# Prueba Técnica Desarrollador Big Data.

El formato de entrega de la prueba debe hacerse sobre un repositorio en GIT con los documentos y los Readme donde se explique el funcionamiento de las clases implementadas.

* Herramientas necesarias:
  + Databricks Community: <https://community.cloud.databricks.com/login.html>
  + Datos Base: Tweets.csv
  + VsCode: [https://code.visualstudio.com](https://code.visualstudio.com/)
  + Python: <https://www.python.org/downloads/>

# Retos Python:

1. Construir un script de python que permita simular los movimientos de todas las piezas de ajedrez dada una posición aleatoria dentro del tablero, el formato de impresión del script se debe hacer de esta manera:

8 para la posición de la pieza.

1 para sus posibles ubicaciones luego del movimiento.

0 para las posiciones donde la piza no puede llegar.

**Caballo Reina**

[[0 0 0 0 0 0 0 0] [[0 **1** 0 0 **1** 0 0 **1**]

[0 0 0 **1** 0 **1** 0 0] [0 0 **1** 0 **1** 0 **1** 0]

[0 0 **1** 0 0 0 **1** 0] [0 0 0 **1** **1** **1** 0 0]

[0 0 0 0 **8** 0 0 0] [**1 1 1 1 8 1 1 1**]

[0 0 **1** 0 0 0 **1** 0] [0 0 0 **1** **1** **1** 0 0]

[0 0 0 **1** 0 **1** 0 0] [0 0 **1** 0 **1** 0 **1** 0]

[0 0 0 0 0 0 0 0] [0 **1** 0 0 **1** 0 0 **1**]

[0 0 0 0 0 0 0 0]] [**1** 0 0 0 **1** 0 0 0]]

1. Construir una clase en python que permita simular un objeto algebraico.

Las operaciones que se deben permitir sobre este elemento son:

* Suma
* Resta
* Multiplicación entre polinomios
* Multiplicación por un escalar
* Evaluación del polinomio

El objetivo del par de retos en python es el de evaluar las capacidades de la persona para adaptarse manejar diferentes tipos de implementación.

Consideraciones:

* Código Duplicado: Todo el código que deba o pueda ser reusado debe ser encapsulado.
* Mantener versionamiento Git.

Consideraciones que agregan puntuación:

* SRP: Principio de única responsabilidad.
* Principio de sustitución de Liskov: Interfaces que permitan a las clases diseñadas extenderse
* DIP: Dependency Inversion Principle, el código debe depender de las interfaces no de las implementaciones.