

# PROYECTO MÓDULO 7

## FECHAS IMPORTANTES

- ~~Marzo 22~~: Explicación de proyecto
- ~~Marzo 25~~: Grupos armados y eligen el proyecto
- ~~Marzo 25 - Abril 4~~: Discusión y aprobación final de propuestas (~~no calificado~~, al menos 1 miembro del grupo presente)
- ~~Abril 5 (última clase)~~: Revisión de primer avance (~~no calificado~~, al menos 1 miembro del grupo presente)
- **Abril 21 (2 semanas luego de acabado el módulo)**: Entrega final (video)

## 5 GRUPOS

- TRES grupos de 5 personas
- DOS grupos de 4 personas

## PROYECTO

Se propone a los bootcampers varios datasets de grandes cantidades de datos. **Los estudiantes deben formar grupos y elegir su proyecto de [aquí](#).**

### COMPONENTE A: Análisis Exploratorio de Datos (EDA)

- En todos los proyectos los estudiantes deberán realizar, en **Polars** o **DuckDB**, un análisis exploratorio a los datos.
- Los análisis a realizar son a discreción de los estudiantes. Idealmente utilizarán los conocimientos aprendidos a lo largo de todos los módulos del bootcamp para elegir los análisis más relevantes.
- El análisis debe incluir **graficación**
  - Los gráficos se pueden hacer con cualquier librería de graficación (incluido Pandas).
  - Los gráficos pueden ser realizados sobre un subconjunto de los datos (filtrado de TOPs, sampling).

### COMPONENTE B: Machine Learning (ML), NLP, o Análisis de Grafos

- En todos los proyectos los estudiantes deberán realizar DOS (en los grupos de 5 personas) o UN (en el grupo de 4 personas) análisis de entre:
  - Machine Learning: Deberán utilizar **Spark MLlib**
  - NLP: Deberán utilizar **Spark MLlib**
  - Análisis de Grafos: Deberán utilizar **Spark GraphFrames**
- **Ejemplos**: un modelo de regresión logística, un análisis del degree centrality de un grafo, un modelo utilizando tokenización TF-IDF, un modelo de xgboost, etc.
- Estos análisis pueden ser realizados en un subconjunto de los datos. Ejemplo: filtrar ciertos aeropuertos, filtrar solamente el TOP 10 de subreddits, filtrar solamente el TOP

100 usuarios, filtrar solamente carros del TOP 3 marcas, 1% de los autos, etc (esto debido a las limitaciones de hardware que tenemos)

### **COMPONENTE C: Análisis de Costos.**

- En todos los proyectos los estudiantes deberán realizar un análisis de costos si quisiéramos ejecutar los análisis que hicieron en alguna de las herramientas que hemos aprendido (Databricks, Snowflake, Google BigQuery, ClickHouse, etc).
- Deben tomar en cuenta aspectos como el movimiento de los datos, alojamiento de datos, tiempos de procesamiento, etc.
- La idea es que presenten un análisis diseminado como el que veremos el día 5 de Abril
  - Para esto tendrán que asumir muchas cosas: herramientas, proveedores, regiones, etc. Por ejemplo:
  - El entrenamiento de mi modelo de predicción en Spark demoró 10 minutos utilizando el 5% de los datos (este 5% puede representar por ejemplo: los registros del top 10 aeropuertos, top 3 reddit, top 5 marcas de autos, top 5 géneros de películas, etc).
  - Si quisiera entrenar el modelo con todos mis datos (es decir, pasar de 5% a 100%), entonces esperaré que mi modelo se entrene en no menos de 200 minutos (mis datos aumentaron x20)
  - Con este valor de 200 minutos ya puedo estimar cuanto me va a costar correr esto en Databricks utilizando la calculadora online:  
<https://www.databricks.com/product/pricing/product-pricing/instance-types>
  - También tengo que considerar que tengo que poner mis datos en algún Data Lake (ver análisis diseminado en diapositivas para obtener links a las fuentes de los precios)

### **COMPONENTE D: Diseño de Pipeline ETL.**

- En todos los proyectos los estudiantes deberán proponer el diseño y propósito de un Pipeline ETL en donde nuevos datos llegan a su dataset con cierta periodicidad (el análisis es enteramente a discreción de los bootcampers).
- Ejemplo: En el dataset de carros usados, podemos diseñar un pipeline ETL que cuando alguien suba la información de su carro usado a la página web de ventas, la página web le recomiende un precio sugerido basado en un modelo de ML. *[El G1 no puede utilizar este mismo ejemplo]*
- ATENCIÓN: NO DEBEN IMPLEMENTAR EL PIPELINE ETL, SIMPLEMENTE PROPONERLO. Se espera entonces alguna especie de diagrama (como los vistos en clase) con cómo se realiza la ingesta de datos, las etapas del pipeline, las herramientas utilizadas, etc.

## **EVALUACION**

- 10% → Calidad del **Análisis Exploratorio de Datos (COMPONENTE A)**

- Diversidad | Importancia | Interesante | Calidad de los gráficos
  - Haber cumplido las promesas luego del 1er avance
- 10% → Calidad del **Análisis de ML / NLP / Grafos (COMPONENTE B)**
  - Relevante al problema | Interesante
  - Haber cumplido las promesas luego del 1er avance
- 10% → **Análisis de Costos (COMPONENTE C)**
  - Congruente | Precios aproximados
  - Se debe adjuntar el análisis como un documento adicional o como texto en los cuadernos.
- 10% → **Diseño de Pipeline ETL (COMPONENTE D)**
  - Utilidad | Interesante | Calidad del diagrama
  - Haber cumplido con las promesas luego del 1er avance
- 30% → **Código fuente** (Cuadernos de Jupyter o Collab)
- 30% → **Presentación de 15 minutos (video)**
  - La presentación debe incluir: Análisis realizados, motivación y resultados | Desafíos en el análisis | Problemas que se encontraron al intentar ejecutar algún análisis | Cómo manejaron la gran cantidad de datos | Algún análisis que intentaron pero no se pudo completar | Análisis de Costos | Pipeline ETL
  - Recomendando obviar la presentación del dataset.
  - Recomendando NO presentar código fuente (al menos que sea absolutamente necesario)
  - Todos deben presentar
- BONUS +10% → **Análisis de costos del Pipeline ETL diseñado.**

## RUBRICA

- **COMPONENTE A: Análisis Exploratorio de Datos (10)**
  - +6: Se cumplieron todas las promesas hechas al momento de presentar el primer avance
    - Disminuye proporcionalmente con las promesas no cumplidas injustificadas
    - Si el análisis no se encuentra presente o no se completó, debe haber justificación o algún otro análisis similar que lo reemplace.
  - +4: Los gráficos son prolijos
    - [Cómo hacer visualizaciones EFECTIVAS | Claves y Consejos](#) .
- **COMPONENTE B: Análisis de ML / NLP / Grafos (10)**
  - +10: Se cumplieron todas las promesas hechas al momento de presentar el primer avance
    - Disminuye proporcionalmente con las promesas no cumplidas injustificadas
    - Si el análisis no se encuentra presente o no se completó, debe haber justificación
  - No se evaluará la calidad de los resultados. Por ejemplo, si un modelo tiene 0.2 de recall esto no va a afectar en la nota. Sin embargo, tiene que haber una explicación de porqué no se obtuvo el recall deseado y cómo se podría mejorar.
- **COMPONENTE C: Análisis de Costos (10)**
  - +5: Se han considerado la mayoría de componentes relevantes al análisis (disminuye proporcionalmente).
  - +3: Se incluyen referencias en forma de links o imágenes a la fuente del precio base de la componente (ejemplo: Se coloca el link al precio por TB de almacenamiento en AWS S3)
  - +2: El análisis se encuentra desglosado y en pasos
  - No se espera exactitud.
  - Si se tomó una muestra de los datos para el análisis especializado, el análisis de costos deberá tener una estimación de los costos utilizando todos los datos.
- **COMPONENTE D: Diseño de Pipeline ETL (10)**
  - +4: Se presenta la propuesta de un diseño de un Pipeline ETL como un diagrama (parecido a los vistos en clase) acompañado por un texto que explique brevemente la idea y el flujo del diagrama.
  - +2: El diseño es algo útil (tiene una buena motivación)
  - +4: Se detallan claramente las herramientas / servicios con las cuales se podría implementar el diseño.
- **Código fuente (30)**
  - +30: Se respetaron las reglas: Polars o DuckDB para el análisis exploratorio y PySpark (y sus sublibrerías: MLib / GraphFrames / GraphX / etc) para el análisis especializado. También pueden aventurarse a utilizar Dask (alternativa a PySpark). Se puede utilizar Pandas para manejar los datos una vez filtrados / agrupados / etc, o para realizar gráficos.
    - Si se desea utilizar otra herramienta, deben decírmelo con anticipación.

- 0: No se respetaron las reglas (ej: se utilizó Pandas para todos los análisis).
- No se evaluará la calidad del código fuente.
- **Presentación (30)**
  - +10: Se presentan los análisis realizados (EDA, especializado y costos)
  - +5: Se presentan las dificultades / problemas / desafíos que tuvieron en el proyecto
  - +5: El video se mantiene dentro de los 15 minutos (máximo 15:00)
  - +5: Buenas prácticas en diapositivas (no son enteramente texto, no tienen fondo negro + fuente amarillo patito)
  - +5: Calidad de la presentación
  - Si un estudiante no está presente en el video, se califica con 0
  - Puede ser un video editado, o una presentación grabada
  - No perderán puntos por mal audio, o con ruido blanco de fondo. Lo único importante es que las diapositivas se puedan ver y el audio se pueda entender
- **Análisis de Costos del Pipeline ETL (BONUS +10)**
  - Los precios tienen coherencia con los sistemas utilizados y con lo que hace el pipeline (es decir, no se han inventado valores)

La nota final es la sumatoria de todas las componentes / 2 (ya que el proyecto equivale al 50% de la nota final). Ejemplos:

- Si se obtiene 90 puntos, la nota del proyecto será  $90/2 = 45$ .
- Si se obtiene más de 100 puntos, el excedente va como puntos adicionales al promedio final. Por ejemplo, si un estudiante saca 105, entonces su nota de proyecto será 52.5.

## RECURSOS IMPORTANTES

- **DuckDB**

- Cuaderno hecho en LAB03 con más ejemplos y links a documentación:
  - 🔗 LAB03 [Profesor]: Batalla de Herramientas.ipynb
- Documentación Oficial: <https://duckdb.org/docs/index>
- Funciones de Strings: <https://duckdb.org/docs/sql/functions/char.html>
- Crear una función en Python para transformar una columna en DuckDB: <https://duckdb.org/docs/api/python/function.html>

- **Polars**

- Cuaderno hecho en LAB03 🔗 LAB03 [Profesor]: Batalla de Herramientas.ipynb
- Documentación Oficial: <https://docs.pola.rs/user-guide/getting-started/>
- De Pandas a Polars: <https://docs.pola.rs/user-guide/migration/pandas/>


- **Spark MLlib**

- Regresión Lineal (LAB02):
  - 🔗 7.1. - 7.2. Machine Learning en Spark (Guía Profesor).ipynb
- Decision Tree: 🔗 7.3. - 7.5. Decision Tree (Guía Profesor).ipynb
- NLP con TF-IDF y Support Vector Classifier:
  - 🔗 7.6. Machine Learning: NLP (Guía Profesor).ipynb

- **Spark GraphFrames**

- Grafos con GraphFrames:
- 🔗 9.0. Grafos (Guía Profesor).ipynb

- **Análisis de Costos**

- Calculadora de Databricks:  
<https://www.databricks.com/product/pricing/product-pricing/instance-types>
- Diapositivas con links a otros precios:  
 VIRTUAL06 - Ejemplo de Arquitecturas y Estimación de Costos.pdf