# Cálculo de Pi

El Valor de Pi puede ser calculado de muchas maneras. Considere el siguiente método para aproximar Pi

* Tenga un cuadrado de lado 4 (Área = 16)
* Haga un Circulo de radio 2 dentro del Cuadrado (Área = 4 π)
* Genere N puntos de manera aleatoria dentro del cuadrado
* Sea p el número de puntos que están dentro del circulo
* Entonces π

Círculo

Descripción generada automáticamente

# Scala challenge

Implemente en Scala la aproximación de π con el algoritmo descrito arriba en paralelo (Spark) y mida el tiempo que tarda en aproximar π.

Genere el Jar que lo pueda ejecutar un *spark-submit* y escriba el comando para correrlo con los valores de 1,000 vueltas y en 4 partes (slices), el comando de *spark-submit*.

**EXTRAS**:

* logee información relevante del proceso (en un log)
* Haga el mismo código, pero secuencial y muestre el tiempo de ejecución vs el paralelizado

## Algoritmo

* Genere un Código en Scala que reciba como 2 parámetros por argumentos:
  + El numero de muestras que se tomaran
  + El número de partes en que se dividirá el computo (paralelo)
* El algoritmo será el siguiente:
  + Crear una spark sesión
  + Asignar los valores de muestras y partes (slices) de los argumentos en una variable
  + Time inicial = tiempo actual
  + Utilizar un map paralelizado (Dentro de SparkContext) para saber si una muestra aleatoria esta dentro del círculo o no.
  + Calcular el valor de pi con la formula mostrada en la sección anterior
  + Tiempo total = tiempo actual – time inicial
  + Mostrar el valor de la aproximación de Pi
  + Mostrar el error del valor de aproximación de Pi
    - Error = abs(estimación - π)

**Pistas**:

* **Locación de los centros**: La localización del centro del cuadrado y circulo son muy importantes, recomiendo que el centro este en la locación (0,0)
* **Números Aleatorios**: Utilice la función import scala.math.random (<https://www.scala-lang.org/api/2.12.4/scala/util/Random.html>) para generar números aleatorios,
  + Esto generara un numero aleatorio entre 0 y 1, necesita transformar esto a un numero entre un numero que se requiera.
* **Determinar si el punto está dentro del círculo o no**:
  + Sea el punto donde el circulo tiene su centro, por ejemplo, si el circulo tiene su centro en el punto (2,4), entonces , por lo tanto
  + Sea el radio del circulo
  + Sea el punto que queremos saber si esta dentro del círculo, entonces:
  + El punto esta dentro del circulo si
  + Ejemplo: si el centro del círculo está en el punto y tiene radio 3, determine si el punto esta dentro del círculo:

**Esto es falso, por lo que el punto no está dentro del circulo**

* **Enfoque Map-Reduce**:
  + Utilice enfoque map reduce para paralelizar con facilidad
    - Map: Sacar un punto aleatorio, si el punto esta dentro del circulo 1, si no 0
    - Reduce: Sumar todos los resultados anteriores
  + Aproxime con la formula mostrada
* Estructura de archivo: Puede utilizar como base el siguiente archivo, la estructura del folder debe ser com/main/pi\_calculation.scala 