



Escuela Técnica Superior
de Ingeniería Informática

Grado en Ingeniería de Computadores

Curso 2021-2022

Trabajo Fin de Grado

**COMPARATIVA ENTRE LAS API DE SPARK EN
SCALA Y PYTHON**

Autor: Oscar Nydza Nicpoñ

Tutor: Juan Manuel Serrano Hidalgo

Agradecimientos

Breves agradecimientos o dedicatoria.

Resumen

Breve resumen del Trabajo de Fin de Grado (TFG). Recomendable entre 250-300 palabras, conteniendo los principales objetivos y resultados derivados del mismo.

Palabras clave:

- Python
- Ciberseguridad
- Aprendizaje automático (pueden ser varias)
- ...

Índice de contenidos

Índice de figuras	IX
Índice de códigos	XI
1. Introducción	XIII
1.1. Contexto y alcance	1
1.2. Estructura del documento	1
1.2.1. Trabajos de grados en informática	1
1.2.2. Trabajos del grado en matemáticas	2
2. Objetivos	3
3. Descripción Informática	5
3.1. Fuentes de datos	6
3.1.1. Tablas de entrada	6
3.2. Programación de queries en PySpark	9
3.3. Programación de queries en Scala/Spark	9
3.3.1. Piloto más consistente de la temporada 2012	9
3.3.2. Dominio de fabricantes en la década de los 90	11
3.4. Despliegue en AWS EMR	12
4. Experimentos / Validación	13
4.1. Análisis de requisitos no funcionales	14
5. Conclusiones y trabajos futuros	15
5.1. Texto de relleno	16
Bibliografía	21
Apéndices	23
A. Este es el primer apéndice	25
A.1. Ejemplo de sección	25

Índice de figuras

Índice de códigos

1

Introducción

Se puede añadir texto antes de empezar la primera sección.

1.1. Contexto y alcance

Contexto. Situar al lector. Objetivo general y alcance del trabajo.

1.2. Estructura del documento

La estructura del TFG no es fija. El tutor indicará una estructura adecuada dependiendo del trabajo concreto.

Se puede incluir dentro de cada apartado secciones adicionales. La copia en papel de la memoria del TFG será encuadernada en pasta dura de color azul (p.e. encuadernación tipo chanel). La portada, que puede ser una pegatina transparente, seguirá el modelo que se adjunta, que incluye el escudo y nombre de la URJC, la titulación cursada por el alumno, el curso académico, el título del TFG, el autor y el o los directores/tutores.

1.2.1. Trabajos de grados en informática

Una posible estructura de la memoria final asociada con cada TFG podría ser la siguiente (leed la normativa de TFG):

1. Introducción
2. Objetivos (incluyendo descripción del problema, estudio de alternativas y metodología empleada)
3. Descripción informática (puede incluir especificación, diseño, implementación y pruebas).
4. Experimentos / validación
5. Conclusiones (incluyendo los logros principales alcanzados y posibles trabajos futuros)
6. Bibliografía
7. Apéndices

1.2.2. Trabajos del grado en matemáticas

Una posible estructura de la memoria final asociada con cada TFG podría ser la siguiente:

1. Introducción
2. Objetivos (incluyendo descripción del problema, estudio de alternativas y metodología empleada)
3. Material y métodos / Metodología / Cuerpo del trabajo (describir las metodologías empleadas en el desarrollo del TFG o el desarrollo del mismo en caso de ser un trabajo de recopilación bibliográfica sobre un tema).
4. Resultados (opcional, dependiendo del tipo de trabajo desarrollado)
5. Conclusiones (incluyendo los logros principales alcanzados y posibles trabajos futuros)
6. Bibliografía
7. Apéndices

2

Objetivos

El principal objetivo de este Trabajo de Fin de Grado realizar una comparativa entre las API de Spark de Scala y de Python. Sin embargo, también existen otros objetivos que irán surgiendo mientras se avanza sobre el principal. Entre ellos estarían:

1. Aprender Scala como lenguaje funcional.
2. Aprender Spark como API tanto para Scala como para Python.
3. Visualizar de los resultados de las queries realizadas usando Plotly.
4. Migrar queries desde PySpark a Scala Spark.
5. Medir y comparar el rendimiento de ambas API.
6. Realizar queries a un cluster AWS EMR.

Como objetivo adicional y debido a que se usará un dataset de la Fórmula 1, se buscará encontrar la siguiente información:

- Piloto más consistente en un periodo de tiempo concreto.
- Piloto más dominante a lo largo de su carrera.
- Fabricante más dominante en la década de los 90.
- En base a lo anterior, cuál ha sido el peor año de esa marca en ese periodo de tiempo.
- Fabricante con las mejores paradas a boxes en la década de los 2010.
- Fabricante con más problemas de fiabilidad en la década de los 2000.
- Temporada más interesante para el espectador.

Además también se buscará hacer análisis de una temporada en concreta, de un piloto o de un fabricante sacando distintas métricas para cada uno de ellos y siendo capaces de visualizarlos gráficamente.

3

Descripción Informática

3.1. Fuentes de datos

Como se mencionó brevemente en el apartado de Objetivos, se ha utilizado un conjunto de datos de la Fórmula 1. Concretamente, este dataset tiene 13 tablas que proporcionan información sobre distintos aspectos de esta competición. Estas tablas son:

- `circuits`
- `constructor_results`
- `constructor_standings`
- `constructors`
- `driver_standings`
- `lap_times`
- `pit_stops`
- `qualifying`
- `races`
- `results`
- `seasons`
- `status`
- `drivers`

3.1.1. Tablas de entrada

En este apartado se proporcionará una descripción de los datos más útiles que contiene cada una de las tablas de entrada.

Tabla `circuits`

Esta tabla contiene información sobre todos los circuitos en los que se ha llevado a cabo un Gran Premio. Las columnas más interesantes son el nombre del circuito, una referencia textual y la localización.

Tabla `constructor_results`

Esta tabla nos proporciona información sobre los resultados de las carreras en base a los constructores.

Tabla constructor_standings

Esta tabla contiene información sobre la clasificación de constructores. Como particularidad, tiene una entrada por carrera y constructor participante. Por tanto, podríamos ver cómo ha ido cambiando la clasificación de constructores a lo largo del campeonato.

Las columnas más interesantes son el identificador de la carrera, identificador del constructor, los puntos, la posición en la clasificación y las victorias hasta ese punto.

Tabla constructors

Esta tabla contiene información sobre los distintos constructores que han participado en algún campeonato mundial de Fórmula 1. Las columnas más interesantes son el id de constructor, la referencia, el nombre del constructor y la nacionalidad.

Tabla driver_standings

Similar a la tabla de clasificación de constructores, pero para pilotos. Tenemos las mismas columnas, salvo que en lugar de tener un id de constructor, lo tenemos de piloto.

Tabla lap_times

Esta tabla es una de las más interesantes, ya que nos da todos los tiempos de vuelta de todos los pilotos desde que hay registros. Esto es, desde parte de 1996 y 1997 al completo.

Las columnas más llamativas podrían ser el id de carrera, el de piloto, la vuelta en cuestión, la posición y el tiempo en milisegundos.

Tabla pit_stops

Esta tabla contiene información de las paradas en boxes. Las columnas más interesantes son los id de carrera y piloto, el índice de parada (si es la primera, segunda, etc), la vuelta en la que se hace y la duración en milisegundos.

Tabla qualifying

Esta tabla nos da información sobre los resultados de todas las rondas de clasificación. Las columnas más interesantes son la posición final y los tiempos en Q1, Q2 y Q3.

Tabla races

Esta tabla contiene información sobre todas las carreras celebradas en la historia de la competición. Contiene columnas como el id del circuito, el nombre del Gran Premio, la fecha y el año en el que se celebró. Esta última quizá sea la más útil de todo el dataset, ya que es la única forma de filtrar las carreras o los resultados por temporada.

Tabla results

Esta tabla es similar a la de resultados por constructor, pero para pilotos. Es la tabla más completa de todas, ya que nos proporciona una entrada por piloto y carrera con información relevante de cómo se ha desarrollado la misma. Las columnas más interesantes pueden ser la posición de salida y la posición final, los puntos, las vueltas dadas, la vuelta más rápida, la velocidad más rápida y, en el caso de que haya habido algún incidente, el id del estado.

Tabla seasons

Quizá se trate de la tabla menos útil, ya que solamente contiene una columna con el año y otra con una url a un artículo de Wikipedia para cada entrada.

Tabla status

Esta tabla nos da información sobre los estados en los que ha podido acabar la carrera un piloto determinado. Contiene un identificador y el estado en cuestión.

Tabla drivers

Contiene información sobre todos los pilotos que han competido a lo largo de la historia. En concreto la información más relevante puede ser el nombre y apellido, el código, la fecha de nacimiento y la nacionalidad.

3.2. Programación de queries en PySpark

3.3. Programación de queries en Scala/Spark

3.3.1. Piloto más consistente de la temporada 2012

En esta query intentaremos averiguar cuál ha sido el piloto más consistente de la temporada 2012. Ya que este término puede resultar ambigüo, en concreto intentaremos averiguar qué piloto tuvo una menor diferencia entre la media de sus vueltas rápidas del campeonato y la media de todas las vueltas de todos los Grandes Premios de la temporada.

Necesitaremos cruzar varias fuentes de datos para esto:

- `racers.csv`
- `lap_times.csv`
- `drivers.csv`
- `results.csv`

Para filtrar cualquier query por temporada o por rango de temporadas, tenemos que filtrar la tabla `racers` según la columna `year`. En nuestro caso, filtramos la tabla para la temporada 2012.

Una vez tenemos todas las carreras de la temporada en cuestión, necesitamos hallar el tiempo medio por vuelta de cada piloto en cada Gran Premio. En la tabla `lap_times.csv` existe una columna llamada `milliseconds`, que denota los milisegundos de la vuelta dada. Haciendo un `right join`, podemos obtener todas las vueltas dadas en una temporada. Usando una ventana que particione los datos por piloto, haremos la media de la columna `milliseconds` para hallar la media de toda la temporada.

Para obtener la media de las vueltas rápidas de cada piloto, debemos usar la tabla `results`, que tiene una entrada por cada piloto y Gran Premio que contiene además su vuelta más rápida según el formato `MM:ss.mmm`, siendo `MM` los minutos, `ss` los segundos y `mmm` los milisegundos. Para convertir este formato a milisegundos y viceversa, he creado dos UDFs llamadas `lapTimeToMs` y `msToLapTime`.

```

val lapTimeToMs = (time: String) => {
  val regex =
    """([0-9]|[0-9][0-9]):([0-9][0-9])\.([0-9][0-9][0-9])""".r
  time match {
    case regex(min,sec,ms) => min.toInt * 60 * 1000 + sec.toInt * 1000
      + ms.toInt
    case "\\N" => 180000
  }
}: Long

```

```

val msToLapTime = (time: Long) => {
  val mins = time / 60000
  val secs = (time - mins*60000)/1000
  val ms = time - mins*60000 - secs*1000

  val formattedSecs = if((secs / 10).toInt == 0) "0" + secs else secs
  // if ms = 00x -> "0"+"0"+x . if ms = 0xx -> "0"+ms
  val formattedMs = if((ms / 100).toInt == 0) "0" + (if((ms /
    10).toInt == 0) "0" + ms else ms) else ms
  mins + ":" + formattedSecs + "." + formattedMs
}: String

```

La función `lapTimeToMs` debe tener en cuenta también que hay entradas que no tienen un tiempo válido, sino que contienen el string `"\\N"`. Esto se debe a que hay ocasiones en las que un piloto no termina la vuelta. Un ejemplo sería si se retirase de la carrera en esa vuelta. Se ha decidido sumar 3 minutos (180000 ms) como penalización.

Una vez tenemos estas dos funciones, ya podemos el tiempo medio de vueltas rápidas. De nuevo particionamos los datos según el piloto y hacemos la media de la columna `fastestLapTime`. Este valor en milisegundos se convierte al formato de tiempo de vuelta con la UDF correspondiente y tras esto se ordena por el diferencial hallado de forma ascendente.

Además de esto, haciendo un análisis previo se observó que había pilotos que habían completado pocas carreras en comparación al resto, ya que a que sustituían a algún otro piloto debido a algún accidente. Se decidió filtrar estos outliers de forma que, independientemente del periodo de tiempo sobre el que queramos lanzar esta query, el resultado fuese significativo. Para ello se usó la tabla obtenida con todas las carreras en la temporada y, a partir de la columna `lap` de la tabla `lap_times` se realizó un conteo de todas las vueltas dadas por cada piloto utilizando, de nuevo, una ventana de datos particionada por piloto.

De este DataFrame queremos obtener tanto la cuenta de vueltas por piloto como el número total de pilotos que han participado en la temporada, que haremos

de la siguiente manera:

```
val (distinctDrivers, allLaps) = lapCount
    .agg(
        countDistinct("driverID"),
        count(col("lap"))
    ).as[(BigInt, BigInt)]
    .collect()(0)

val avgLapsThisPeriod = allLaps.toInt / distinctDrivers.toInt
```

Habiendo obtenido la media de vueltas por piloto de la temporada, podemos filtrar aquellos cuyo conteo sea inferior.

Solo quedaría formatear los datos de salida. Para ello primero se hace un join con la tabla de pilotos, lo cual nos permite concatenar nombre y apellido de cada uno de los participantes en una nueva columna. En esta caso, como solo nos interesa el piloto y su diferencial, estas serán las únicas columnas que dejaremos en el DataFrame.

3.3.2. Dominio de fabricantes en la década de los 90

Con esta query se pretende hallar qué fabricante ha sido el más dominante en la década de los 90. En concreto intentaremos hallar el número de mundiales ganados y el número de carreras ganadas.

Se usarán las siguientes fuentes de datos:

- `racess.csv`
- `constructor_standings.csv`
- `constructors.csv`

Al igual que en la query anterior, si queremos fijar nuestra atención en un periodo de tiempo, tenemos que hacerlo filtrando la columna `year` de la tabla `racess.csv`. En este caso, necesitamos todas las carreras entre el año 1990 y el año 1999.

Una vez obtenidas todas las carreras de la década, tenemos que obtener la última carrera de cada temporada. Esto es debido a que en `constructor_standings.csv` tenemos la clasificación resultante al final de cada carrera. Para ello, usaremos crearemos una ventana en la que particionaremos los datos por año y que usaremos con la función `max()` sobre la columna `round`, que nos indica el índice de la carrera, es decir, la primera carrera de la temporada tendrá `round === 1`, para crear una columna llamada `max` en la que guardaremos el índice de la última carrera de la temporada. Finalmente, filtraremos los datos

para quedarnos con aquellos en los que la columna `round == max`.

Tras esto último, unimos las tabla `constructor_standings.csv` con la recién obtenida para quedarnos con los resultados en las últimas carreras y filtramos según la columna `position == 1` para quedarnos con los ganadores. Teniendo esto, podemos ver también que la columna `wins` nos proporciona el número de victorias de cada escudería en esa temporada, así que, creando una ventana en la que particionemos por fabricante podemos hallar tanto la suma de victorias como el conteo de apariciones de cada una.

Pasando ya a la presentación de los datos, se filtrarían los constructores duplicados y se ordenarían los datos según el total de campeonatos ganados primero y, en caso de empate, por número de victorias. Además, se hace un join con la tabla de constructores para obtener su nombre.

Lo interesante de esta query es que se puede usar para cualquier periodo de tiempo. Podemos averiguar por ejemplo qué fabricante ha sido el más dominante en toda la historia de la competición y qué constructor ha dominado ciertos años concretos.

3.4. Despliegue en AWS EMR

4

Experimentos / Validación

4.1. Análisis de requisitos no funcionales

5

Conclusiones y trabajos futuros

En este capítulo se detallan las conclusiones derivadas del TFG y la propuesta de posibles trabajos futuros.

Las citas del texto Autor [1], Autor [2], Autor [3], Autor [4] y Autor [5] deben ir referenciadas en la bibliografía.

5.1. Texto de relleno

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc

vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisi. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit

ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus

quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Bibliografía

- [1] M. Giaquinta and S. Hildebrandt, *Calculus of variations II*. Springer Science and Business Media, 2013, vol. 311.
- [2] S. Fortune and C. J. Van Wyk, “Efficient exact arithmetic for computational geometry,” in *Proceedings of the Ninth Annual Symposium on Computational Geometry*, 1993, pp. 163–172.
- [3] S. Fortune, “Voronoi diagrams and delaunay triangulations,” *Computing in Euclidean geometry*, pp. 225–265, 1995.
- [4] J. C. Mitchell, “Social networks,” *Annual review of anthropology*, vol. 3, no. 1, pp. 279–299, 1974.
- [5] C. B. Morrey Jr, *Multiple integrals in the calculus of variations*. Springer Science and Business Media, 2009.

Apéndice



Este es el primer apéndice

A.1. Ejemplo de sección

Sección del apéndice