



MÁSTER UNIVERSITARIO EN BIG DATA

TRABAJO FIN DE MASTER

ANÁLISIS DE KPIS EN GOLF PROFESIONAL: IMPLEMENTACIÓN DE UN DASHBOARD EN POWER BI

Autor: Sofía Olalla Fernández de Soto

Director: José Luis Gahete Díaz

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Análisis de KPIs en Golf Profesional: Implementación de un Dashboard en Power BI
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2024/25 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.

Fdo.: Sofía Olalla Fernández de Soto

Fecha: 09/05/2025

Autorizada la entrega del proyecto

José Luis Gahete Díaz

Fdo.: José Luis Gahete Díaz

Fecha: 09/05/2025



MÁSTER UNIVERSITARIO EN BIG DATA

TRABAJO FIN DE MASTER

ANÁLISIS DE KPIS EN GOLF PROFESIONAL: IMPLEMENTACIÓN DE UN DASHBOARD EN POWER BI

Autor: Sofía Olalla Fernández de Soto

Director: José Luis Gahete Díaz

Madrid

ANÁLISIS DE KPIS EN GOLF PROFESIONAL: IMPLEMENTACIÓN DE UN DASHBOARD EN POWER BI

Autor: Olalla Fernández de Soto, Sofía

Director: Gahete Díaz, José Luis

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

El objetivo del proyecto presentado en el siguiente documento fue analizar el rendimiento de una jugadora profesional de golf mediante la recopilación y el procesamiento de diferentes aspectos clave de su juego.

Para ello se ha desarrollado un dashboard en Power BI que permite visualizar claramente cuales son los principales KPIs de su rendimiento, como por ejemplo el juego largo, el juego corto o el *putt*.

Para poder volcar toda esta información en el dashboard, se tuvo que hacer un proceso de limpieza y estructuración de los datos recogidos por la jugadora durante diferentes torneos. Una vez realizado este paso, se realizó un estudio estadístico para poder identificar patrones y diferentes áreas de mejora, de forma que pudiera planificar mejor sus entrenos. Este estudio estadístico, se plasmó en el dashboard, de forma que la visualización de los datos permitiera a la jugadora tomar mejores decisiones sobre sus entrenamientos, enfocándose en los aspectos de su juego que requirieran una mayor atención.

Con los resultados obtenidos se puede destacar la importancia de la analítica de datos en el mundo deportivo, ya que proporciona una herramienta visual, interactiva e intuitiva que facilita la optimización del rendimiento en el golf profesional.

ANALYSIS OF KPIS IN PROFESSIONAL GOLF: IMPLEMENTATION OF A DASHBOARD IN POWER BI

Author: Olalla Fernández de Soto, Sofía

Supervisor: Gahete Díaz, José Luis

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

The main objective of the project presented in this document was analysing the performance of a professional golf player *through* the collection and processing of some key aspects of her game.

For this purpose, a dashboard in Power BI was designed, so that the principal KPIs of her performance, such as long game, short game and *putting* could be clearly seen.

In order to put all this information in the dashboard, the data collected by the player during different tournaments needed to be cleaned and restructured. Once this step was finished, a statistical study was conducted to identify patterns and areas of improvement, allowing her to plan her training sessions better. This statistical study was then incorporated into the dashboard, enabling the player to visualize the data more effectively, and focused on the aspect of her game that required more attention.

The results obtained, proof the importance of data analytics in the sports world, due to the fact that it provides a visual, interactive and easy to use tool, which makes it easier to improve in the professional world.

Índice de la memoria

Capítulo 1. Introducción	5
Capítulo 2. Descripción de las Tecnologías.....	9
Capítulo 3. Estado de la Cuestión	10
Capítulo 4. Definición del Trabajo	13
4.1 Justificación.....	13
4.2 Objetivos	13
4.3 Metodología.....	14
4.3.1 Recogida de datos.....	14
4.3.2 Limpieza y estructuración de los datos.	16
4.3.3 Desarrollo del código para el análisis estadístico.	16
4.3.4 Creación de dashboard	17
4.3.5 Validación de los datos	17
4.4 Planificación y Estimación Económica.....	17
Capítulo 5. Desarrollo del proyecto	20
5.1 Recogida de datos.....	20
5.2 Desarrollo del código para el análisis estadístico.....	24
5.2.1 Código de juego largo.	25
5.2.2 Código de juego medio.....	25
5.2.3 Código del juego corto.	26
5.2.4 Código del putt.	27
5.3 Creación del dashboard en Power BI.	27
5.3.1 Modelado de datos del juego largo.	28
5.3.2 Modelado de datos del juego medio.....	28
5.3.3 Modelado de datos del juego corto.	29
5.3.4 Modelado de datos del putt.	30
5.4 Validación de los datos.....	31
Capítulo 6. Análisis de Resultados.....	32

6.1	Panel de juego largo.	32
6.2	Panel de juego medio.	33
6.3	Panel de juego corto.	34
6.4	Panel de putt.	35
Capítulo 7. Conclusiones y Trabajos Futuros		37
Capítulo 8. Bibliografía		39
Anexo I: Código de Python		41
Anexo II: Código de Power BI		47

Índice de figuras

Figura 1: Palos de golf y sus distancias medias.....	5
Figura 2: Tabla del juego largo.....	14
Figura 3: Tabla de juego medio.....	15
Figura 4: Tabla de juego corto.....	15
Figura 5: Tabla de putt	15
Figura 6: Esquema del flujo de datos	20
Figura 7: Diferentes pestañas del Excel	21
Figura 8: Panel del juego largo.....	32
Figura 9: Panel de juego medio	33
Figura 10: Panel de juego corto.....	34
Figura 11: Panel de putt.....	35

Índice de tablas

Tabla 1: Planificación del proyecto	18
Tabla 2: Estimación económica del proyecto.....	19
Tabla 3: Columnas y posibilidades del juego largo.....	21
Tabla 4: Columnas y posibilidades del juego medio.....	22
Tabla 5: Columnas y posibilidades del juego corto.....	23
Tabla 6: Columnas y posibilidades del putt	24

Capítulo 1. INTRODUCCIÓN

El golf es un deporte que comenzó a practicarse en el siglo XV en Escocia, pero no fue oficialmente fundado hasta 1744 cuando sus normas fueron puestas por escrito [1]. Actualmente, el golf es un deporte que se juega en 130 países y el número estimado de personas que lo practica es entre 55 y 80 millones, considerándose uno de los deportes más practicados mundialmente [2].

El principal objetivo del golf es, con ayuda de los catorce palos permitidos, meter una bola de pequeño tamaño en un hoyo en el menor número de golpes posibles.

Cada uno de los palos tiene una longitud diferente y la cara del palo tiene un ángulo determinado, de manera que con cada uno de ellos se hace más o menos distancia. Cuanto más largo sea el palo y menos grados tenga la cara, más distancia recorrerá la bola. El palo con el que mayor distancia realiza un jugador es el conocido como *Driver* (Figura 1), y por tanto será con el que mayor dispersión tendrán los golpes. Al contrario, cuanto más corto sea el palo y más ángulos tenga la cara del palo, menos distancia recorrerá la bola. Estos palos suelen ser más precisos y se suelen utilizar para dejar la bola lo más cerca posible del hoyo.

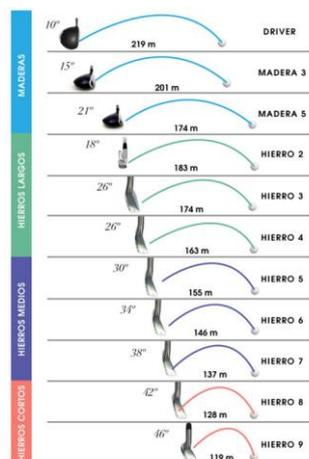


Figura 1: Palos de golf y sus distancias medias

Un recorrido estándar consta de 18 hoyos, que en función de su distancia se definen como pares tres, cuatro o cinco. Esto, llamado el par del hoyo, indica el número en el que debería realizarse cada hoyo, para un total de 72 golpes o par del campo. Cada hoyo consta de tres partes importantes, en el caso de los pares cuatro o cinco y dos partes importantes en el caso de los pares tres:

- *Tee de salida*: Se encuentra en los tres tipos de hoyos, y es la posición desde donde se da el primer golpe de cada hoyo. Dependiendo del tipo de hoyo, la distancia a la que se encuentra es mayor o menor.
- *Calle*: Solo se encuentra en los pares cuatro y pares cinco. Se trata del lugar óptimo donde se quiere colocar la bola después de pegar el primer golpe. Se trata de césped que está cortado a una longitud óptima y une el *tee de salida* con el *green*. En caso de fallar la *calle*, la jugadora se podría encontrar en dos situaciones diferentes, que pueden ser las siguientes:
 - *Bunker de calle*: se encuentra a lo largo de la *calle*, suelen encontrarse en los laterales, pero hay algunos casos en los que también están en el centro de la misma. Se trata de obstáculos arenosos, para dificultar el juego y exigir una mayor precisión a los jugadores. En este caso penalizan tiros imprecisos desde el *tee de salida*.
 - *Rough*: Se trata de los laterales de la *calle*, donde el césped está más alto y por lo tanto es más difícil tanto pegar a la bola como controlarla.
- *Green*: Se encuentra en los tres tipos de hoyos y se trata de la superficie donde el césped está muy corto cortado y va a ser donde la bola vaya rodando hacia el hoyo. Normalmente se trata de superficies con un pequeño relieve donde se puede colocar el hoyo en diferentes posiciones. Al igual que la *calle*, también se puede fallar a los mismos sitios, con la diferencia de que los *bunkers* de *green* están rodeando el *green*, no tiene por qué ser todo *bunkers* ya que el resto del espacio, a excepción de la entrada que sería *calle*, está rodeado por el *rough*.

Al contrario que en la mayoría de deportes, una puntuación más baja equivale a un mejor resultado e indica el nivel del jugador, el cual viene asociado a un *hándicap*. Cuanto menor

sea el *hándica* mayor será el nivel del jugador. El *hándicap* indica la cantidad de golpes que un jugador puede dar por encima del número de golpes establecido por el campo [3], es decir, si un jugador tiene *hándicap* tres, esto indicará que puede dar tres golpes por encima del par del campo que por norma general es 72. En este caso, los datos que se van a utilizar son de una jugadora profesional por lo que su *hándicap* es cero y busca hacer un resultado igual o menor de 72 golpes en cada vuelta.

En este trabajo se va a estudiar la precisión de una jugadora profesional, por lo tanto es necesario entender los tipos de golpes que hay y a que parte del juego hacen referencia. A continuación se van a explicar los diferentes tipos de juego y los golpes que se suelen dar:

- **Juego largo:** Se considera juego largo a los golpes que se dan desde el *tee* en los pares cuatro y en los pares cinco. Normalmente se utiliza los palos con los que más distancia se hace, que son las maderas o los hierros largos (*Figura 1*).
- **Juego medio:** Serían los golpes realizados desde el *tee de salida* en los pares tres y desde la *calle* en los pares cuatro y cinco, donde la distancia a la que se encuentra el *green* está entre los 180 metros y los 100 metros. En el caso de los pares cinco, este golpe va a ser un golpe de aproximación para luego dejarse un golpe más corto en el que debería de tener más precisión a la hora de dejar la bola cerca de la bandera.
- **Juego corto:** se trata de todos aquellos golpes que se encuentran a menos de 100m, los cuales suelen darse cuando se falla el *green* en el segundo golpe de los pares cuatros o en el primero de los pares tres, o en los pares cinco cuando no se llega a *green* y hay que dar un tercer golpe para llegar. Normalmente a estos golpes se les llama *Approach*. En el caso de que la jugadora falle el *green*, la jugadora va a buscar hacer *Approach* y *putt*, para así no perder golpes contra el campo.
- **Putt:** Se trata de los golpes que se dan dentro del *green*, con el *putter*. El juego del golf está pensado para tener dos golpes con este palo y dentro de esta superficie para cumplir el *hadicap*, por lo tanto en los pares tres, solo se tiene un golpe para alcanzar el *green*, en los pares cuatro, se tienen dos golpes para alcanzar el *green* y en los pares se tienen tres golpes para alcanzar el *green*.

Dentro de este trabajo se van a estudiar los cuatros aspecto del juego anteriormente mencionados con un análisis estadístico y posteriormente un dashboard de visualización de los datos recopilados. Con este informe se pretende facilitarle a la jugadora la planificación de los entrenamientos, ya que será capaz de ver en que aspecto de su juego debe de invertir más horas de entrenamiento.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este apartado se van a mostrar las tecnologías utilizadas a lo largo del proyecto. Se van a colocar en orden de uso y se va a resumir que fue lo que se hizo con ellas.

Para la realización de este proyecto se utilizaron las siguientes tecnologías:

- Microsoft Excel: para la recogida de los datos por parte de la jugadora profesional. Para que fuera más sencillo, la estructura del Excel se le mandaba al inicio del proyecto con todas las columnas que debía completar.
- Python: para realizar la limpieza, estructuración y el estudio estadístico previo. Como resumen, se cogió el Excel que proporcionaba la jugadora y se dividió en varias hojas de cálculo con los diferentes aspectos del juego que se querían estudiar.
- Power BI: donde se plasmó el estudio estadístico en un dashboard para que fuera más fácil de entender y aplicar por parte de la jugadora. El dashboard al igual que el estudio estadístico previo, se dividió por aspectos del juego, para que fuera más fácil de entender por parte de la jugadora, además se le explico como actualizar el Excel dentro del Power BI, de tal manera que tuviera acceso a los datos nada más acabar los diferentes torneos y saber que en que poder centrarse a la hora de entrenar durante la siguiente semana.

Capítulo 3. ESTADO DE LA CUESTIÓN

El golf siempre se ha considerado un deporte de percepción y técnica, que en los últimos años ha sufrido una gran evolución debido al uso de herramientas de analítica avanzada. Hoy en día, tanto los jugadores amateurs como los profesionales, emplean tanto sistemas basados en sensores como algoritmos predictivos y análisis estadísticos, para poder mejorar su rendimiento a la hora de entrenar o de salir al campo [4].

Esta tendencia ha hecho que la expansión de métricas como *Strokes Gained*, plataformas como *Data Golf* y dispositivos como *Arccos 360*, lo cuales permiten tomar los datos de los diferentes aspectos técnicos del juego de forma precisa y en tiempo real, sean cada vez más utilizados [5].

Hay diferentes métricas, que se consideran métricas clásicas como los “*putts por ronda*” o los “*greenes en regulación*” que tienen limitaciones a la hora de evaluar si una vuelta ha sido buena o mala [6]. Por ejemplo, si se consigue meter la bola desde fuera del *green*, el resultado es muy bueno, pero enmascara un mal golpe de aproximación a *green* o una mala salida o ambas. Otro ejemplo, sería tener un número bajo de *putts*, ya que puede ser por unos buenos chips, después de fallar el *green*, no tiene por qué ser por que le jugador sea un buen *putteador* [5][6][7].

Mark Broadie creo una nueva medida, denominada *Stroke Gained*, que cuantifica cuánto gana o pierde un jugador con comparación con el promedio del resto de jugadores desde una posición determinada del campo. Esta métrica permite evaluar los diferentes aspectos del juego más importantes, como son el juego largo, el juego corto y el *putting*. [8].

De forma adicional se utiliza la métrica *Shot Value*, que lo que hace es evaluar cada golpe según su impacto en el número esperado de golpes para completar el hoyo [6]. Además hay diferentes métricas que permiten analizar la precisión direccional, los errores en la distancia y el potencial alcanzado por cada jugador [8].

Se han utilizado diferentes sistemas con lo que se han recopilado millones de datos sobre los diferentes golpes desde 2023, permitiendo realizar estudios complejos sobre el rendimiento profesional. Además, hay plataformas que permiten tanto visualizar datos históricos como las probabilidades de victoria en tiempo real de los torneos, ya que ofrecen modelos predictivos ajustados a la dificultad que presenta el campo [9][10].

También hay herramientas que permiten a los golfistas, tanto amateurs como profesionales, analizar su juego con sensores inalámbricos que registran los golpes y luego permiten relacionarlos con mapas digitales del campo [4].

Se ha demostrado que el análisis de datos mejora tanto el entrenamiento como la estrategia a seguir a la hora de jugar al golf. Cuando los jugadores amateurs, tiene oportunidades de corrección inmediata, la puntuación mejora considerablemente, además este mecanismo permite disminuir la presión psicológica sobre los jugadores [11].

En los jugadores profesionales, uno de los más conocidos, Dustin Johnson, utilizó los datos para poder identificar sus debilidades en los golpes entre los 45 y 110 metros. Estos datos los saco con la ayuda del dispositivos del *Track Man*, que le permitió colocarse entre los mejores del circuito dentro de ese rango [12].

Hay varios estudios que indican que la mayor diferencia entre los jugadores profesionales y los jugadores amateurs, cuando se habla de rendimientos, se puede apreciar en el juego largo. En el caso de Tiger Woods, el mejor jugador de la historia del golf, su gran rendimiento, provienen en dos tercios de su juego largo, mientras que una pequeña arte se atribuye a su *putt*. Este aspecto se mantiene al comparar los grupos amateurs, donde la diferencia principal se encuentra la capacidad de dar golpes a más de 100 metros de distancia. Cuanta mayor sea la capacidad para pegar buenos golpes a más de 100 metros, menor será el *hándicap* de los jugadores [6][8][13].

El uso de la analítica de datos en un deporte como el golf, ha demostrado ser una herramienta eficaz a la hora de identificar los aspectos del juego donde el jugador debe de mejorar, de forma que se puedan optimizar los entrenamientos, buscar la mejor estrategia de juego que

se adapte al jugador y por tanto reducir la variabilidad en el rendimiento del jugador y que sea más estable. Además las métricas avanzadas mencionadas anteriormente, junto con las plataformas de análisis de datos y las diferentes tecnologías, que cada vez son más accesibles para todo tipo de jugadores, permiten que tanto los jugadores amateurs como los profesionales puedan tener una mejora continua en los diferentes aspectos de su juego.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Este trabajo surge debido a que una jugadora de golf profesional estaba buscando una forma de realizar unas estadísticas sobre su *putt*, para ver en que tenía que mejorar, para poder organizar mejor sus entrenamientos y ser más efectiva a la hora de mejorar.

A un nivel profesional es muy importante los pequeños detalles, por lo que se empezó primero con los aspectos del *putt*, como por ejemplo cual era el porcentaje de acierto a menos de 2 metro. A medida que la jugadora veía una mejora en su *putt*, gracias a los entrenamientos planteados por los números que salían en las estadísticas, tomo la decisión de añadir más aspectos al estudio estadístico, como el juego largo, el juego medio y el juego corto.

Por lo tanto el trabajo empezó de menos a más, por eso se planteó realizar un dashboard de Power BI donde la jugadora tuviera toda la información recogida. Además era una forma de que ella pudiera actualizar los datos de forma inmediata una vez finalizada el torneo, ya que tenía acceso al dashboard mediante la web de Power BI.

4.2 OBJETIVOS

Los objetivos del proyecto son los siguientes:

- Plantear una recogida de los datos que fuera fácil para la jugadora.
- Limpieza y la estructuración de los datos, para que su análisis fuera lo más sencillo posible
- Desarrollar un código, para realizar un análisis estadístico.
- Crear un “dashboard”, de manera que fuera más fácil e intuitivo para la jugadora entender los datos que se le estaban mostrando.

4.3 METODOLOGÍA

En este apartado se va a comentar la metodología que se ha utilizado para conseguir cada uno de los objetivos plasmados en el apartado anterior.

Esta metodología integral asegura que la recogida, tratamiento, análisis y presentación de los datos se realicen de manera eficiente y centrada en la experiencia de la jugadora, facilitando la toma de decisiones basada en información clara y confiable.

4.3.1 RECOGIDA DE DATOS.

Para realizar la recogida de datos se llevaron a cabo los siguientes pasos:

1. Definir qué datos eran necesarios para realizar el estudio: Para ello se tuvo en cuenta la experiencia de la jugadora y los datos recogidos en el libro “Every Shot Counts” escrito por Mark Broadie [14]. Se dividió la información a completar en los siguientes apartados:
 - a. Juego largo, donde se le pedía si había cogido la *calle* y en caso de fallarla y si la había fallado por el lado bueno o el lado malo y a qué lado de la *calle* se encontraba (*Figura 2*).

	Calle	Lado (drive)	Posicion
0	Si	NaN	NaN
1	Si	NaN	NaN
2	Si	NaN	NaN
3	Si	NaN	NaN
4	No	Malo	Der

Figura 2: Tabla del juego largo

- b. Juego medio, donde se le pedía la distancia que tenía al *green*, donde estaba colocada la bola, si en la *calle* o en el *rough*, si había cogido el *green*, en caso

de haberlo fallado si se había quedado por el lado bueno o malo y a que distancia la había dejado de la bandera (*Figura 3*).

	Distancia a green	Localizacion	Green	Lado	Resultado
0	149	Calle	No	Malo	9.0
1	152	Calle	No	Bueno	16.0
2	118	Calle	Si	NaN	9.0
3	147	Calle	Si	NaN	15.0
4	90	Calle	Si	NaN	3.5

Figura 3: Tabla de juego medio

- c. Juego corto, donde se le pedía la distancia que tenía de *approach*, si había hecho *approach* y *putt*, desde donde hacia el *approach* y la distancia a la que la deja de la bandera (*Figura 4*).

	Distancia app	App y putt	Localizacion app	m result
0	9	Si	Bunker	4.0
1	16	No	Calle	14.0
2	14	Si	Calle	1.5
3	7	Si	Rough	1.0
4	8	Si	Bunker	1.0

Figura 4: Tabla de juego corto

- d. *Putt*, donde se le pedía la distancia del *putt*, si la había metido o no y en caso de no meterla si se había quedado corta o se había ido larga y si había fallado por arriba o por abajo, además de la caída que tenía el *putt* (*Figura 5*).

	Distancia	Embocado	Resultado 1	Resultado 2	Caida
0	10.0	NO	LARGA	ABAJO	IZQ-DER
1	6.0	SI	NaN	NaN	IZQ-DER
2	1.0	SI	NaN	NaN	RECTO
3	12.0	NO	CORTA	ABAJO	DER-IZQ
4	1.0	SI	NaN	NaN	DER-IZQ

Figura 5: Tabla de putt

Se decidió que para que a la jugadora no se le hiciera muy pesado recoger las estadísticas lo haría al finalizar cada una de las vueltas de los torneos.

2. Recogida de datos: se le proporciono una plantilla de Excel con diferentes desplegables para que fuera más fácil completarla. Además, se trataba de un documento compartido por lo que en el momento que se completaba, ya se podía recargar el análisis estadístico, para que cogiera los nuevos datos.

4.3.2 LIMPIEZA Y ESTRUCTURACIÓN DE LOS DATOS.

Los datos que proporcionaba la jugadora, estaban todos en un mismo archivo de Excel. Puesto que para cada parte del informe, solo se iban a utilizar una serie de columnas, se tomó la decisión de dividir la tabla de la jugadora en una tabla para cada aspecto del juego, por lo que se tendría lo que se ha visto en el apartado anterior, cuatro tablas diferentes, cada una utilizada en una parte del análisis.

4.3.3 DESARROLLO DEL CÓDIGO PARA EL ANÁLISIS ESTADÍSTICO.

El código se desarrolló en Python utilizando Visual Studio Code. Al igual que en el caso de las tablas, donde se guardaba la información para cada aspecto del juego, se decidió hacer el código dividido según el aspecto del juego que se quisiera estudiar, por lo tanto hay un notebook (archivo .ipynb) para cada una de las partes, los cuales serían los siguientes:

- *EstadisticasDrive.ipynb*: Hace referencias a los aspectos del juego largo.
- *EstadisticasGolpesGreen.ipynb*: Hace referencias a los aspectos del juego medio.
- *EstadisticasApproach.ipynb*: Hace referencia a los aspectos del juego corto.
- *EstadisticasPutt.ipynb*; Hace referencia al *putt*.

El código se basa principalmente en funciones que realizan diferentes filtros y cálculos para calcular un porcentaje sobre un aspecto del juego. Luego dichos resultados se enseñan por pantalla y se pueden ver los resultados.

Puesto que esto era complicado de entender por parte de la jugadora, se tomó la decisión de realizar el mismo informe de forma más grafica con ayuda de Power BI.

4.3.4 CREACIÓN DE DASHBOARD

Para la creación del dashboard se utilizó Power Bi, puesto que se tenía licencia para poder utilizarlo y gracias a la página web, la jugadora podía verlo en cualquier parte del mundo de forma actualizada.

El diseño se centró en que fuera fácil de interpretar, con gráficos que resultaran fáciles de leer. Al igual que en el caso del código se dividió en cuatro pestañas:

- *Drive*
- *Golpe a green*
- *Approach*
- *Putt*

En cada una de las pestañas había diferentes KPIs para que se viera de forma fácil, además de filtros y diferentes gráficos y tablas.

4.3.5 VALIDACIÓN DE LOS DATOS

Para validar los datos, puesto que todos los cálculos se habían realizado anteriormente en el código, se comprobó que coincidieran entre sí, es decir, cálculos obtenidos mediante Python debían de ser los mismos que los que aparecían en Power BI.

Para que los datos se actualizarán de forma automática, se le enseñó a la jugadora a volver a insertar el Excel en Power BI, de forma que cada vez que acabara un torneo pudiera meter la información y así poder plantear su siguiente semana de entrenamientos, sin necesidad de contar con nadie para la actualización de datos.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

A la hora de realizar la planificación del proyecto se tomó la decisión de dividir el proyecto en diferentes fases, con una duración diferente y cada una de ellas con diferentes tareas a

realizar. En la siguiente tabla (*Tabla 1*), se puede ver un resumen de cómo se planteó la planificación del proyecto:

Fase	Duración	Actividad a realizar
Primera fase	1 semana	Planificación y diseño
Segunda fase	1 semana	Desarrollo de recogida de Datos
Tercera fase	2-3 semanas	Limpieza y estructuración de datos y desarrollo del código para el análisis estadístico
Cuarta fase	2-3 semanas	Creación del dashboard
Quinta fase	1 semana	Validación de los datos y ajustes finales
Sexta fase	1 semana	Entrega y plan de mantenimiento

Tabla 1: Planificación del proyecto

Teniendo en cuenta la planificación anterior, el coste del proyecto se va a calcular en base a un precio por hora. Como la duración del proyecto fue de aproximadamente 10 semanas y teniendo en cuenta que se realizó mientras se trabajaba a tiempo completo en una empresa, se va a poner de media que se pudieran dedicar unas 20 horas a la semana, teniendo en cuenta los fines de semana, haciendo un total de 200 horas para realizar el proyecto.

Teniendo en cuenta que este trabajo lo desarrollaría un analista de datos junior y viendo las tarifas que cobran, que es alrededor de unos 30,00 euros la hora trabajando como freelance [15]. Además es necesario una licencia de Power BI de tipo Power BI Pro, ya que era necesario la utilidad de poder compartir el informe por internet [16].

Por lo que la estimación económica quedaría de la siguiente forma (*Tabla 2*):

	Horas en realizar el proyecto	Precio	Total
Analista de datos junior	200 horas	30€/h	6000€
Licencia de Power Bi	2 licencias	9.40€	18.80€
Total			6018.80€

Tabla 2: Estimación económica del proyecto

Capítulo 5. DESARROLLO DEL PROYECTO

En el siguiente capítulo, se va a explicar todo el desarrollo del proyecto de manera extensa, explicando bien los pasos seguidos, las diferentes decisiones que se tuvieron que tomar y las dificultades que se fueron encontrando a lo largo del proyecto.

El flujo de los datos se puede ver en la siguiente imagen (*Figura 6*):



Figura 6: Esquema del flujo de datos

Para poder llevar a cabo el anterior flujo (*Figura 6*), fue necesario pensar una forma para que la recogida de datos fuera sencilla y desarrollar tanto el código con el análisis estadístico como el dashboard para la visualización de los datos. En un principio el desarrollo del código y el del dashboard, se desarrollaron con un tamaño de datos muy pequeño, simplemente para poder hacer las pruebas y ver que tanto el código como el dashboard obtenían los mismos resultados.

5.1 RECOGIDA DE DATOS.

La recogida de datos se realizó mediante un documento Excel compartido, donde la jugadora iba añadiendo los diferentes datos que se le pedían.

Para que fuera más fácil, todas las columnas del Excel, que no fueran numéricas, tenían despleables con las diferentes opciones con las que se podía rellenar el campo. A la hora

de rellenar el Excel, y para que fuera más sencillo, dentro del mismo Excel había una pestaña para cada aspecto del juego (*Figura 7*).

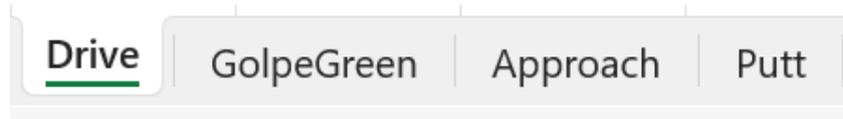


Figura 7: Diferentes pestañas del Excel

A continuación, se va a explicar cómo era cada una de las tablas que debía de rellenar la jugadora y los posibles datos para cada una de las columnas.

La primera tabla que se va a explicar es la del juego largo (*Figura 2*), que hace referencia a la pestaña de *Drive* (*Figura 7*). Esta tabla tiene las siguientes columnas y posibilidades (*Tabla 3*):

Columnas	Posibilidades
<i>Calle</i>	SI/NO
Lado (<i>drive</i>)	BUENO/MALO
Posición	DER/IZQ

Tabla 3: Columnas y posibilidades del juego largo

La columna de *calle* hace referencia a si el golpe que realiza desde el *tee de salida* acaba en la *calle* o no. En caso de que sea que sí, las otras dos columnas no se rellenaban. En caso de que fallara la *calle*, se debía de rellenar las otras dos columnas, con el lado al que había fallado la *calle* y cuál era su posición. Estos dos campos, se rellenaban con respecto al siguiente golpe y bajo el juicio de la jugadora.

La siguiente tabla sería la de juego medio (*Figura 3*), que hace referencia a la pestaña *GolpeGreen* (*Figura 7*; *Error! No se encuentra el origen de la referencia.*). Esta tabla tiene las siguientes columnas y posibilidades (*Tabla 4*):

Columnas	Posibilidades
Distancia a <i>green</i>	Se trata de un campo numérico, que indica la distancia del golpe que debe de dar.
Localización	CALLE/ROUGH/BUNKER
<i>Green</i>	SI/NO
Lado	BUENO/MALO
Resultado	Es un campo numérico, que marca la distancia a la que la deja del hoyo

Tabla 4: Columnas y posibilidades del juego medio

En esta tabla hay dos campos numéricos, el de Distancia a *Green* y el de Resultado. El primero hace referencia a la distancia del golpe que debe de pegar y el segundo hace referencia a la distancia a la que la deja del hoyo. En cuanto al resto de los campos, el de Localización hace referencia a desde que superficie pega el golpe, el de *Green* si la bola acaba en el *green* o no, en caso de que sí que acabe en el *green* el campo de lado, no había que completarlo, mientras que si fallaba el *green* era necesario que indicara porque lado lo había fallado. Al igual que en el caso del juego largo, este campo se rellenaba según el juicio de la jugadora.

La siguiente tabla sería la del juego corto (*Figura 4*), que hace referencia a la pestaña *Approach* (*Figura 7*). Esta tabla tiene las siguientes columnas y posibilidades (*Tabla 5*):

Columnas	Posibilidades
Distancia <i>Approach</i>	Se trata de un campo numérico que indica la distancia del golpe que debe de realizar
<i>Approach</i> y <i>putt</i>	SI/NO
Localización	CALLE/ROUGH/BUNKET
Resultado	Se trata de un dato numérico que indica a que distancia la deja del hoyo después de realizar el golpe

Tabla 5: Columnas y posibilidades del juego corto.

En esta tabla (*Tabla 5*) hay dos campos numéricos, *Distancia Approach* y *Resultado*, el primero hace referencia a la distancia del golpe que debe de pegar y el segundo hace referencia a la distancia que la deja del hoyo. El campo de *Approach* y *putt* hace referencia a si desde donde se encuentra la mete en dos golpes y el campo de *Localización* hace referencia a la superficie desde la cual se da el golpe.

La última tabla seria la del *putt* (*Figura 5*), que hace referencia a la pestaña *Putt* (*Figura 7*). Esta tabla tiene las siguientes columnas y posibilidades (*Tabla 6*):

Columnas	Posibilidades
Distancia	Se trata de un campo numérico que indica la distancia del <i>putt</i> que tiene que tirar
Embocado	SI/NO
Resultado 1	LARGA/CORTA
Resultado 2	ARRIBA/ABAJO
Caída	RECTA/IZQ-DER/DER-IZQ

Tabla 6: Columnas y posibilidades del putt

En esta tabla (*Tabla 6*) solo cuenta con un campo numérico, que es la de Distancia, donde se recoge la longitud del *putt* que debe de tirar la jugadora. La columna de Embocado indica si ha metido el *putt* o no, en caso de embocarlo, no era necesario rellenar las columnas de Resultado 1 y Resultado 2, pero si la de Caída, que indica hacia donde caía la bola a medida que iba avanzando en el *putt*. En Resultado 1 se indicaba si se había pasado o no había llegado al hoyo y en el Resultado 2, si la había dejado por encima del hoyo o por debajo. El primer campo tiene que ver con la fuerza con la que tira el *putt* y el segundo campo tiene que ver con la caída que le daba al *putt*.

Estas eran las cuatro tablas que debía de rellenar la jugadora después de jugar cada uno de los torneos. Estos datos los iba recogiendo a medida que iba jugando y luego los insertaba en las tablas mostradas.

5.2 DESARROLLO DEL CÓDIGO PARA EL ANÁLISIS ESTADÍSTICO.

El código, al igual que las tablas está dividido en cuatro partes, las cuales se van a explicar a continuación.

5.2.1 CÓDIGO DE JUEGO LARGO.

A partir de la tabla de juego largo (*Figura 2*) se han calculado los siguientes campos:

- El porcentaje de *calles* cogidas.
- El porcentaje de las *calles* falladas a la derecha y a la izquierda.
- El porcentaje de las *calles* falladas por el lado bueno.

Para realizar este código, el cual era muy sencillo, no hizo falta ni hacer funciones, puesto que no era algo que se fuera a utilizar más de una vez. Se hizo utilizando las librerías de *pandas* y *numpy* para poder hacer el tratamiento de los datos.

5.2.2 CÓDIGO DE JUEGO MEDIO.

En este caso, el código se organizó en varias funciones, ya que iba a ser código que se iba a utilizar recurrentemente, ya que los cálculos que se hicieron dependían de la distancia a la que se encontraba del *green*. Al igual que en el caso anterior, se utilizaron las librerías de *pandas* y *numpy* para poder hacer el tratamiento de los datos.

Con este código se han calculado los siguientes campos a partir de la tabla de juego medio (*Figura 3*), además el código desarrollado se puede ver en el *Anexo I: Código de Python (Código juego medio.)*:

- El porcentaje de *greenes* que cogía la jugadora según la distancia.
- La mediana y la media de la distancia a la que la deja del hoyo dependiendo de la distancia. En este caso se cogió ambas medidas por que la media tenían en cuenta todos los golpes y a ojos de la jugadora era lo mejor, pero se le convenció para añadir la mediana, ya que esta no se ve afectada por los outliers. Además, para estas dos medidas también se tuvo en cuenta si cogía el *green* o no, puesto que las distancias iban a ser mucho más grandes en caso de no coger el *green*.
- El porcentaje de golpes que daba según la distancia a la que se encontraba, por ejemplo, cuál era el porcentaje que daba entre veinte y treinta metros.

- De los *greenes* que fallaba, había que calcular el porcentaje que fallaba por el lado bueno.

Todos estos parámetros se calcularon para rangos de distancia de diez metros, empezando a diez metros del *green* y acabando a doscientos sesenta metros de *green*.

5.2.3 CÓDIGO DEL JUEGO CORTO.

Al igual que en el caso anterior, se dividió el código en funciones, de manera que los cálculos se pudieran hacer para rangos de distancia. En este caso, además también había que tener en cuenta la superficie desde donde se pegaba el golpe, por lo tanto, para un mismo rango de distancia, había tres posibilidades, *calle*, *Bunker* o *rough*. También se utilizaron las librerías de *pandas* y *numpy* para poder hacer el tratamiento de los datos.

Con este código se han calculado los siguientes campos a partir de la tabla de juego corto (*Figura 4*), además el código desarrollado se puede ver en el *Anexo I: Código de Python (Código del juego corto)*:

- Porcentaje de veces que era capaz de hacer un “*up and down*” para un rango de distancia y en una superficie determinada. Un “*up and down*” consiste en meter la bola en el hoyo en dos golpes desde fuera del *green*, es decir no perder golpes con respecto al campo por haber fallado el *green*.
- La mediana y la media de la distancia a la que la deja del hoyo dependiendo de la distancia. En este caso se cogió ambas medidas por que la media tenían en cuenta todos los golpes y a ojos de la jugadora era lo mejor, pero se le convenció para añadir la mediana, ya que esta no se ve afectada por los outliers, o algún mal golpe que pegara.

Todos estos parámetros se calcularon según un rango de distancia de diez metros, empezando en distancias menores de diez metros y con un máximo de sesenta metros. Además, había que tener en cuenta también la superficie desde la cual pegaba el golpe.

5.2.4 CÓDIGO DEL *PUTT*.

En este caso, el código se organizó en varias funciones, ya que iba a ser código que se utilizar recurrentemente, ya que los cálculos que se hicieron dependían de la distancia a la que se tiraba el *putt*. Al igual que en el caso anterior, se utilizaron las librerías de pandas y numpy para poder hacer el tratamiento de los datos.

Con este código se han calculado los siguientes campos a partir de la tabla de *putt* (Figura 5), además el código desarrollado se puede ver en el *Anexo I: Código de Python (Código del putt.)*:

- Porcentaje de acierto según distancia.
- Porcentaje de *putt* tirados para los diferentes tipos de caída según un rango de distancia.
- Porcentaje de aciertos según el tipo de caída que tenía el *putt* y teniendo en cuenta la distancia del *putt*.
- Porcentaje de falladas por arriba según la distancia a la que tiraba el *putt*.

Todos estos parámetros se calcularon según un rango de mucho más pequeños que en los casos anteriores, puesto que las distancias en el *putt* son mucho más pequeñas, por lo que las distancias van desde menos de dos metros hasta nueve metros.

Este es el código desarrollado para realizar el estudio estadístico según los datos que quería conocer la jugadora sobre su juego.

5.3 CREACIÓN DEL DASHBOARD EN POWER BI.

Para que los datos fueran más fáciles de entender por parte de la jugadora, se tomó la decisión de crear un dashboard en Power BI. Se calcularon los mismos parámetros que se calcularon en el código de análisis estadístico que se acaba de explicar. En esta sección se va a explicar cómo se creó el dashboard.

Lo más importante es saber, que el dashboard al igual que la recogida de datos y al igual que el código, tienen cuatro apartados, cada uno de ellos haciendo uso de los datos recogidos en cada una de las tablas completadas por la jugadora, para cada uno de los aspectos de su juego.

5.3.1 MODELADO DE DATOS DEL JUEGO LARGO.

Para realizar este modelado se utilizó la tabla del juego largo (*Figura 2*). Todo el código utilizado para crear el dashboard se puede ver en el *Anexo II: Código de Power BI (Modelado de datos del juego largo.)*

Lo primero que se hizo fue completar el Excel y tanto en la columna Lado (*drive*) como en la de Posición, las cuales la jugadora había dejado vacías en caso de coger la *calle*, se completaron con la palabra *Calle* para que en el dashboard se indicara.

Además se tuvieron que crear tres medidas, las cuales fueron las siguientes:

- PorcentajeBueno, donde se calculaba el porcentaje de *calles* que había fallado por el lado bueno.
- PorcentajeMalo, donde se calculaba el porcentaje de *calles* que había fallado por el lado malo.
- PorcentajeCalle, donde se calculaba el porcentaje de *calles* cogidas por la jugadora.

Al igual que en el caso del código, este fue el modelado más sencillo de hacer.

Los gráficos que se incluyeron en esta parte del dashboard fueron un “Pie chart” y un “Stacked column chart”.

5.3.2 MODELADO DE DATOS DEL JUEGO MEDIO.

Para realizar el modelado del juego medio, se utilizó la tabla de juego medio (*Figura 3*). Todo el código utilizado para crear el dashboard se puede ver en el *Anexo II: Código de Power BI (Modelado de datos del juego medio.)*

En este caso lo primero que se hizo fue cambiar el tipo de dato de la columna Distancia a *green*.

Lo siguiente que se hizo fue sustituir los puntos por comas en la columna de Resultado, puesto que sino no se ponía poner de tipo número.

El siguiente paso fue añadir una columna donde se calculara el rango de distancia donde se encontraba la jugadora. Los rangos eran de diez metros, por lo tanto si la jugadora se encontraba a 145m, el rango donde se encontraría sería 140m. También se aprovechó para cambiar el tipo a tipo entero.

Luego se duplicó esta columna para luego poder ordenarlo en los gráficos. En este caso esta columna debía de ser de tipo texto.

Al igual que en el modelado del juego largo, se tuvo que añadir datos en la columna de Lado, ya que la jugadora en caso de coger *green* la dejaba vacía. Y por último se cambió el nombre de las columnas

Después de hacer todas estas modificaciones sobre la tabla, se calculó la siguiente medida:

- *PorcentajeGreen*, en la cual se calculaba el porcentaje de *greenes* que cogía la jugadora.

Los gráficos que se utilizaron para representar los datos en este aspecto del juego fueron un “Pie chart”, un “Stacked column chart”, y una tabla

5.3.3 MODELADO DE DATOS DEL JUEGO CORTO.

El modelado de los datos del juego corto se hizo sobre la tabla de juego corto (*Figura 4*).
Anexo II: Código de Power BI (Modelado de datos del juego corto.)

Lo primero que se hizo fue cambiar el tipo en la columna de Distancia app.

Luego se añadió una columna donde se calculó el rango de distancia para luego poder utilizarlo en diferentes gráficos y se puso de tipo entero.

En la columna de `m result`, se sustituyó los puntos por comas y se modificó el tipo a tipo `number`,

Por último se duplicó la columna de `RangoDistancia` y se puso de tipo `texto` para luego poder utilizarlo.

Después realizar estos cambios se calculan las siguientes medidas:

- `PorcentajeAppPutt`, donde se calculaba el porcentaje de `aproach` y `putt`.
- `AppPutBunker`; donde se calculaba el porcentaje de `aproach` y `putt` desde el `bunker`.
- `AppPutCalle`, donde se calculaba el porcentaje de `aproach` y `putt` desde la `calle`.
- `AppPutRough`, donde se calculaba el porcentaje de `aproach` y `putt` desde el `rough`
- `MedianaPorRangoSuperficie`, donde se calculaba la mediana del resultado obtenido después de dar el golpe.

Los gráficos que se utilizaron para representar los datos del juego corto fueron un “Stacked column chart”, y una tabla.

5.3.4 MODELADO DE DATOS DEL *PUTT*.

Para realizar el modelado de los datos del `putt`, se utilizaron los datos de la tabla de `putt` (Figura 5). Todo el código utilizado para crear el dashboard se puede ver en el *Anexo II: Código de Power BI (Modelado de datos del putt.)*

Lo primero que se hizo fue sustituir los puntos por comas de la columna `Distancia`, para poder luego cambiarlo a tipo número.

Luego se añade una columna donde se calcula el rango de distancia. Estos rangos son de un metro hasta llegar a siete metros, luego de siete metros a diez metros y por ultimo de `putts` mayores de diez metros.

Después se crea una columna que ordene los rangos para luego poder utilizarlo dentro del dashboard.

Una vez realizadas estas modificaciones sobre la tabla, se crearon las siguientes medidas:

- Acierto, donde se ha calculado el porcentaje de acierto
- CaidaDerIzq, donde se ha calculado el porcentaje de *putts* tirados de derecha a izquierda.
- CaidaIzqDer, donde se ha calculado el porcentaje de *putts* tirados de izquierda a derecha.
- CaidaRecto, donde se ha calculado el porcentaje de *putts* tirados rectos.
- FalladasArriba, donde se ha calculado el porcentaje de *putts* fallados por arriba.

Los gráficos que se utilizaron para representar los datos del *putt* fueron un “Pie chart”, un “Stacked column chart”, y una tabla.

5.4 VALIDACIÓN DE LOS DATOS.

Para realizar la validación de los datos, se comprobó que lo obtenido en el estudio estadístico realizado mediante el código, era el mismo dato que mostraba en el dashboard de power BI. Para ello se fue comparando poco a poco lo obtenido en cada parte del trabajo.

Capítulo 6. ANÁLISIS DE RESULTADOS

Como resultado del proyecto se entregó el dashboard de Power BI, que se ha explicado como se ha creado en el apartado anterior. En este apartado se va a mostrar el resultado final del dashboard.

Como ya se ha comentado previamente, el dashboard está dividido en cuatro apartados, cada uno haciendo uso de los datos de cada una de las tablas rellenas por la jugadora.

6.1 PANEL DE JUEGO LARGO.

El resultado obtenido para el panel del juego largo se puede ver en la siguiente imagen (Figura 8):

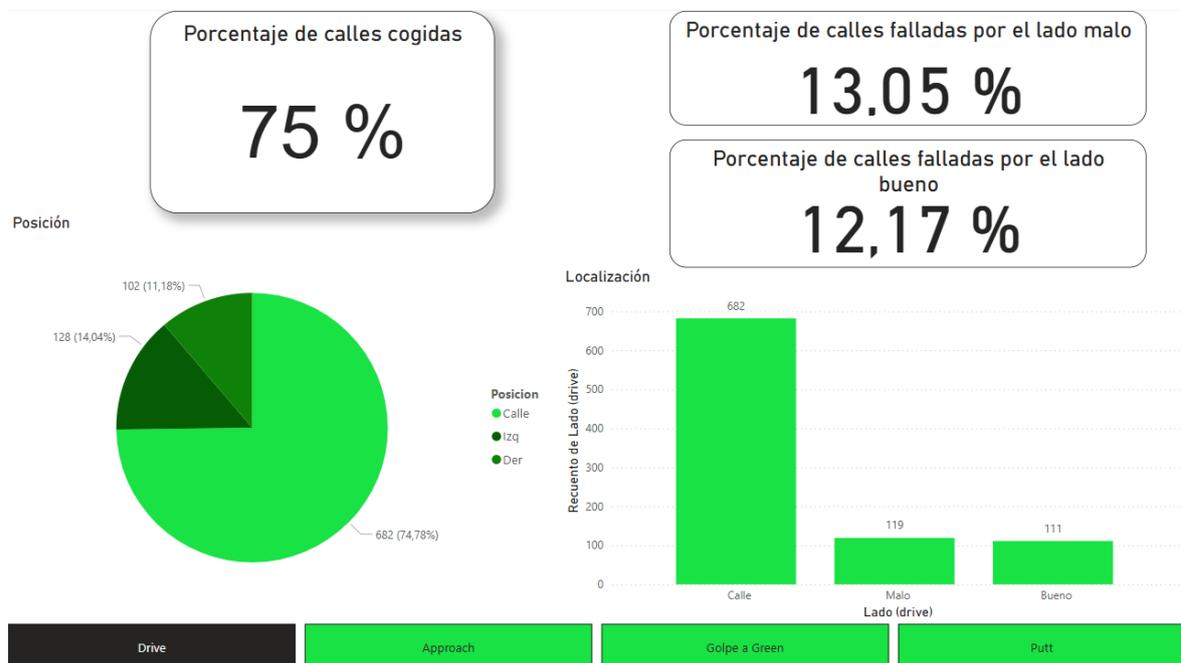


Figura 8: Panel del juego largo

En este panel se pueden ver como KPIs el porcentaje de *calles* cogidas y el porcentaje de *calles* falladas por el lado bueno y por el lado malo.

Como se puede ver en el dashboard, la jugadora tiene un porcentaje alto a la hora de coger las *calles* desde el *tee de salida* y aproximadamente tiene el mismo porcentaje a la hora de fallar por el lado malo que por el lado bueno. Es cierto que tiende a fallar un poco más por la parte izquierda de la *calle*, por lo que debería de centrarse en intentar mejorar ese fallo a la izquierda, para aumentar el porcentaje de *calles* cogidas.

6.2 PANEL DE JUEGO MEDIO.

El resultado obtenido para el juego medio se puede ver en la siguiente imagen (Figura 9):



Figura 9: Panel de juego medio

En el panel del juego medio (Figura 9), como KPI tenemos el porcentaje de *greenes* que coge en regulación. En este panel, la jugadora puede poner filtros, para ver la información de los golpes cuando coge *green* y cuando no. Además, tiene la información sobre la mediana de la distancia a la que deja la bola dependiendo de la distancia a la que se encuentra, que se puede ver que sigue una línea ascendente, cuanto más lejos se encuentra del *green* más lejos deja la bola del hoyo, además el porcentaje de *greenes* fallados a medida que va aumentando la distancia desde donde está suele aumentar. También tiene información sobre el número

de golpes que da según la distancia a la que se encuentra. En este caso se puede ver que sigue una distribución normal, dando muchos más golpes en los valores que están céntricos. Por último, cuenta con una tabla donde puede ver el porcentaje de *greenes* cogidos según la distancia a la que pega el golpe.

Viendo los datos, la recomendación para la jugadora, sería que practicara los golpes entre 140 y 160 metros, ya que son las distancias a las que más golpes pega, para aumentar la probabilidad de coger el *green* y mejorar sus resultados.

6.3 PANEL DE JUEGO CORTO.

El panel de juego corto se puede ver en la siguiente imagen (*Figura 10*)

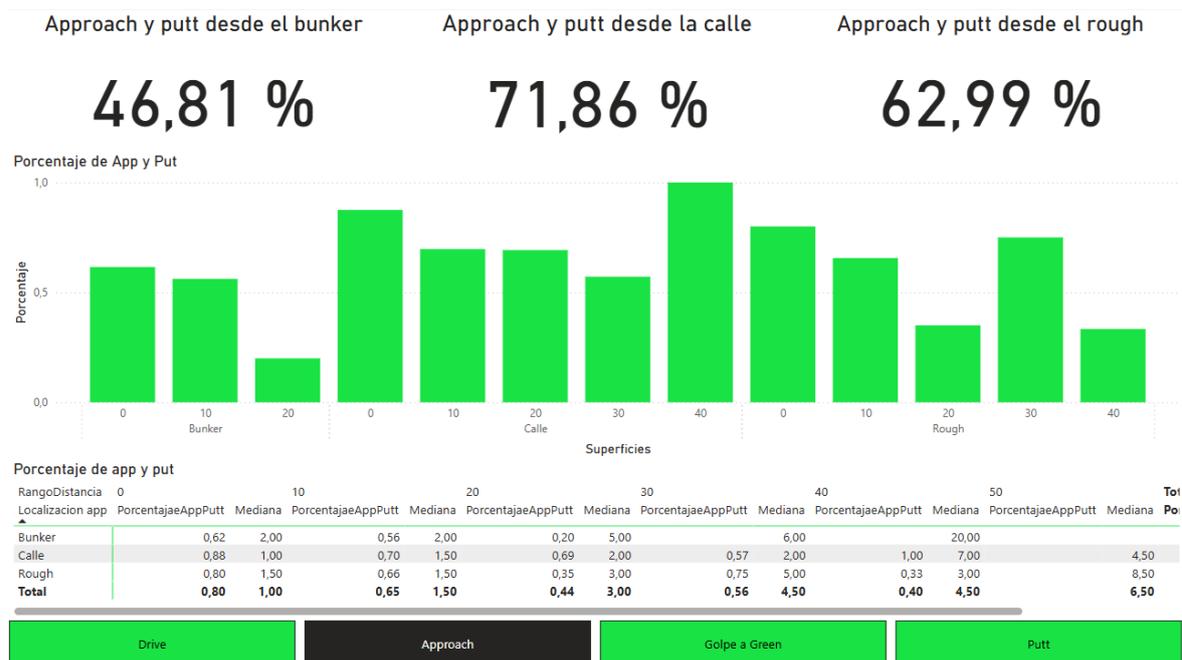


Figura 10: Panel de juego corto

En este caso, las KPIs seleccionadas fueron los porcentajes de *Approach* y *putt* desde las tres superficies posibles. Viendo los resultados, como primer consejo, la jugadora debería de practicar la salida de *Bunker* para intentar aumentar ese porcentaje, sobre todo en las

distancias entre treinta y cincuenta metros, puesto que ahí el porcentaje de *Approach* y *putt* es cero por ciento.

Este panel (*Figura 10*) cuenta con la información del porcentaje de *Approach* y *putt* que hace desde las diferentes superficies y en diferentes rangos de distancia y en la tabla también cuenta con la información de la mediana de la distancia a la que la deja.

6.4 PANEL DE PUTT.

El panel del *putt* se puede ver en la siguiente imagen (*Figura 11*)



Figura 11: Panel de putt

En este panel (*Figura 11*) se puede ver toda la información referente al *putt*. Como en este caso las distancias son más cortas y deben de ser mucho más precisas, se tomó la decisión de meter un filtro en el que la jugadora pudiera seleccionar las distancias que quisiera estudiar.

A grandes rasgos el consejo que se le daría a la jugadora es que debe prestar atención al tema de las caídas, puesto que el porcentaje de acierto cuando hay caída en el *putt* es mucho más pequeño que cuando son rectos, además cabe destacar que normalmente le da más caída que la que tiene el *putt*, puesto que más de la mitad de ellos las falla por arriba.

Los *putts* que más debería de practicar son aquellos entre un metro y tres metros, puesto que son los que más tira.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Una vez realizado el proyecto y después de analizar detenidamente el trabajo realizado, las conclusiones que se extraen son las siguientes:

- La metodología que se utilizó para la recogida de datos, facilitó en gran medida la tarea de la jugadora, de tal manera que la experiencia fue fácil, sencilla y poco tediosa para ella.
- Se realizó una limpieza y estructuración de los datos, colocando cada una de las tablas en su Excel, de tal manera que luego todo estuviera ordenado y no hubiera posibilidad de mezclar los datos. De tal manera que se garantizó así la coherencia y calidad de los datos para su posterior análisis.
- Se desarrolló un código funcional para llevar a cabo el análisis estadístico, permitiendo extraer la información relevante y útil para la jugadora.
- Finalmente se creó el dashboard, para que la jugadora pudiera ver la información extraída de forma clara, para poder plantear sus entrenamientos e ir mejorando en cada entrenamiento.

Como conclusión global, se ha realizado un trabajo que como resultado ha dado una herramienta integral que cubre todo el ciclo de los datos, desde la extracción hasta la visualización, cumpliendo con todos los objetivos y aportando valor a su destinataria final.

Como trabajos futuros, para seguir con el desarrollo del proyecto se podrían realizar las siguientes mejoras:

- Automatizar la recogida de los datos. Lo ideal sería que la jugadora no tuviera que proporcionar los datos y que se realizara mediante un sistema automático, como un modelo que a través de un video fuera capaz de detectar todos los campos necesarios para luego su posterior análisis. Aunque esto es difícil de conseguir, primero por que el deporte femenino no tiene mucha retransmisión y además se necesitaría todo el

recorrido y segundo porque habría que desarrollar un modelo que fuera capaz de realizar ese análisis.

- Ampliación de análisis estadístico. Se podría complementar con técnicas más avanzadas a la vez que se meten más variables para estudiar.
- Ampliación del dashboard. Se podrían añadir opciones para que la jugadora pudiera añadir diferentes graficas según lo que quisiera estudiar en ese momento.
- Desarrollo de una aplicación móvil o web. Lo ideal sería que la jugadora tuviera acceso desde cualquier lugar del mundo y en cualquier dispositivo electrónico, para poder ver que debe de mejorar.

Capítulo 8. BIBLIOGRAFÍA

- [1] B. Kelley, “Origins of Golf: When and Where Did Golf Begin?,” liveaboutdotcom. Accessed: Apr. 24, 2022. [Online]. Available: <https://www.liveabout.com/when-and-where-did-golf-begin-1561081>
- [2] K. Evans and N. Tuttle, “Improving performance in golf: current research and implications from a clinical perspective,” *Braz J Phys Ther*, vol. 19, no. 5, p. 381, Sep. 2015, doi: 10.1590/BJPT-RBF.2014.0122.
- [3] G. D. Wells, M. Elmi, and S. Thomas, “Physiological correlates of golf performance,” *J Strength Cond Res*, vol. 23, no. 3, pp. 741–750, May 2009, doi: 10.1519/JSC.0B013E3181A07970.
- [4] “Optimize Your Golf Game with Advanced Analytics.” Accessed: Apr. 04, 2025. [Online]. Available: <https://www.datanami.com/2017/08/08/optimize-golf-game-advanced-analytics/>
- [5] “Traditional Statistics V Strokes Gained.”
- [6] M. Broadie, “Assessing Golfer Performance Using Golfmetrics.”
- [7] “Data analytics in golf: How a revolution in preparation is changing the sport | IBTimes UK.” Accessed: Apr. 04, 2025. [Online]. Available: <https://www.ibtimes.co.uk/data-analytics-golf-how-revolution-preparation-changing-sport-1563932>
- [8] M. Broadie, “Assessing golfer performance on the PGA tour,” *Interfaces (Providence)*, vol. 42, no. 2, pp. 146–165, Mar. 2012, doi: 10.1287/inte.1120.0626.
- [9] “The Increasing Presence Of Data Analytics In Golf | Sport Performance Analysis.” Accessed: Apr. 04, 2025. [Online]. Available:

- <https://www.sportperformanceanalysis.com/article/increasing-presence-of-data-analytics-in-golf>
- [10] “Data Golf is taking analytics to a whole new level (pay attention, gamblers).” Accessed: Apr. 04, 2025. [Online]. Available: <https://golf.com/news/features/data-golf-analytics-new-level-pay-attention-gamblers/>
- [11] T. Suzuki, J. P. Sheahan, I. Okuda, and D. Ichikawa, “Investigating factors that improve golf scores by comparing statistics of amateur golfers in repeat scramble strokes and one-ball conditions,” *Journal of Human Sport and Exercise*, vol. 16, no. 4, pp. 1–13, 2021, doi: 10.14198/jhse.2021.164.09.
- [12] “How Dustin Johnson used data and analytics to become one of the best golfers in the world – GeekWire.” Accessed: Apr. 04, 2025. [Online]. Available: <https://www.geekwire.com/2016/how-dustin-johnson-used-data-to-rise-to-the-top-of-the-golf-world/>
- [13] N. James, “The Statistical Analysis of Golf Performance,” *Int J Sports Sci Coach*, vol. 2, no. 1_suppl, pp. 231–249, Jun. 2007, doi: 10.1260/174795407789705424.
- [14] Mark Broadie, *Every shot counts*, 1st ed., vol. 1. New York: Penguin Group, 2014.
- [15] “talent.com,”
<https://es.talent.com/salary?job=analista+de+datos#:~:text=%C2%BFCu%C3%A1n%20gana%20un%20Analista%20de,hasta%20%E2%82%AC%2040.190%20al%20a%C3%B1o.>
- [16] “Microsoft,” <https://www.microsoft.com/es-es/power-platform/products/power-bi/pricing>.

ANEXO I: CÓDIGO DE PYTHON

CÓDIGO JUEGO MEDIO.

- El porcentaje de *greenes* que cogía la jugadora según la distancia.

```
def golpe_a_green(distmin, distmax):
    # Filtrar el DataFrame global 'df_golf_golpe_green' para obtener solo las
    # filas
    # donde la 'Distancia a green' está entre 'distmin' (inclusive) y 'distmax'
    # (exclusivo)
    # y seleccionar únicamente las columnas 'Distancia a green', 'Green' y 'Lado'
    df = df_golf_golpe_green.loc[
        (df_golf_golpe_green['Distancia a green'] >= distmin) &
        (df_golf_golpe_green['Distancia a green'] < distmax),
        ['Distancia a green', 'Green', 'Lado']
    ]
    # Imprimir las primeras filas del DataFrame filtrado para verificar que la
    # selección es correcta
    print(df.head())
    # Contar la cantidad de ocurrencias para cada valor en la columna 'Green'
    # y ordenar los resultados en orden ascendente
    golpes_si = df.value_counts('Green').sort_values()
    # Calcular el total de golpes (filas) en el DataFrame filtrado
    total = len(df)
    # Calcular el porcentaje de golpes que llegaron al green (por cada valor en
    # 'Green')
    # Se divide el número de golpes por el total y se multiplica por 100
    porcentaje_green = golpes_si / total * 100
    # Retornar el porcentaje calculado
    return porcentaje_green
```

- La mediana y la media de la distancia a la que la deja del hoyo dependiendo de la distancia.

```
def resultado(distmin, distmax):
    # Filtrar el DataFrame global 'df_golf_golpe_green' para obtener únicamente
    # las filas
    # donde:
    # - 'Distancia a green' es mayor o igual a 'distmin' y menor a 'distmax'
    # - La columna 'Green' tiene el valor 'Si'
    # Se seleccionan solo las columnas 'Distancia a green' y 'Resultado'
    df = df_golf_golpe_green.loc[
        (df_golf_golpe_green['Distancia a green'] >= distmin) &
```

```
(df_golf_golpe_green['Distancia a green'] < distmax) &
(df_golf_golpe_green['Green'] == 'Si'),
['Distancia a green', 'Resultado']
]
# Calcular la mediana de la columna 'Resultado'
# Calcular la media (promedio) de la columna 'Resultado'
median = np.mean(df.Resultado)
# Retornar la mediana y la media como una tupla
return mediana, median

def resultado_sin_green(distmin, distmax):
    # Filtrar el DataFrame global 'df_golf_golpe_green' para seleccionar las
    # filas
    # donde la 'Distancia a green' está en el rango [distmin, distmax)
    # y se seleccionan solo las columnas 'Distancia a green' y 'Resultado'
    df = df_golf_golpe_green.loc[
        (df_golf_golpe_green['Distancia a green'] >= distmin) &
        (df_golf_golpe_green['Distancia a green'] < distmax),
        ['Distancia a green', 'Resultado']
    ]
    # Calcular la mediana de la columna 'Resultado'
    mediana = np.median(df.Resultado)
    # Calcular la media (promedio) de la columna 'Resultado'
    median = np.mean(df.Resultado)
    # Retornar la mediana y la media en forma de tupla
    return mediana, median
```

- El porcentaje de golpes que daba según la distancia a la que se encontraba, por ejemplo, cuál era el porcentaje que daba entre veinte y treinta metros.

```
def porcentaje_golpes(distmin, distmax):
    # Filtrar el DataFrame global 'df_golf_golpe_green' para seleccionar las
    # filas
    # donde 'Distancia a green' se encuentra en el rango [distmin, distmax)
    df = df_golf_golpe_green.loc[
        (df_golf_golpe_green['Distancia a green'] >= distmin) &
        (df_golf_golpe_green['Distancia a green'] < distmax)
    ]
    # Calcular la cantidad de golpes dentro del rango especificado
    longitud = len(df)
    # Calcular el total de golpes registrados en el DataFrame global
    longitud_total = len(df_golf_golpe_green)
    # Calcular el porcentaje de golpes en el rango respecto al total
    porcentaje_golpes_dist = longitud / longitud_total * 100
    # Retornar el porcentaje calculado
    return porcentaje_golpes_dist
```

- De los *greenes* que fallaba, había que calcular el porcentaje que fallaba por el lado bueno.

```
def golpe_lado(distmin, distmax):
    # Filtrar el DataFrame global 'df_golf_golpe_green' para seleccionar filas
    # donde 'Distancia a green' esté entre 'distmin' (inclusive) y 'distmax'
    (exclusivo)
    # y seleccionar las columnas 'Distancia a green', 'Green' y 'Lado'
    df = df_golf_golpe_green.loc[
        (df_golf_golpe_green['Distancia a green'] >= distmin) &
        (df_golf_golpe_green['Distancia a green'] < distmax),
        ['Distancia a green', 'Green', 'Lado']
    ]
    # Calcular la cantidad de ocurrencias para cada valor en la columna 'Lado'
    # y ordenar los resultados en orden ascendente
    lado = df.value_counts('Lado').sort_values()
    # Determinar el total de golpes contabilizados en el rango de 'Lado'
    # Si solo hay un valor único en 'Lado'
    if len(lado) == 1:
        total_lado = df.value_counts('Lado').sort_values()[0]
    # Si hay dos valores diferentes en 'Lado'
    if len(lado) == 2:
        total_lado = df.value_counts('Lado').sort_values()[0] +
df.value_counts('Lado').sort_values()[1]
    # Calcular el porcentaje que representa cada valor de 'Lado' respecto al
    total
    porcentaje_lado = lado / total_lado * 100
    # Retornar el porcentaje calculado
    return porcentaje_lado
```

CÓDIGO DEL JUEGO CORTO.

- Porcentaje de veces que era capaz de hacer un “*up and down*” para un rango de distancia y en una superficie determinada.

```
def up_and_down(distmin, distmax, loc):
    # Filtrar el DataFrame global 'df_golf_approach' para seleccionar las filas
    donde:
    # - 'Distancia app' se encuentra en el rango [distmin, distmax)
    # - 'Localizacion app' es igual a 'loc'
    # Se selecciona únicamente la columna "App y putt"
    df = df_golf_approach.loc[
        (df_golf_approach['Distancia app'] >= distmin) &
        (df_golf_approach['Distancia app'] < distmax) &
        (df_golf_approach['Localizacion app'] == loc),
        ["App y putt"]
    ]
```

```
# Calcular la cantidad de ocurrencias para cada valor en la columna "App y putt"
up_and_down = df.value_counts("App y putt")
# Calcular el total de registros en el DataFrame filtrado
total = len(df)
# Calcular el porcentaje que representa cada valor de "App y putt" respecto al total
porcentaje = up_and_down / total * 100
# Retornar el porcentaje calculado
return porcentaje
```

- La mediana y la media de la distancia a la que la deja del hoyo dependiendo de la distancia.

```
def resultado(distmin, distmax, loc):
    # Filtrar el DataFrame global 'df_golf_approach' para seleccionar las filas donde:
    # - 'Distancia app' se encuentra en el rango [distmin, distmax)
    # - 'Localizacion app' es igual a 'loc'
    # Se seleccionan únicamente las columnas 'Distancia app' y 'm result'
    df = df_golf_approach.loc[
        (df_golf_approach['Distancia app'] >= distmin) &
        (df_golf_approach['Distancia app'] < distmax) &
        (df_golf_approach['Localizacion app'] == loc),
        ['Distancia app', 'm result']
    ]
    # Calcular la mediana de la columna 'm result'
    mediana = np.median(df['m result'])
    # Calcular la media de la columna 'm result'
    media = np.mean(df['m result'])
    # Retornar la mediana y la media en forma de tupla
    return mediana, media
```

CÓDIGO DEL PUTT.

- Porcentaje de acierto según distancia.

```
def acierto(distmin, distmax):
    # Filtrar el DataFrame global 'df_golf' para seleccionar filas donde:
    # - 'Distancia' es mayor que 'distmin' y menor o igual a 'distmax'
    # Se seleccionan únicamente las columnas 'Embocado', 'Resultado 2' y 'Caida'
    df = df_golf.loc[
        (df_golf.Distancia > distmin) & (df_golf.Distancia <= distmax),
        ['Embocado', 'Resultado 2', 'Caida']
    ]
```

```
# Contar la cantidad de ocurrencias en la columna 'Embocado' y ordenar los resultados
# Esto nos dará la cantidad de veces que se obtuvo cada valor en 'Embocado'
embocado = df.value_counts('Embocado').sort_values()
# Calcular el total de golpes (filas) en el DataFrame filtrado
tirados = len(df)
# Calcular el porcentaje de golpes embocados respecto al total de golpes
# Se multiplica por 100 para obtener el valor en porcentaje
acierto = embocado / tirados * 100
# Retornar el porcentaje calculado
return acierto
```

- Porcentaje de *putt* tirados para los diferentes tipos de caída según un rango de distancia.

```
def caida_tirados(distmin, distmax):
# Filtrar el DataFrame global 'df_golf' para seleccionar filas donde:
# - 'Distancia' es mayor que 'distmin' y menor o igual a 'distmax'
# Se seleccionan las columnas 'Embocado', 'Resultado 2' y 'Caída'
df = df_golf.loc[
    (df_golf.Distancia > distmin) & (df_golf.Distancia <= distmax),
    ['Embocado', 'Resultado 2', 'Caída']
]
# Calcular el total de registros (filas) en el DataFrame filtrado
total = len(df)
# Contar las ocurrencias para cada valor en la columna 'Caída' y ordenarlas
caida = df.value_counts('Caída').sort_values()
# Calcular el porcentaje de cada valor de 'Caída' respecto al total
# Multiplicamos por 100 para obtener el valor en porcentaje
tirados = caida / total * 100
# Retornar el porcentaje calculado para cada categoría de 'Caída'
return tirados
```

- Porcentaje de aciertos según el tipo de caída que tenía el *putt* y teniendo en cuenta la distancia del *putt*.

```
# Convertir los valores de la columna 'Distancia' de df_golf:
# Reemplaza las comas por puntos y convierte la cadena resultante en un número de punto flotante (float)
df_golf['Distancia'] = df_golf['Distancia'].str.replace(',', '.').astype(float)
def acierto_caída(distmin, distmax, caída):
# Filtrar el DataFrame global 'df_golf' para obtener las filas donde:
# - 'Distancia' es mayor que 'distmin' y menor o igual a 'distmax'
# - La columna 'Caída' es igual al valor indicado en el parámetro 'caída'
# Se seleccionan las columnas 'Embocado', 'Resultado 2' y 'Caída'
df = df_golf.loc[
```

```
(df_golf['Distancia'] > distmin) &
(df_golf['Distancia'] <= distmax) &
(df_golf['Caida'] == caida),
['Embocado', 'Resultado 2', 'Caida']
]
# Contar la cantidad de ocurrencias para cada valor en la columna 'Embocado'
# y ordenar estos conteos de forma ascendente
emb_caida = df.value_counts('Embocado').sort_values()
# Calcular el porcentaje de golpes embocados respecto al total de registros
filtrados
# Multiplica por 100 para obtener el resultado en porcentaje
# Retornar el porcentaje calculado
return acierto_caida
```

- Porcentaje de falladas por arriba según la distancia a la que tiraba el *putt*.

```
def falladas_arriba(distmin, distmax):
# Filtrar el DataFrame global 'df_golf' para obtener las filas donde:
# - La columna 'Distancia' es mayor que 'distmin' y menor o igual a 'distmax'
# Se seleccionan únicamente las columnas 'Embocado', 'Resultado 2' y 'Caida'
df = df_golf.loc[
    (df_golf.Distancia > distmin) & (df_golf.Distancia <= distmax),
    ['Embocado', 'Resultado 2', 'Caida']
]
# Dentro del DataFrame filtrado, seleccionar solo las filas donde el valor en
'Embocado' es 'NO'
# Esto indica que se trata de golpes que no han embocado
df_arriba = df.loc[df['Embocado'] == 'NO', ['Embocado', 'Resultado 2',
'Caida']]
# Contar las ocurrencias de cada valor en la columna 'Resultado 2'
# y ordenar los resultados de forma ascendente.
# Esto nos muestra cuántas falladas hay clasificadas por cada resultado en
'Resultado 2'
falladas_arriba = df_arriba.value_counts('Resultado 2').sort_values()
# Calcular el total de golpes fallados (no embocados) contando el número de
filas en df_arriba
total_falladas = len(df_arriba)
# Calcular el porcentaje de falladas para cada categoría de 'Resultado 2'
# Dividiendo el número de ocurrencias por el total y multiplicando por 100
para obtener el porcentaje
porcentaje_falladas_arriba = falladas_arriba / total_falladas * 100
# Retornar el porcentaje calculado de falladas (por cada categoría de
'Resultado 2')
return porcentaje_falladas_arriba
```

ANEXO II: CÓDIGO DE POWER BI

MODELADO DE DATOS DEL JUEGO LARGO.

- Completar las casillas vacías con la palabra *Calle*

```
= Table.ReplaceValue("#Changed Type1", "", "Calle", Replacer.ReplaceValue, {"Lado (drive)"}))  
= Table.ReplaceValue("#Replaced Value", "", "Calle", Replacer.ReplaceValue, {"Posicion"})
```

- PorcentajeBueno, donde se calculaba el porcentaje de *calles* que había fallado por el lado bueno.

```
PorcentajeBueno =  
DIVIDE (COUNTROWS (FILTER ('EstadisticasDrive2023', 'EstadisticasDrive2023' [Lado (drive)] = "Bueno")), COUNTROWS ('EstadisticasDrive2023'))
```

- PorcentajeMalo, donde se calculaba el porcentaje de *calles* que había fallado por el lado malo.

```
PorcentajeMalo =  
DIVIDE (COUNTROWS (FILTER ('EstadisticasDrive2023', 'EstadisticasDrive2023' [Lado (drive)] = "Malo")), COUNTROWS ('EstadisticasDrive2023'))
```

- PorcentajeCalle, donde se calculaba el porcentaje de *calles* cogidas por la jugadora.

```
PorcentajeCalle =  
DIVIDE (COUNTROWS (FILTER ('EstadisticasDrive2023', 'EstadisticasDrive2023' [Calle] = "Si")), COUNTROWS ('EstadisticasDrive2023'))
```

MODELADO DE DATOS DEL JUEGO MEDIO.

- Cambio de tipo de dato de la columna Distancia a *green*.

```
= Table.TransformColumnTypes("#Promoted Headers",{{"Distancia a green",  
Int64.Type}})
```

- Sustituir los puntos por comas en la columna de Resultado

```
= Table.ReplaceValue("#Changed Type",".",",",Replacer.ReplaceText,{"Resultado"})  
= Table.TransformColumnTypes("#Replaced Value",{{"Resultado", type number}})
```

- Calcular el rango de distancia de 10m en 10m y cambiar a tipo entero.

```
=Table.AddColumn("#Changed Type1", "RangoDistancia", each  
Number.RoundDown([Distancia a green]/10)*10)  
= Table.TransformColumnTypes("#Added Custom",{{"RangoDistancia", Int64.Type}})
```

- Duplicar la columna

```
= Table.TransformColumnTypes("#Duplicated Column",{{"RangoDistancia - Copy", type  
text}})
```

Al igual que en el modelado del juego largo, se tuvo que añadir datos en la columna de Lado, ya que la jugadora en caso de coger *green* la dejaba vacía. Para ello se utilizó la siguiente fórmula:

- Añadir la palabra *green* en la casilla donde la jugadora había cogido *green* y la había dejado vacía

```
= Table.ReplaceValue("#Sorted Rows","", "Green", Replacer.ReplaceValue, {"Lado"})
```

- Cambio del nombre de las columnas

```
= Table.RenameColumns("#Replaced Value1",{{"RangoDistancia", "RangoDistancia1"},  
{"RangoDistancia - Copy", "RangoDistancia"}})
```

- *PorcentajeGreen*, en la cual se calculaba el porcentaje de *greenes* que cogía la jugadora.

```
PorcentajeGreen =
DIVIDE (CALCULATE (COUNTROWS ('EstadisticasGolpeGreen2023'), 'EstadisticasGolpeGreen2023' [Green]="Si"),
CALCULATE (COUNTROWS ('EstadisticasGolpeGreen2023'), 'EstadisticasGolpeGreen2023' [Green]<>BLANK ()))
```

MODELADO DE DATOS DEL JUEGO CORTO.

- Cambiar tipo de dato en la columna *Distancia app*

```
= Table.TransformColumnTypes("#Added Custom",{"Distancia app", Int64.Type}))
```

- Calculo del rango de distancia

```
= Table.AddColumn("#Removed Columns", "RangoDistancia", each
Number.RoundDown([Distancia app]/10)*10)
= Table.TransformColumnTypes("#Replaced Value",{"m result", type number},
{"RangoDistancia", Int64.Type}))
```

- Sustitución de los puntos por comas y modificación del tipo a number en la columna *m result*

```
= Table.ReplaceValue("#Added Custom1", ".", ",", Replacer.ReplaceText, {"m result"})
= Table.TransformColumnTypes("#Replaced Value", {"m result", type number},
{"RangoDistancia", Int64.Type}))
```

- Duplicó la columna *RangoDistancia* y se puso de tipo texto

```
= Table.DuplicateColumn("#Removed Columns1", "RangoDistancia", "RangoDistancia - Copy")
= Table.TransformColumnTypes("#Duplicated Column1", {"RangoDistancia - Copy", type text}, {"App y putt", type text}, {"Localizacion app", type text}))
```

- *PorcentajeAppPutt*, donde se calculaba el porcentaje de *aprouch* y *putt*.

```
PorcentajaeAppPutt =  
DIVIDE (CALCULATE (COUNTROWS ('EstadisticasApproach2023'), 'EstadisticasApproach2023' [A  
pp y putt]="Si"),  
CALCULATE (COUNTROWS ('EstadisticasApproach2023'), 'EstadisticasApproach2023' [App y  
putt]<>BLANK ()))
```

- *AppPutBunker*; donde se calculaba el porcentaje de approach y putt desde el bunker.

```
AppPuttBunker= divide (COUNTROWS (FILTER ('EstadisticasApproach2023',  
'EstadisticasApproach2023' [App y putt]="Si" &&  
'EstadisticasApproach2023' [Localizacion app]="Bunker" )),  
COUNTROWS (FILTER ('EstadisticasApproach2023', EstadisticasApproach2023 [Localizacion  
app]="Bunker")))
```

- *AppPutCalle*, donde se calculaba el porcentaje de approach y putt desde la calle.

```
AppPuttCalle = divide (COUNTROWS (FILTER ('EstadisticasApproach2023',  
'EstadisticasApproach2023' [App y putt]="Si" &&  
'EstadisticasApproach2023' [Localizacion app]="Calle" )),  
COUNTROWS (FILTER ('EstadisticasApproach2023', EstadisticasApproach2023 [Localizacion  
app]="Calle")))
```

- *AppPutRough*, donde se calculaba el porcentaje de approach y putt desde el rough

```
AppPuttRough = divide (COUNTROWS (FILTER ('EstadisticasApproach2023',  
'EstadisticasApproach2023' [App y putt]="Si" &&  
'EstadisticasApproach2023' [Localizacion app]="Rough" )),  
COUNTROWS (FILTER ('EstadisticasApproach2023', EstadisticasApproach2023 [Localizacion  
app]="Rough")))
```

- *MedianaPorRangoSuperficie*, donde se calculaba la mediana del resultado obtenido después de dar el golpe.

```
MedianaPorRangoSuperficie = MEDIANX (FILTER ('EstadisticasApproach2023',  
NOT (ISBLANK ('EstadisticasApproach2023' [m result])), 'EstadisticasApproach2023' [m  
result])
```

MODELADO DE DATOS DEL PUTT.

- Sustituir los puntos por comas en la columna de Distancia

```
= Table.ReplaceValue("#Promoted  
Headers", ".", ",", Replacer.ReplaceText, {"Distancia"})  
= Table.TransformColumnTypes("#Replaced Value", {"Distancia", type number})
```

- Calcular el rango de distancia

```
= Table.AddColumn("#Changed Type", "RangoDistancia", each if [Distancia] <= 1  
then "1m" else if [Distancia] <= 2 then "2m" else if [Distancia] <= 3 then "3m"  
else if [Distancia] <= 4 then "4m" else if [Distancia] <= 5 then "5m" else if  
[Distancia] <= 6 then "6m" else if [Distancia] <= 7 then "7m" else if [Distancia]  
<= 10 then "10m" else "Mayor de 10m")
```

- Se crea una columna para ordenar los rangos

```
= Table.AddColumn("#Sorted Rows", "Orden ", each if [RangoDistancia] = "1m" then  
1 else if [RangoDistancia] = "2m" then 2 else if [RangoDistancia] = "3m" then 3  
else if [RangoDistancia] = "4m" then 4 else if [RangoDistancia] = "5m" then 5  
else if [RangoDistancia] = "6m" then 6 else if [RangoDistancia] = "7m" then 7  
else if [RangoDistancia] = "10m" then 8 else 9)
```

- Acierto, donde se ha calculado el porcentaje de acierto

```
Acierto =  
DIVIDE(CALCULATE(COUNTROWS('EstadisticasPutt2023'), 'EstadisticasPutt2023'[Embocad  
o]="SI"),  
CALCULATE(COUNTROWS('EstadisticasPutt2023'), 'EstadisticasPutt2023'[Embocado]<>BLA  
NK()))
```

- CaidaDerIzq, donde se ha calculado el porcentaje de *putts* tirados de derecha a izquierda.

```
CaidaDerIzq =  
DIVIDE(CALCULATE(COUNTROWS('EstadisticasPutt2023'), 'EstadisticasPutt2023'[Caida]=  
"DER-IZQ"),  
CALCULATE(COUNTROWS('EstadisticasPutt2023'), 'EstadisticasPutt2023'[Caida]<>BLANK(  
)))
```

- CaidaIzqDer, donde se ha calculado el porcentaje de *putts* tirados de izquierda a derecha.

```
CaidaIzqDer =  
DIVIDE (CALCULATE (COUNTROWS ('EstadisticasPutt2023'), 'EstadisticasPutt2023' [Caida]=  
"IZQ-DER"),  
CALCULATE (COUNTROWS ('EstadisticasPutt2023'), 'EstadisticasPutt2023' [Caida]<>BLANK (  
)))
```

- CaidaRecto, donde se ha calculado el porcentaje de *putts* tirados rectos.

```
CaidaRecta =  
DIVIDE (CALCULATE (COUNTROWS ('EstadisticasPutt2023'), 'EstadisticasPutt2023' [Caida]=  
"RECTO"),  
CALCULATE (COUNTROWS ('EstadisticasPutt2023'), 'EstadisticasPutt2023' [Caida]<>BLANK (  
)))
```

- FalladasArriba, donde se ha calculado el porcentaje de *putts* fallados por arriba.

```
FalladasArriba =  
DIVIDE (CALCULATE (COUNTROWS ('EstadisticasPutt2023'), 'EstadisticasPutt2023' [Embocad  
o]="NO" && EstadisticasPutt2023[Resultado 2]="ARRIBA"),  
CALCULATE (COUNTROWS ('EstadisticasPutt2023'), 'EstadisticasPutt2023' [Resultado  
2]<>BLANK ()))
```