Ordenación de aperturas

En el campo columnas aparecerá un icono propio de la ordenación descendente que muestra que dicho campo ha sufrido algún tipo de modificación en la ordenación.

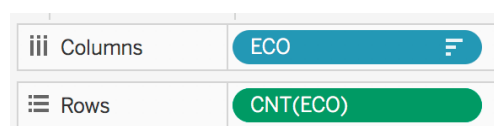


Figura 47. Columnas y filas para contabilizar aperturas

Aparecerá un gráfico que se conoce como larga cola. El primer 20% de los datos contiene el 80% del área y el 80% de los datos restantes suman el 20% del área que falta. En el caso de las aperturas, muestra que el 20% de éstas, aparecen aproximadamente en el 80% de las partidas jugadas. Este gráfico es muy relevante para el estudio ya que está afirmando que hay muchos tipos de aperturas que son marginales y que se utilizan en pocas ocasiones.

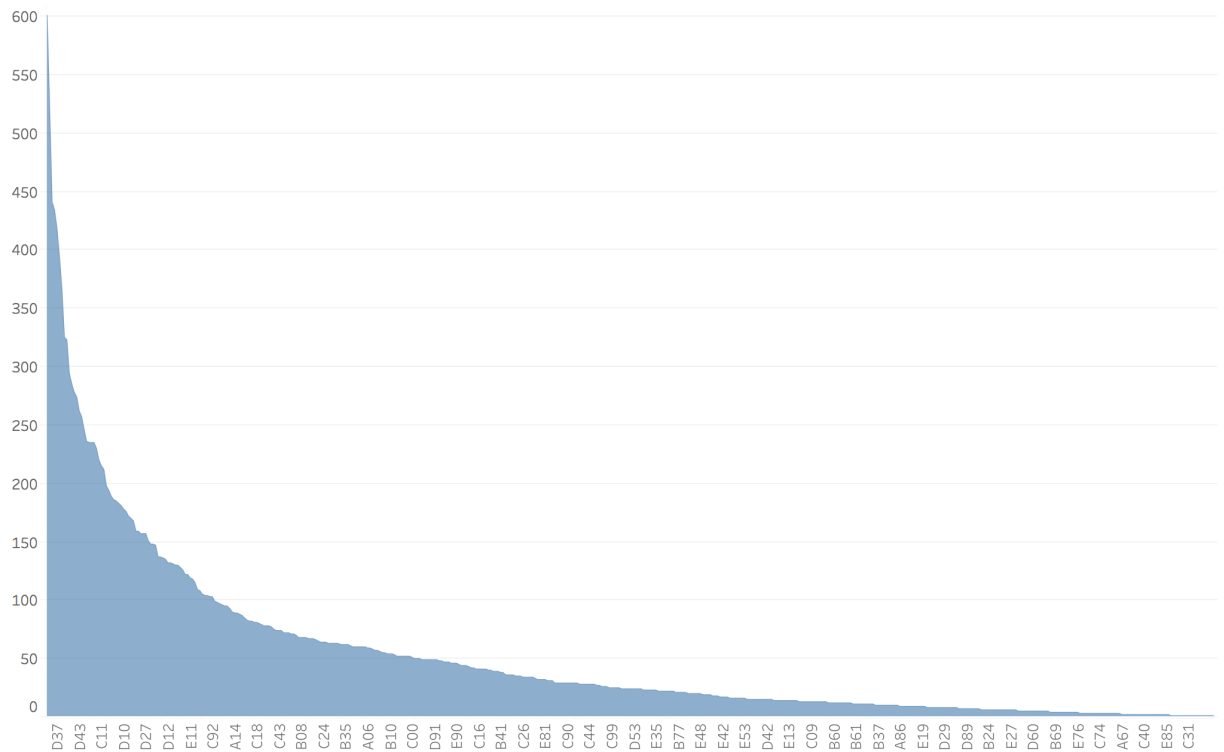


Figura 48. Gráfico de larga cola para las aperturas

Se va a cambiar el gráfico de larga cola por un gráfico de dispersión con formas. Para realizar esta parte bastará con seleccionar el desplegable que aparece a la izquierda del gráfico llamado *Marks* y elegir la opción *Shape*.

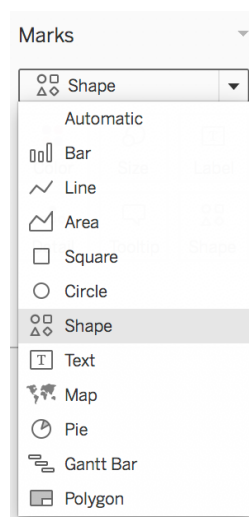


Figura 49. Desplegable Marks

Después de realizar este cambio, se obtiene un gráfico de dispersión ordenado de mayor a menor con un punto por cada apertura.

Most played openings

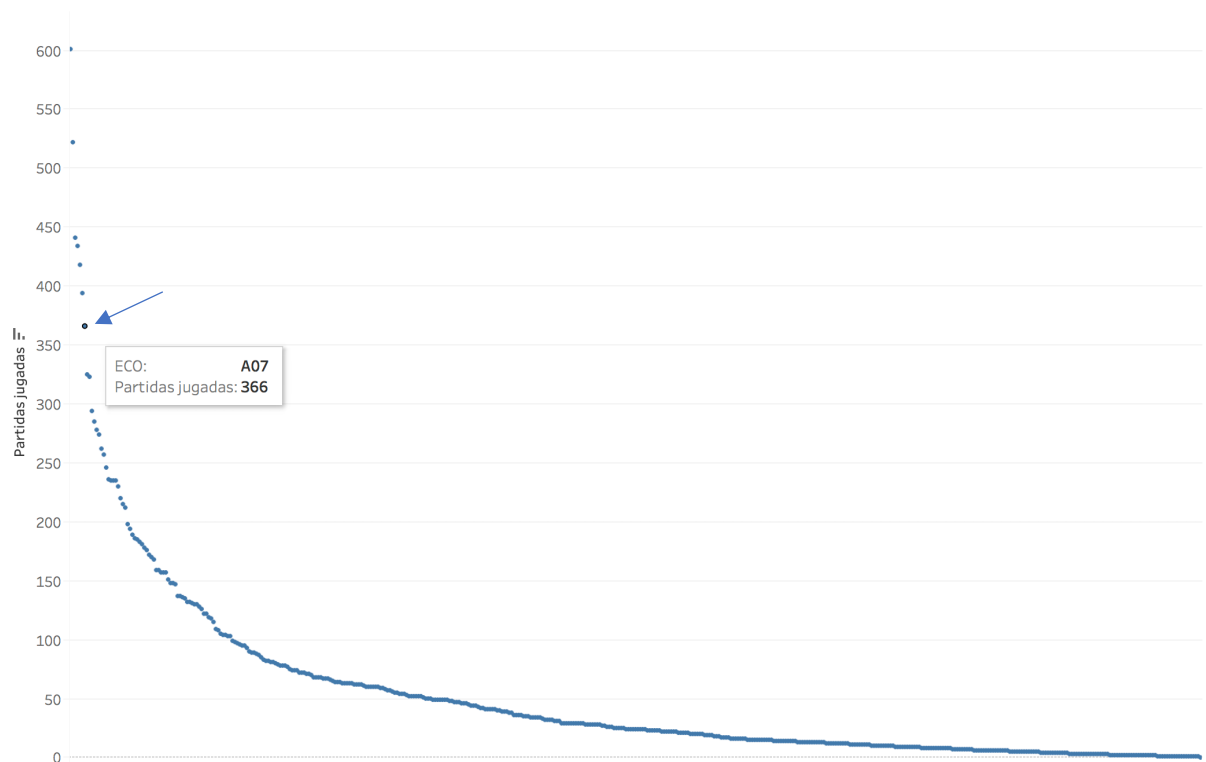


Figura 50. Gráfico de aperturas más jugadas

Pasando el puntero del ratón por encima de cada punto aparece el nombre de la apertura y las partidas jugadas con ella.

El siguiente gráfico que se va a realizar, es un gráfico de burbujas empaquetadas para poder analizar el éxito de cada apertura con blancas, negras y tablas.

El uso de un gráfico de burbujas empaquetadas sirve para identificar rápidamente, por el tamaño, qué aperturas son las más utilizadas en cada resultado: ganan blancas, negras o se hacen tablas. Todas ellas, representadas por el porcentaje de éxito en cada uno de los tres campos. Se codificará por tamaño el número de veces que se utiliza, y por el color el porcentaje de éxito que tiene.

Para realizar este gráfico se crea una nueva página y se coloca *Result* en las columnas para crear las divisiones por los resultados. Como se hizo en la primera gráfica, se ordenan las columnas de forma manual para que aparezcan en el orden: White, Draw y Black. Se añade el campo *Result* a *Color*, *Size* y *Tooltip*, y el campo *ECO* a *Label* para que aparezca el código de la apertura en la burbuja.

En el desplegable de *Marks*, se selecciona *Circle* para que se creen los círculos que formarán el gráfico de burbujas empaquetadas.

Una vez ya están todos los campos colocados, se empieza modificando el campo *Result* que se ha puesto en *Size* para modificar el tamaño de las burbujas. Haciendo clic derecho se cambia el campo de dimensión a medida eligiendo *Count*.



Figura 51. Gráfico de burbujas empaquetadas, primer paso

Ya está codificado el tamaño de las burbujas respecto del número de partidas que ha ganado el blanco, el negro o se han hecho tablas con cada apertura.

El siguiente paso consiste en codificar el color de las burbujas por el porcentaje de éxito que tiene cada una. Para ello, al igual que con el campo *Result* de *Size*, hay que cambiar el de *Color* de dimensión a medida eligiendo *Count*. Una vez hecho esto, hay que realizar un cálculo de porcentaje seleccionando *Percent of Total*, pero esta vez, en la opción de *Compute Using*, hay que elegir *Result* para que el porcentaje no sea calculado sobre el total de partidas existentes y solamente sea utilizado el número de partidas que tienen la misma apertura. Es decir, el total del porcentaje de White, Draw y Black tiene que sumar el 100%, ya que se computa utilizando los tres posibles resultados.

Cuando ya se tienen las burbujas codificadas por color, se pulsa en *Color* y se edita para elegir el gradiente deseado.

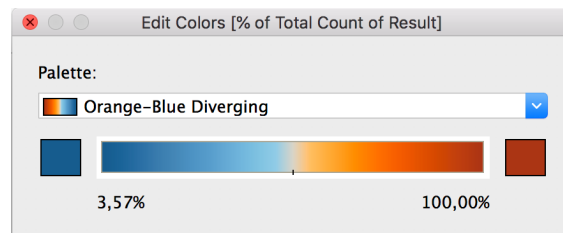


Figura 52. Gradiente de color seleccionado

En este caso se elige que el azul oscuro marque el peor porcentaje de éxito y que el rojo oscuro sea el 100%.

Ahora queda modificar el último campo *Result* que se encuentra en *Tooltip*, cambiándolo de dimensión a medida con *Count* para que cuando se coloque el ratón encima de una burbuja aparezca la información del código de la apertura, el resultado obtenido, el número de veces que se ha jugado con ese resultado y el porcentaje que representa de éxito.

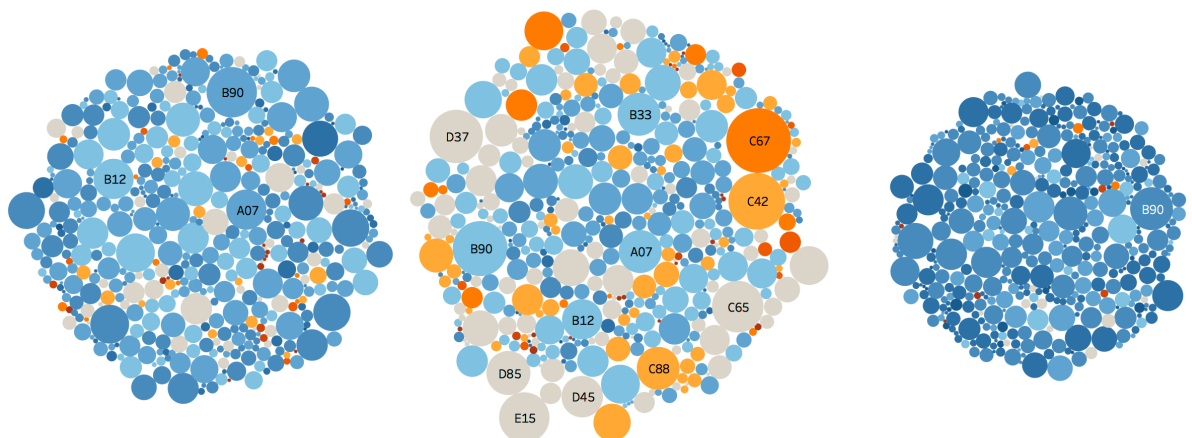


Figura 53. Gráfico de burbujas empaquetadas

Con este gráfico, se puede ver fácilmente qué apertura es la más jugada con cada resultado, pero eso no quiere decir que sea la más efectiva. El propio usuario toma la decisión de con qué tamaño quedarse y elegir la que más le convenga por el color. Hay muchas burbujas que son muy pequeñas y de color rojo, lo que significa que quizás solo exista una partida y por lo tanto tendrá el 100% de éxito en su resultado, esto no da información sobre la efectividad de la apertura ya que es una muestra escasa.

5.3. Creación del cuadro de mando

El siguiente paso de las visualizaciones es juntarlas en un cuadro de mando o *dashboard* para el uso de filtros interactivos y conseguir sacar más información de que la se muestra en general.

Para la elaboración del cuadro de mando, se utilizará el botón de *New Dashboard*. Una vez creado, se elegirá el tamaño automático en el menú de arriba a la izquierda y se arrastrarán las páginas en las que se han creado las visualizaciones individuales al centro de la herramienta. Para poder colocarlas libremente, se seleccionará la flecha que aparece en cada visualización y se optará la opción *Floating* para que así sea flotante.

Se añaden dos filtros: uno para el jugador de blancas y otro para el jugador de negras. Este filtro afecta a todas las gráficas mostrando solamente las partidas en las que dicho jugador juega con el color elegido. También se puede elegir un jugador en cada filtro y aparecerían las partidas en las que han jugado el uno contra el otro con la disposición dada en el filtro, es decir, el jugador A con blancas y el jugador B con negras, mostraría todas las partidas que hayan jugado juntos siendo A el jugador de blancas y B el de negras, pero no al contrario. De esta forma, se pueden analizar los resultados que tiene cierto jugador con blancas, negras o haciendo tablas.

La gráfica de *Results*, si no hay ningún filtro activo, muestra el porcentaje de éxito de cada resultado para el total de partidas. Cuando se filtra, por ejemplo, por un jugador de blancas, lo que se verá en dicha gráfica será en White el porcentaje de sus victorias, en Draw las tablas que hace cuando juega con blancas y en Black el porcentaje de partidas que ha perdido, ya que él está jugando con blancas en todas las partidas.

Se crea una acción de destacado (*Highlight*), esto es, una acción que cuando se haga clic en algún gráfico a elección deseada, se destaque lo que se pretende mostrar. Para crear la acción, se accede en el menú de herramientas de Tableau a *Dashboard* y se despliega un menú en el que se selecciona *Actions...* En la ventana que aparece se selecciona *Add Action* > y se elige *Highlight...*

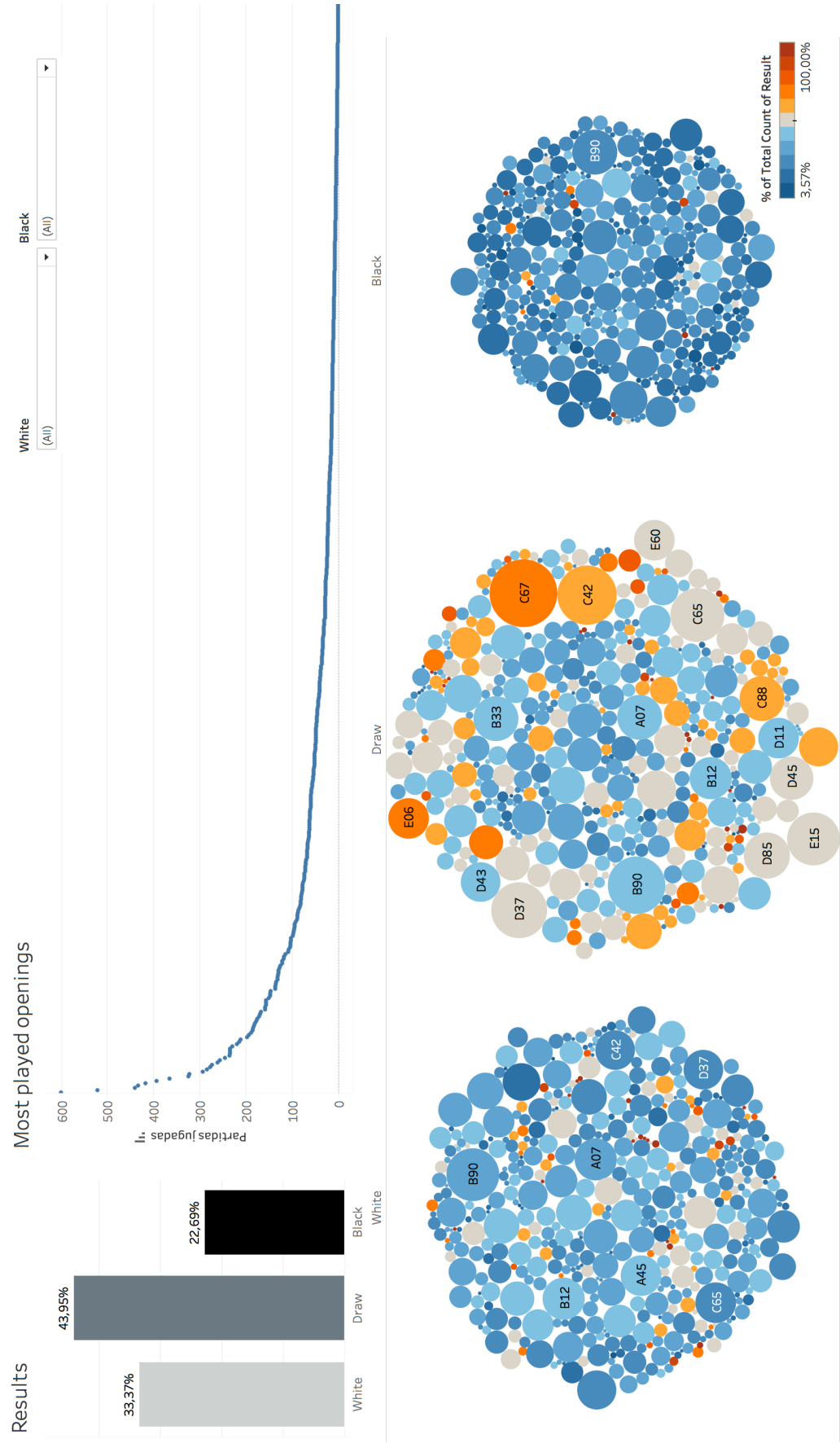


Figura 54. Cuadro de mando

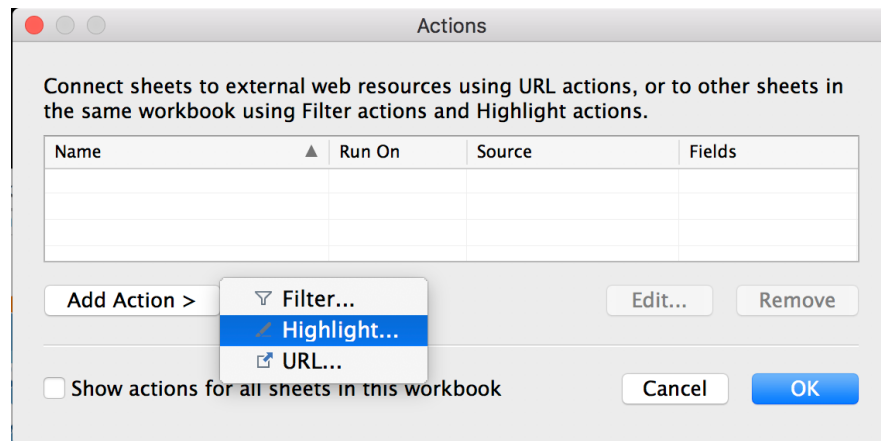


Figura 55. Ventana de Actions...

Una vez dentro, hay que seleccionar en *Source Sheets* las páginas que se quiere que realicen la acción: *Bubble Graph* y *Most played openings*. En *Target Sheets* se eligen las páginas de destino que serán las mismas que en el caso anterior. En *Run action on* se escoge *Select*. Y para terminar, en *Highlighting* se selecciona *Selected Fields* → *ECO* para destacar la apertura seleccionada en las burbujas y en la gráfica de dispersión. De este modo, se encontrarían de una manera sencilla para hacer los análisis deseados.

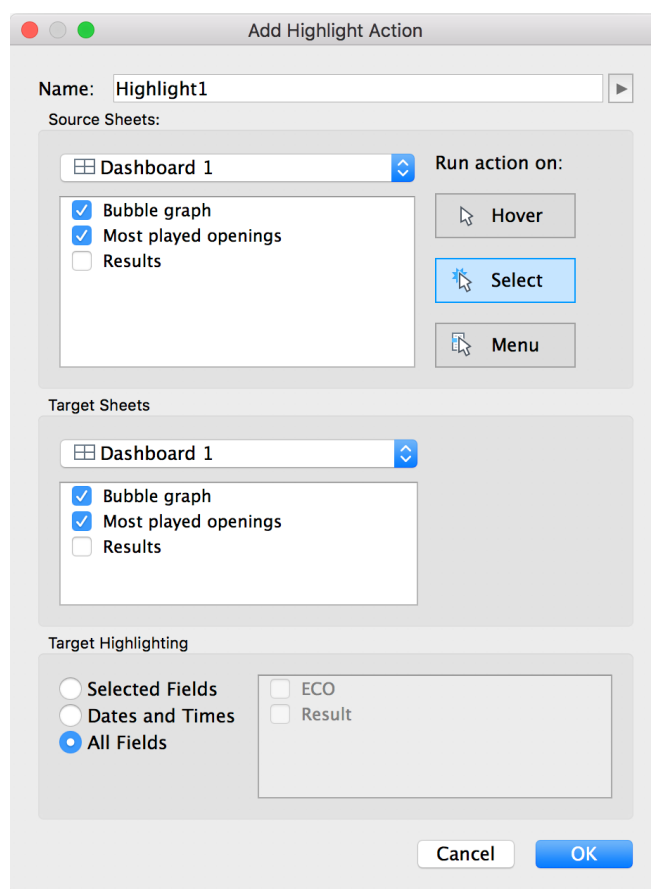


Figura 56. Highlight action

Ahora cuando se hace clic sobre una burbuja, se destacan las burbujas de los otros resultados y así se pueden encontrar fácilmente y comparar los porcentajes de éxito.

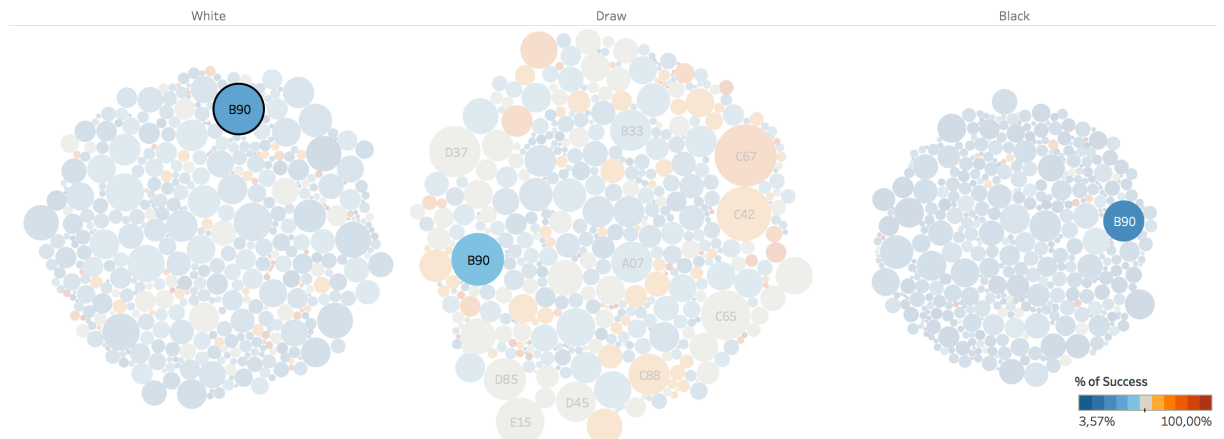


Figura 57. Ejemplo de Highlight con apertura B90

Accediendo, en el menú que aparece al hacer clic en una burbuja, a *View Data...* se puede obtener toda la información sobre todas las partidas que tienen dicha apertura. Esto es muy útil para utilizar el campo *Moves* y copiarlo a una web o aplicación de ajedrez para reproducir la partida en un tablero y así poder estudiar las partidas que han empleado la apertura seleccionada en la burbuja. También se puede obtener información del nombre del evento en el que se ha jugado dicha apertura y la ciudad.

5.4. Creación del grafo

En esta parte, se va a realizar el grafo entre los doce jugadores de ajedrez seleccionados para ver cuántas veces se enfrentan entre ellos en los tableros. Esto dará una idea de si los jugadores suelen competir en los mismos torneos y ver cuál es más activo con los demás.

Un grafo está compuesto por vértices y arcos. Los vértices son los objetos que están unidos por los arcos para representar las relaciones que existen entre ellos. En este caso, el arco muestra el número total de partidas jugadas entre los jugadores unidos por él.

Para realizar el grafo se utiliza MicroStrategy Desktop. Se abre la aplicación y se crea un nuevo *Dossier*. En el apartado de *Datasets* hay que seleccionar *New Data* → *Files from Disk* y seleccionar el archivo JSON en el que están almacenadas las partidas.

Una vez hecho esto ya se tiene cargada toda la información para empezar a realizar el grafo. Se selecciona a la derecha de la ventana el icono de *Network* en el apartado de *Gallery*.

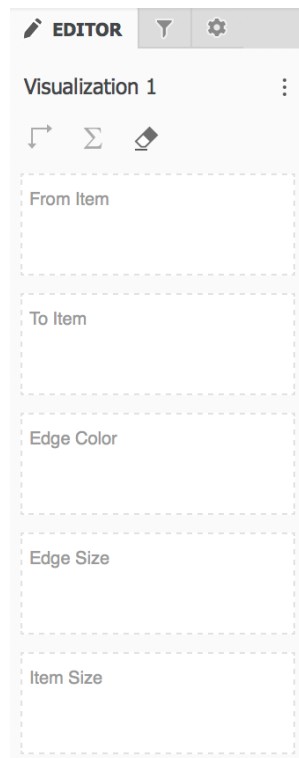


Figura 58. Editor para Network

En el editor para *Network* (ver Figura 58) hay que elegir los vértices que se van a unir. Para ello, se colocará en *From Item* el campo *White*, y en *To Item* el campo *Black*, para conectar todos los jugadores de blancas con los distintos jugadores de negras que hayan compartido tablero. Aparecerán todos los jugadores distintos que forman la base de datos, pero solamente se quiere la información de los doce jugadores que se eligieron para hacer los análisis. Por lo tanto, hay que filtrar esos doce jugadores añadiendo al filtro los campos *White* y *Black* y eligiendo en cada campo a los doce jugadores. De esta forma, ya aparecen los doce jugadores unidos todos entre sí ya que por lo menos se han enfrentado entre ellos una vez.

Para que sea más visual, se añade el campo *Row Count* a *Edge Color*, *Edge Size* e *Item Size* ya que son el número de veces que aparece la unión entre dos jugadores. De esta forma, el color de los arcos irá de rojo a verde para codificar el número de encuentros de menor a mayor. De igual modo, se codifica el tamaño del arco de menos grueso a más para codificar el número de encuentros como se hizo con el color. Por último, se modifica el tamaño de los vértices para ver cuál es el que más ha jugado contra los demás.

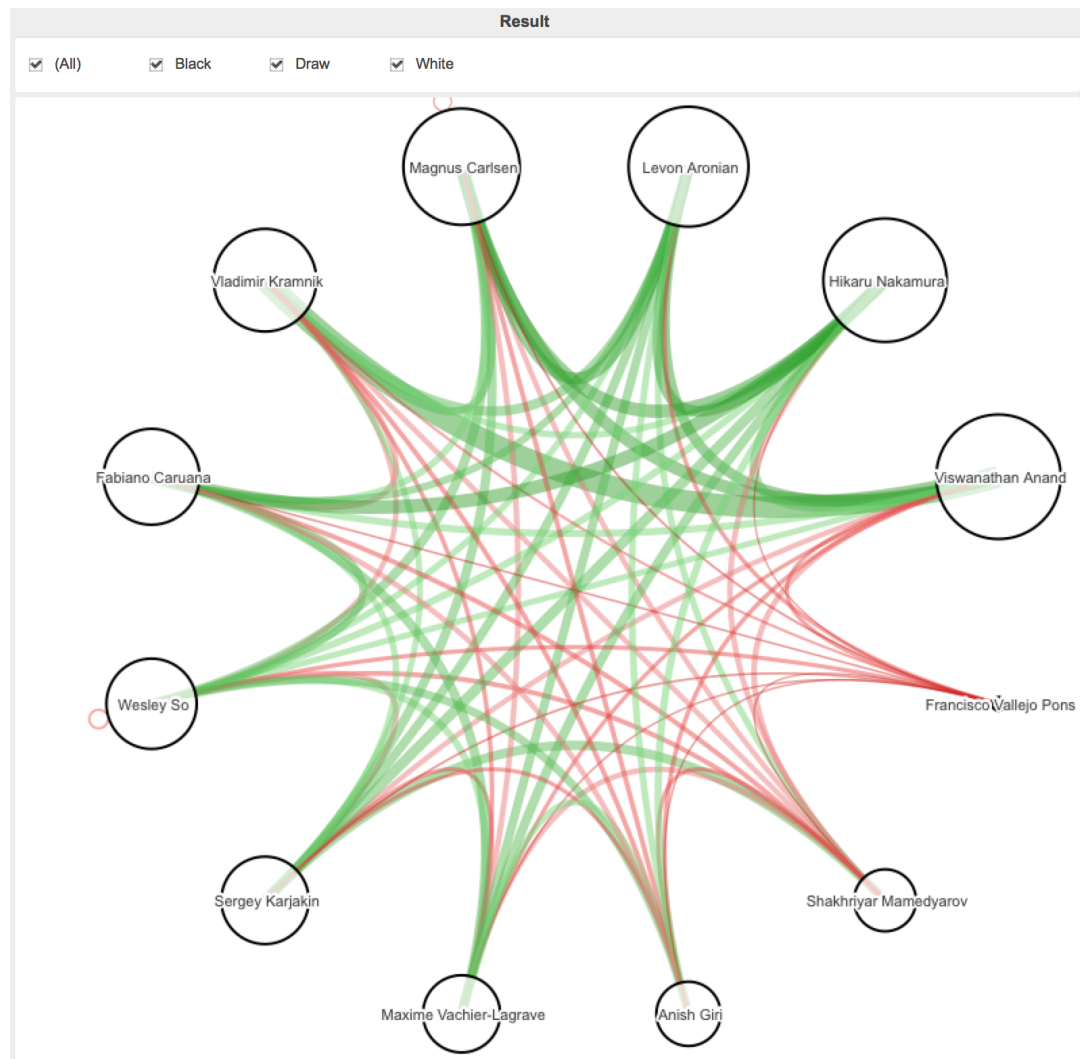


Figura 59. Grafo de los jugadores

Se añade un filtro con el resultado para elegir, por ejemplo, aquellas veces en las que se hayan enfrentado y hayan obtenido un resultado en particular. De esta forma, colocando el ratón encima de cualquier arco aparece la información de cuántas partidas han jugado entre ellos, jugando un ajedrecista con blancas, el otro con negras y viceversa. Aparece la información en los dos sentidos del vértice ya que es un grafo dirigido. Se separan las partidas cuando juegan con blancas o con negras con otro jugador en particular.

El orden de colocación empieza, si lo comparamos con las tres en un reloj y va en sentido antihorario del que más al que menos ha jugado. También se observa fácilmente que Vladimir Kramnik y Viswanathan Anand son los jugadores que más veces se han enfrentado ya que su arco es el de mayor tamaño. Es un valor esperado, ya que son los jugadores más veteranos del circuito de ajedrez mundial.

5.5. Ejemplo de uso

En este apartado, se va a realizar un ejemplo del uso de la visualización interactiva para analizar una serie de resultados utilizando los filtros.

En nuestro caso, queremos analizar las aperturas más utilizadas en la base de datos. Para ello, vamos a seleccionar del apartado *Most played openings* todas las aperturas que superan las 200 partidas.

Observamos que se han destacado en el gráfico de burbujas aquellas que cumplen con los requisitos que hemos puesto.

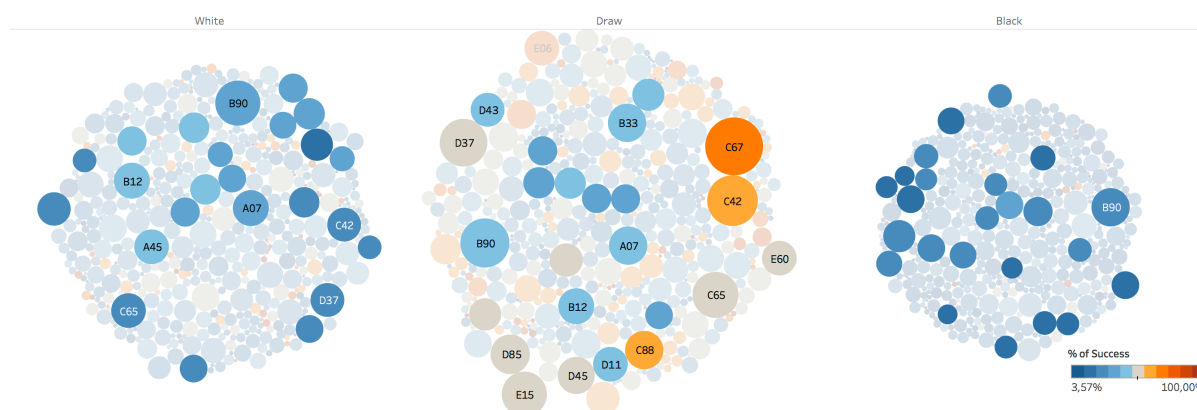
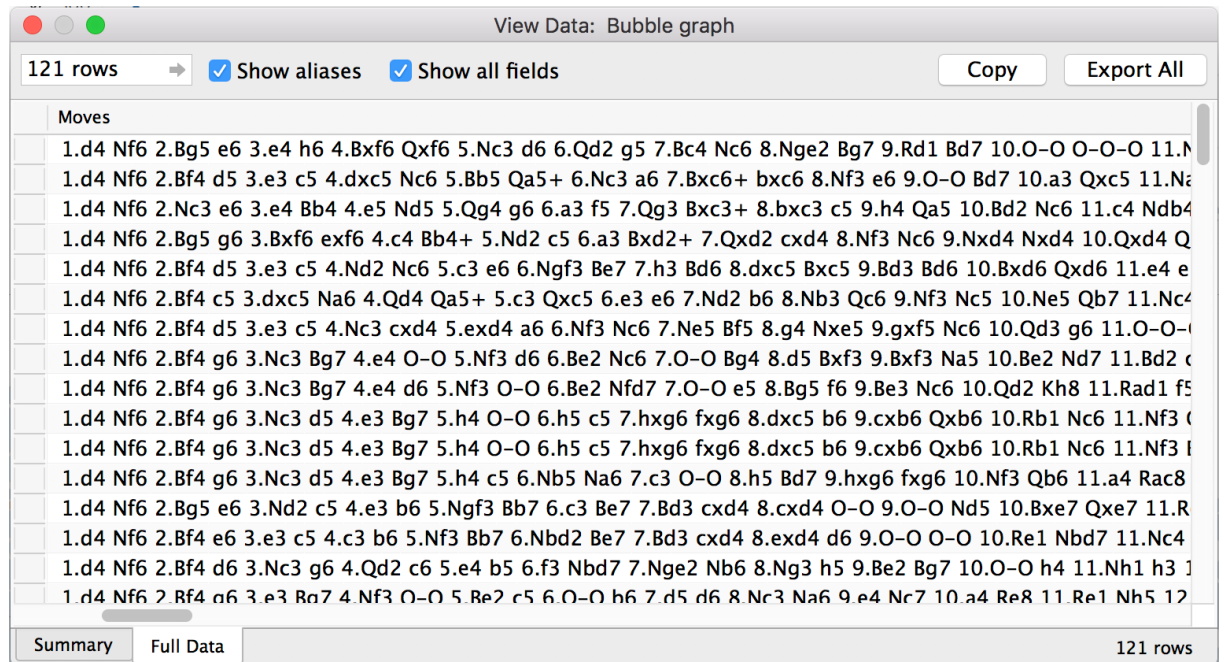


Figura 60. Ejemplo de uso, burbujas

Vamos a fijarnos en el gráfico de burbujas de la izquierda (ver Figura 60) que es el corresponde a las victorias del blanco y fijándonos en la leyenda con ayuda del *Tooltip* vamos a seleccionar la apertura con más porcentaje de victorias que es la que corresponde con el código ECO de A45. Vemos que tiene un total de 121 victorias para el blanco con esta apertura que representa el 41,16%, para las tablas tiene un 33,67% y para la victoria de negras tiene un 25,17%. Esta apertura nos podría resultar útil aprenderla ya que los resultados son bastante buenos.



Moves
1.d4 Nf6 2.Bg5 e6 3.e4 h6 4.Bxf6 Qxf6 5.Nc3 d6 6.Qd2 g5 7.Bc4 Nc6 8.Nge2 Bg7 9.Rd1 Bd7 10.O-O O-O-O 11.N
1.d4 Nf6 2.Bf4 d5 3.e3 c5 4.dxc5 Nc6 5.Bb5 Qa5+ 6.Nc3 a6 7.Bxc6+ bxc6 8.Nf3 e6 9.O-O Bd7 10.a3 Qxc5 11.Na
1.d4 Nf6 2.Nc3 e6 3.e4 Bb4 4.e5 Nd5 5.Qg4 g6 6.a3 f5 7.Qg3 Bxc3+ 8.bxc3 c5 9.h4 Qa5 10.Bd2 Nc6 11.c4 Ndb4
1.d4 Nf6 2.Bg5 g6 3.Bxf6 exf6 4.c4 Bb4+ 5.Nd2 c5 6.a3 Bxd2+ 7.Qxd2 cxd4 8.Nf3 Nc6 9.Nxd4 Nxd4 10.Qxd4 Q
1.d4 Nf6 2.Bf4 d5 3.e3 c5 4.Nd2 Nc6 5.c3 e6 6.Ngf3 Be7 7.h3 Bd6 8.dxc5 Bxc5 9.Bd3 Bd6 10.Bxd6 Qxd6 11.e4 e
1.d4 Nf6 2.Bf4 c5 3.dxc5 Na6 4.Qd4 Qa5+ 5.c3 Qxc5 6.e3 e6 7.Nd2 b6 8.Nb3 Qc6 9.Nf3 Nc5 10.Ne5 Qb7 11.Nc4
1.d4 Nf6 2.Bf4 d5 3.e3 c5 4.Nc3 cxd4 5.exd4 a6 6.Nf3 Nc6 7.Ne5 Bf5 8.g4 Nxe5 9.gxf5 Nc6 10.Qd3 g6 11.O-O-O
1.d4 Nf6 2.Bf4 g6 3.Nc3 Bg7 4.e4 O-O 5.Nf3 d6 6.Be2 Nc6 7.O-O Bg4 8.d5 Bxf3 9.Bxf3 Na5 10.Be2 Nd7 11.Bd2 c
1.d4 Nf6 2.Bf4 g6 3.Nc3 Bg7 4.e4 d6 5.Nf3 O-O 6.Be2 Nfd7 7.O-O e5 8.Bg5 f6 9.Be3 Nc6 10.Qd2 Kh8 11.Rad1 f5
1.d4 Nf6 2.Bf4 g6 3.Nc3 d5 4.e3 Bg7 5.h4 O-O 6.h5 c5 7.hxg6 fxg6 8.dxc5 b6 9.cxb6 Qxb6 10.Rb1 Nc6 11.Nf3 c
1.d4 Nf6 2.Bf4 g6 3.Nc3 d5 4.e3 Bg7 5.h4 O-O 6.h5 c5 7.hxg6 fxg6 8.dxc5 b6 9.cxb6 Qxb6 10.Rb1 Nc6 11.Nf3 f
1.d4 Nf6 2.Bf4 g6 3.Nc3 d5 4.e3 Bg7 5.h4 c5 6.Nb5 Na6 7.c3 O-O 8.h5 Bd7 9.hxg6 fxg6 10.Nf3 Qb6 11.a4 Rac8
1.d4 Nf6 2.Bg5 e6 3.Nd2 c5 4.e3 b6 5.Ngf3 Bb7 6.c3 Be7 7.Bd3 cxd4 8.cxd4 O-O 9.O-O Nd5 10.Bxe7 Qxe7 11.R
1.d4 Nf6 2.Bf4 e6 3.e3 c5 4.c3 b6 5.Nf3 Bb7 6.Nbd2 Be7 7.Bd3 cxd4 8.exd4 d6 9.O-O O-O 10.Re1 Nbd7 11.Nc4
1.d4 Nf6 2.Bf4 d6 3.Nc3 g6 4.Qd2 c6 5.e4 b5 6.f3 Nbd7 7.Nge2 Nb6 8.Ng3 h5 9.Be2 Bg7 10.O-O h4 11.Nh1 h3 1
1.d4 Nf6 2.Bf4 d6 3.e3 Bc7 4.Nf3 O-O 5.Be2 c5 6.O-O h6 7.d5 d6 8.Nc3 Na6 9.e4 Nc7 10.a4 Re8 11.Re1 Nh5 12

Figura 61. Información de los movimientos realizados

Podemos recoger la información de todas las partidas (ver *Figura 61*) que nos interesan que son aquellas en las que el blanco ha ganado. Para realizar esto, hacemos clic derecho en la burbuja y pulsamos en *View Data...*, dentro de la ventana que aparece, entramos en *Full Data* y ahí exportaremos todas las partidas con los movimientos que se realizaron en ellas. Podemos utilizar un tablero nuestro para ir viendo los movimientos que se fueron realizando y así estudiar cómo gracias a la apertura consiguió la ventaja el blanco para ganar, o podemos utilizar un software o página web de ajedrez en la que podremos copiar la partida y aparecerán todos los movimientos para navegar por ellos.

6. Conclusiones y trabajo futuro

Como conclusión cabe destacar que se ha cumplido, tanto con los objetivos específicos como con el objetivo principal del trabajo.

Se ha conseguido extraer un gran número de partidas siendo éstas alrededor de 25000, suficientes para poder realizar análisis con bastante fiabilidad. Se han procesado como estaba previsto para poder utilizar su información en las visualizaciones realizadas con Tableau.

Respecto al desarrollo de las visualizaciones, se ha conseguido crear un cuadro de mando sencillo para el usuario que responda a las posibles preguntas sobre los resultados de las aperturas y también para analizar a Grandes Maestros de ajedrez en concreto. Visualizar qué es lo que más juegan, ellos y entre ellos, a elección del usuario con la utilización de los filtros.

El planteamiento que se puede realizar para su trabajo futuro es que la base de datos de las partidas se vaya incrementando con las nuevas partidas que se disputen de forma automática. También se puede plantear como uso personal. Utilizando las propias partidas del usuario y sustituyendo la base de datos por la suya para así analizar sus aperturas y ver dónde hay que mejorar a título propio.

Sería interesante también, la inclusión de todo lo realizado en el trabajo en una aplicación web que contara también con un tablero para poder visualizar las aperturas directamente al seleccionarlas en la visualización.

7. Bibliografía

- Chess.com. (2007). *Chess.com*. Recuperado el 1 de Abril de 2018, de <https://www.chess.com/>
- Chess24. (2014). *Chess24*. Recuperado el 1 de Abril de 2018, de <https://www.chess24.com/>
- ChessBase. (2018). *ChessBase*. Recuperado el 1 de Abril de 2018, de <https://es.chessbase.com/>
- 2700Chess. (2018). *Live Chess Ratings*. Recuperado el 1 de Abril de 2018, de <https://2700chess.com/>
- Booij, M. d. (2018). *Marco's Website - Caissa Tools*. Recuperado el 1 de Abril de 2018, de https://www.debooy.eu/Java/caissatools_en.html
- Gómez, C. (2015). *Data Discovery con MicroStrategy 10: Una Guía Práctica*.
- MicroStrategy. (1989). *MicroStrategy*. Recuperado el 1 de Abril de 2018, de <https://www.microstrategy.com/es>
- MongoDB. (2008). *MongoDB*. Recuperado el 1 de Abril de 2018, de <https://www.mongodb.com/es>
- MongoDB Documentation. (2008). *The MongoDB 4.0 Manual*. Recuperado el 1 de Abril de 2018, de <https://docs.mongodb.com/manual/>
- Murray, D. G. (2016). *Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software*. John Wiley & Sons Inc.
- NoSQLBooster. (2018). *NoSQLBooster*. Recuperado el 1 de Abril de 2018, de <https://nosqlbooster.com/home>
- PGN Mentor. (2018). *Download PGN Files*. Recuperado el 1 de Abril de 2018, de PGN Mentor: <https://www.pgnmentor.com/files.html#players>
- Sinha, C. (2017). *Tableau 10 for Beginners: Step by Step guide to developing visualizations in Tableau 10*. Createspace Independent Pub.
- Tableau. (2003). *Tableau*. Recuperado el 1 de Abril de 2018, de <https://www.tableau.com/>