**UNIVERSIDAD TECNOLÓGICA NACIONAL**
**FACULTAD REGIONAL CONCEPCIÓN DEL URUGUAY**
**DEPTO. ING. EN SISTEMAS DE INFORMACIÓN**

# *Algoritmo de Clasificación para Datos Masivos*

**M. SOLEDAD RETAMAR**

*Director:* **DR. GUILLERMO LEGUIZAMÓN**
*Co-Directora:* **MCS. NORMA HERRERA**

*Tesis para optar al grado de*
*Especialista en Ciencias de la Computación*
*con Orientación en Bases de Datos*

*Año: 2017*

# 1 Justificación

Las grandes cantidades de datos que se producen en la actualidad, sumadas a su heterogeneidad, hacen que las herramientas tradicionales de análisis de datos no resulten adecuadas para su recopilación, almacenamiento, gestión y análisis. En este contexto se comienza a hablar del término Big Data, haciendo referencia a características como gran volumen, velocidad y variedad de producción de los datos, y a las herramientas que se utilizan para encontrar valor en las mismas.
Clasificar estos datos complejos no puede realizarse con las herramientas tradicionales de análisis de datos, por lo que resulta necesario diseñar algoritmos especiales para analizar estos grandes volúmenes de datos.
En esta tesis se propone desarrollar un algoritmo de clasificación para datos masivos inspirado en el tradicional algoritmo C4.5 que demuestre un desempeño eficiente implementado en las tecnologías existentes para el tratamiento de datos masivos

# 2 Fundamentación

Esta tesis abordará el diseño de un algoritmo de clasificación para datos masivos. Debido a la amplitud del área de estudio esta sección se organiza en cuatro subsecciones. En la sección 2.1 se hará una breve descripción de los conceptos y características principales que presentan las Bases de Datos Masivos, en la sección 2.2 se introduce al concepto de Minería de Datos, se describen las principales aplicaciones de la tarea de clasificación y las diversas técnicas para llevar a cabo esta tarea. En la sección 2.3 se abordará mas especiíficamente el algoritmo de clasificación C4.5 y algunas implementaciones realizadas sobre datos masivos. Por último, en la sección 2.4, se introducirá a las principales herramientas para el tratamiento de Datos Masivos.

## 2.1 Bases de Datos Masivos

En la actualidad se producen diariamente grandes volúmenes de datos de diversos tipos (ej., textos, imágenes, audio, videos) y desde los más variados orígenes (ej., web, GPS, redes sociales). Debido a los avances en las tecnologías de la información que han facilitado la recolección y el almacenamiento [1] de los datos se estima que para el 2020 nuestro planeta alcanzaría los 44 Zettabytes [2] de datos, lo que implicaría un volúmen 10 veces mayor que en 2013.

En este contexto, surge el término *Datos Masivos* o *Big Data* (en inglés) para referirse a conjuntos de datos cuyo tamaño supera la capacidad de las herramientas tradicionales o son demasiado complejos para procesarlos en una sola máquina. Aunque el significado del término *Big Data* sigue siendo objeto de algunas discusiones [3,4] se ha optado por definirlo a través de sus principales características [5]:

> Volúmen: hace referencia al tamaño de los datos, enormes cantidades de información que necesitan ser procesadas y analizadas para obtener conocimiento útil y valioso.
>
> Variedad: distintos tipos de datos provenientes de diversas fuentes que pueden organizarse tanto en forma estructurada como no estructurada.
>
> Velocidad: referido a la rapidez con que se generan y procesan los datos.

Variabilidad: referido a los cambios en las estructuras de datos y en cómo los usuarios quieren interpretarlos.

Valor: se refiere al valor que implica para las organizaciones analizar y convertir en conocimiento estos grandes volúmenes de datos. La información obtenida debería implicar una ventaja estratégica para la toma de decisiones gracias a la posibilidad de responder preguntas que antes no era posible.

## 2.2 Minería de Datos

El Descubrimiento de Conocimiento en Bases de Datos, KDD (por sus siglas en inglés Knowledge Discovery in Databases), se define como el proceso no trivial de identificar patrones válidos, desconocidos, potencialmente útiles y comprensibles en los datos [6]. Como se ilustra en la figura 1 el proceso de KDD consta de una secuencia iterativa de etapas: Recopilación e integración de datos; Selección, Limpieza y transformación de datos; Minería de datos; Evaluación; Difusión, uso y monitorización de modelos.

La fase de Minería de Datos es la más característica del KDD por eso con frecuencia se utiliza esta fase para nombrar todo el proceso. Su objetivo es producir nuevo conocimiento que pueda ser útil al usuario. Esto se realiza construyendo un modelo basado en los datos recopilados a tal fin.

Si bien la posibilidad de aplicar técnicas de Minería de Datos en el contexto de Big Data representa una oportunidad para la toma de decisiones y la planificación estratégica de muchas organizaciones, implica también nuevos problemas y desafíos [7] debido a que las técnicas tradicionales no siempre son escalables a la complejidad y magnitud de estos repositorios de datos. Por este motivo, es necesario re-diseñar dichas técnicas y algoritmos de modo que puedan ser aplicados a problemas del mundo real que impliquen grandes volúmenes de datos. Una de las principales técnicas aplicadas en el proceso de KDD es la clasificación, que consiste en asignar una clase o categoría a datos u objetos en base a sus atributos utilizando un modelo creado con datos previamente clasificados.
Algunos dominios de aplicación frecuentemente encontrados en la literatura [8] para la tarea de clasificación se indican continuación:

- Marketing de Clientes
  Dado que el problema de clasificación relaciona las características de las variables para identificar una clase, se puede usar para predecir los intereses de compra de los clientes en base a los registros de las compras anteriores.

- Diagnóstico de Enfermedades
  En los últimos años, el uso de métodos de minería de datos en la tecnología médica ha adquirido cada vez mayor atracción. A través de los atributos registrados en las historias clínicas de pacientes y sus correspondientes diagnósticos, es posible realizar predicción de enfermedades en base a un conjunto de características presentes en los pacientes.

- Detección supervisada de Eventos
  En muchos escenarios temporales, las etiquetas de clase pueden estar asociados con marcas de tiempo correspondientes a los acontecimientos inusuales. Por ejemplo, una actividad inusual puede ser representada como una etiqueta de clase. En tales casos, los métodos de clasificación de series de tiempo suelen ser muy útiles.

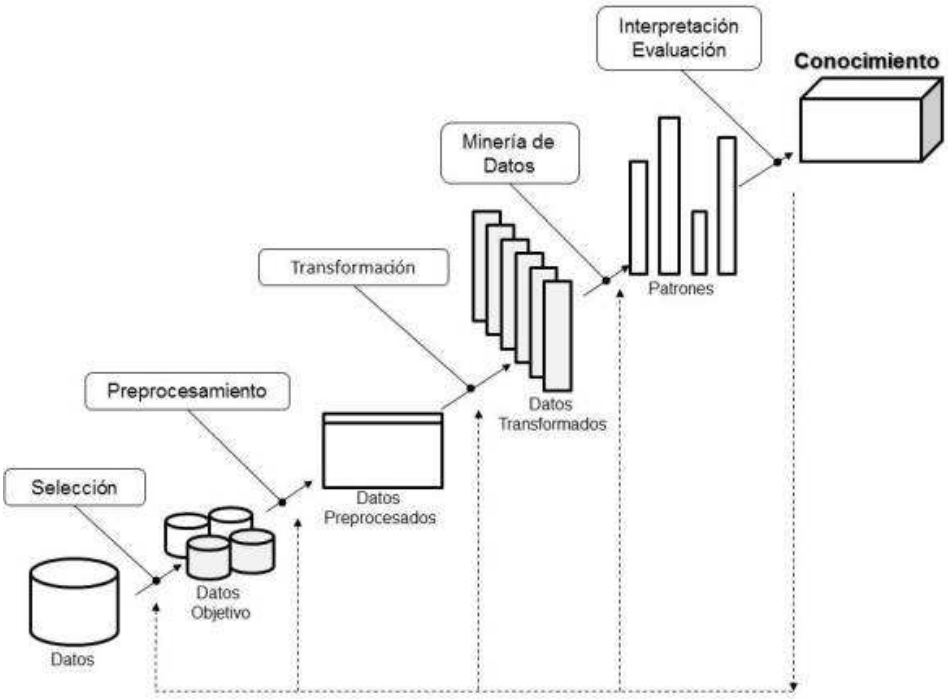



Figure 1: Tareas en el Proceso de KDD

- Análisis de Datos Multimedia
  A menudo es deseable llevar a cabo la clasificación de grandes volúmenes de datos multimedia como fotos, vídeos, audio u otros datos más complejos. El análisis de estos datos puede ser difícil, debido a la complejidad de la función de espacio subyacente y la brecha semántica entre los valores de características e inferencias correspondientes.

- Análisis de Datos Biológicos
  Estos datos se representan a menudo como secuencias discretas o en forma de redes que permiten predecir, aplicando modelos de clasificación, propiedades particulares de las secuencias de interés.

- Filtrado y clasificación de Documentos
  Las aplicaciones como servicios de noticias o repositorios digitales, requieren la clasificación de un gran número de documentos en tiempo real. Esta aplicación se conoce como categorización de documentos, y es una importante área de investigación en sí misma.

### 2.2.1 Algoritmos de Clasificación

El problema de la clasificación puede enunciarse como sigue [8]: *Dado un conjunto de puntos de entrenamiento y sus etiquetas asociadas, determinar la clase de etiqueta para una instancia de prueba sin etiquetar.* Así, el problema de clasificación segmenta en tantos grupos la instancia de prueba como etiquetas de clases definidas haya. Los algoritmos de clasificación típicamente contienen dos fases:

Fase de entrenamiento: se construye un modelo de las instancias de entrenamiento el cual puede estar representado de varias formas tales como reglas de clasificación, árbol de decisión, fórmulas matemáticas o redes neuronales [9]. En algunos casos, cuando el aprendizaje es muy lento, la fase de entrenamiento se omite por completo [8] y la clasificación se realiza directamente en las instancias de prueba.

Fase de prueba: el modelo se utiliza para asignar una etiqueta a una instancia de prueba no etiquetada. Generalmente la salida de un algoritmo de clasificación podrá ser un etiquetado discreto, donde se asigna una clase de pertenencia a cada instancia, o una puntuación numérica, donde para cada instancia se devuelve una probabilidad de pertenecer a cada una de las clases existentes.

Existen diversas técnicas aplicadas a la clasificación de datos [8], se describen a continuación los métodos principales:

**Métodos probabilísticos.** Son los más relevantes entre los métodos de clasificación de datos [8]. Estos algoritmos utilizan la inferencia estadística para determinar la clase de una instancia dada. Están basados en la teoría probabilística, específicamente en el teorema de Bayes, el cual permite estimar la probabilidad de un suceso a partir de la probabilidad de que ocurra otro del cual depende el primero [10]. Uno de los algoritmos más conocidos es el Naive Bayes que ha demostrado [11] ser fácil de implementar, poseer eficiencia computacional y tasa de clasificación y obtener resultados precisos cuando el número de registros es muy grande.

**Algoritmo del vecino más próximo (KNN) y variantes.** El algoritmo del vecino más próximo (Nearest Neighbour, NN) tiene como característica que su implementación es relativamente sencilla. En el contexto tratado de la clasificación la idea básica es la siguiente: se calcula la similitud entre la instancia a clasificar y cada uno de los datos de entrenamiento, las instancias más parecidas estarán indicando a qué clase o categoría se debe asignar el dato que se desee categorizar. Otra ventaja mencionada en [11] es la robustez cuando los datos de entrenamiento presentan ruido, aunque se menciona también que puede ser sensible a atributos irrelevantes y tener tiempos de ejecución excesivos para encontrar los vecinos cercanos cuando el conjunto de datos es muy grande.

Existe una variante muy conocida de este algoritmo que es el KNN (k-nearest neighbour), este algoritmo consiste en tomar las k instancias más relevantes, en lugar de sólo la primera [12].

**Redes Neuronales.** Otros métodos muy utilizados para resolver el problema de clasificación son las *Redes Neuronales*. Estos modelos computacionales surgieron como un intento de simular la estructura y el comportamiento del cerebro humano. Se basan en el aprendizaje a través de la experiencia, con la consiguiente extracción del conocimiento a partir de la misma. Entre los algoritmos mas conocidos en esta área se encuentran los llamados Mapas Auto-organizados o SOM (Self Organizing Maps) de Kohonen [13]. Entre las principales ventajas de estas técnicas se mencionan [11] la facilidad de uso, que posee pocos parámetros para ajustar, no necesita ser reprogramada y es aplicable a una amplia gama de problemas; mientras que las desventajas principales son la complejidad en determinar la cantidad de neuronas y capas necesarias, presentan un tiempo elevado de procesamiento y pueden tener un aprendizaje muy lento.

**Máquinas de Vectores de Soporte.** Las Máquinas de Vectores de Soporte o Support Vector Machines $SVM$ son un conjunto de algoritmos de aprendizaje supervisado [14] cuyo objetivo consiste en separar las clases existentes en un espacio de alta dimensionalidad mediante la construcción de hiperplanos óptimos que mejor separen las clases. A pesar de sus buenos fundamentos teóricos y buen desempeño al generalizar, las SVM no son adecuadas para clasificación con grandes conjuntos de datos debido a su excesivo computacional [15].

**Algoritmos genéticos.** Representan una abstracción del concepto biológico de evolución natural y tiene numerosas aplicaciones en problemas de optimización [13]. Su funcionamiento está basado en los mecanismos de selección natural, combinando la supervivencia del más apto con un intercambio de información entre miembros de una población de posibles soluciones. Durante cada generación o iteración se producen nuevas poblaciones mediante la aplicación de los denominados operadores genéticos: selección, cruza y mutación. Cada solución en la población está asociada a un valor de aptitud, dependiendo de la función a optimizar. El operador de selección escoge una solución de la población actual para que continúe en la siguiente población en base a una probabilidad proporcional a su valor de aptitud; el operador de cruce, crossover (en inglés), combina con una probabilidad los segmentos desde un punto de cruce y el operador de mutación cambia cada posición de la cadena solución con una probabilidad denominada probabilidad de mutación. Los algoritmos genéticos se han utilizado para resolver diversas tareas de la minería de datos, tales como clasificación, clustering, minería web y reglas de asociación [16]. En las tareas de clasificación se han presentado principalmente dos dificultades: suelen necesitar el conjunto de datos completo en la memoria principal y a partir de cierto tamaño se ha observado pérdida en la calidad de los resultados.

**Árboles de Decisión.** Es uno de los métodos más utilizados para crear clasificadores. A través de esta técnica, el conjunto de datos de entrenamiento es dividido en dos o mas partes utilizando alguno de sus atributos. Cada uno de los subconjuntos resultantes representan una rama en el árbol que se debe volver a dividir utilizando otro atributo o característica. Este proceso se repite hasta que sólo haya datos de la misma clase en cada rama del árbol.

Las principales ventajas de esta técnica [11] se encuentran en que los modelos son fáciles de interpretar, tiene una implementación simple, soporta valores continuos y discretos y es tolerante al ruido presente en los datos. Una de las desventajas señaladas es que no tiene buen desempeño cuando el conjunto de datos de entrenamiento es pequeño. El C4.5, desarrollado por Quinlan [17] para mejorar su antecesor ID3, es uno de los más conocidos en este grupo y será analizado en la siguiente sección.

## 2.3 Algoritmo C4.5

El algoritmo C4.5 comienza analizando todo el conjunto de datos para crear la primera división del nodo raíz del árbol. Cada partición o división del conjunto de datos, *S*, se realiza probando todos los posibles valores de las instancias en cada dimensión o atributo *A*, y después se selecciona a la mejor partición de acuerdo a algún criterio. Este proceso se realiza de manera recursiva.

Algunos de los criterios para la elección del atributo (A) para realizar la división de los datos, S, están basados en alguno de los siguientes índices [18]: Entropía (ecuación 1), Split Info (ecuación 2), Information Gain (ecuación 3) y Gain Ratio (ecuación 4).

$$E(S) = -\sum_{i=1}^{N} p_i \log(p_i) \qquad (1)$$

$$IG(S,A) = E(S) - \sum_{i=1}^{N} \frac{S_i}{S} E(S_i) \qquad (3)$$

$$SI(S) = -\sum_{i=1}^{N} \frac{S_i}{S} (\log(\frac{S_i}{S})) \qquad (2)$$

$$GR(S,A) = \frac{IG(S,A)}{SI(S))} \qquad (4)$$

Luego, con cada subconjunto formado realiza un proceso iterativo que consiste en volver a analizar los datos que forman parte del subconjunto buscando un nuevo atributo y su correspondiente valor donde hacer una nueva división de datos.
C4.5 utiliza la relación de ganancia de información como el criterio predeterminado para elegir los mejores atributos de división. Este proceso iterativo continúa hasta llegar a un subconjunto donde todos los datos son de la misma clase y donde ya no se requiere una nueva división. O hasta que la cantidad de datos de una clase sea tan pequeña que no se justifica una nueva división. Esta cantidad de datos por la cual no se debe llevar a cabo una nueva división es un parámetro del algoritmo C4.5.

Es difícil determinar a priori cuántas veces se recorre la base de datos en el algoritmo C4.5, ya que depende de cómo queden formados los nuevos subconjuntos de datos.

Respecto a su predecesor permite trabajar con atributos numéricos y nominales, tratar con conjuntos de datos con valores de faltantes y podar los árboles después de la creación. La principal desventaja del algoritmo C4.5 es que requiere más cantidad de tiempo de procesamiento y memoria a medida que aumentan las muestras.

### 2.3.1 Implementaciones sobre BigData

Aunque se han propuesto muchos enfoques para analizar conjuntos de datos de tamaño pequeño a mediano sólo unos pocos han sido adaptados para manejar grandes conjuntos de datos [19]. A continuación se mencionarán algunas propuestas de algoritmos basados en árboles de decisión donde se ha abordado el problema de la escalabilidad, capaces de trabajar con grandes conjuntos de datos:

- SLIQ (Supervised Learning In Quest)
  Es un árbol de decisión diseñado para clasificar grandes cantidades de datos. En una primera etapa, realiza una tarea de pre-clasificación en la fase de crecimiento del árbol, lo cual hace que la clasificación no sea compleja para cada nodo. La idea principal del algoritmo se basa en tener una lista por cada atributo. La principal problemática que presenta es tener que almacenar en memoria todo el conjunto de entrenamiento para construir el árbol de decisión [8]

- SPRINT (Scalable PaRallelizable INndution of decision Trees)
  Este algoritmo es una mejora del algoritmo SLIQ, continúa con la idea de tener una

lista por cada atributo. Su diferencia con SLIQ radica en la forma de cómo se representan las listas para cada atributo. Al igual que SLIQ presenta problemas de memoria [20].

- RainForest
  Este algoritmo trata principalmente de reducir las estructuras de manera que puedan ser almacenadas en memoria. Se accede al conjunto hasta dos veces en cada nivel del árbol, con lo que resulta en un mayor tiempo de procesamiento. La idea es formar conjuntos de listas denominados AVC (AtributoValor, Clase) para describir a los objetos del conjunto de entrenamiento. Estas listas contendrán a todos los posibles valores que un atributo pueda tomar, además de su frecuencia en el conjunto de entrenamiento. De esta manera, las listas no serán de gran longitud y con ello podrán ser almacenadas en memoria [21].

- C4.5
  El algoritmo C4.5 clásico es uno de los enfoques más utilizados en tareas de clasificación de conjuntos datos pequeños y medianos. Sin embargo, hay varios inconvenientes del algoritmo C4.5 aplicado a grandes conjuntos de datos. En [22], se propone una implementación del algoritmo C4.5 con MapReduce con el objetivo de escalar a dicho algoritmo y mejorar los tiempos de entrenamiento y evitar los altos costos en las operaciones de lectura y escritura por no poder mantener grandes conjuntos de datos en memoria. En el estudio experimental, se realizaron pruebas con conjuntos de datos de diversos tamaños y con distintas cantidades de nodos. Los resultados empíricos indicaron que la implementación del algoritmo C4.5 utilizando el paradigma MapReduce era eficiente en relación al tiempo y además, escalable.

  Posteriormete, en [23] se desarrolla un algoritmo paralelo basado en el árbol de decisión C4.5 llamado MR-C4.5-Tree usando también el paradigma MapReduce. Los autores lo comparan con el algoritmo C4.5 clásico y demuestran que el árbol MR-C4.5 propuesto es factible de escalar eficazmente y obtiene tasas de éxito similares a las del algoritmo tradicional para grandes conjuntos de datos pero en un tiempo menor. Una de las desventajas que presenta es que agrega algunos parámetros: profundidad del árbol, cantidad mínima de datos por clase y tasa mínima de precisión.

A pesar de las posibilidades tecnológicas existentes en la actualidad para procesar de manera paralela grandes bases de datos [24], existe un compromiso y un balance que se debe tener en cuenta entre el esquema de comunicación, el uso de memoria, y la sincronización para que estos algoritmos sean eficientes .

## 2.4 Herramientas

A continuación se describen algunas de las herramientas más utilizadas para realizar tareas de análisis de Big Data. Todas las herramientas mencionadas son de código abierto y pueden ser utilizadas sin ningún costo.

### 2.4.1 Paradigma MapReduce

MapReduce es un paradigma de programación [25] desarrollado por Google para procesar y/o generar conjuntos de datos grandes que no se ajustan a la capacidad de memoria física.

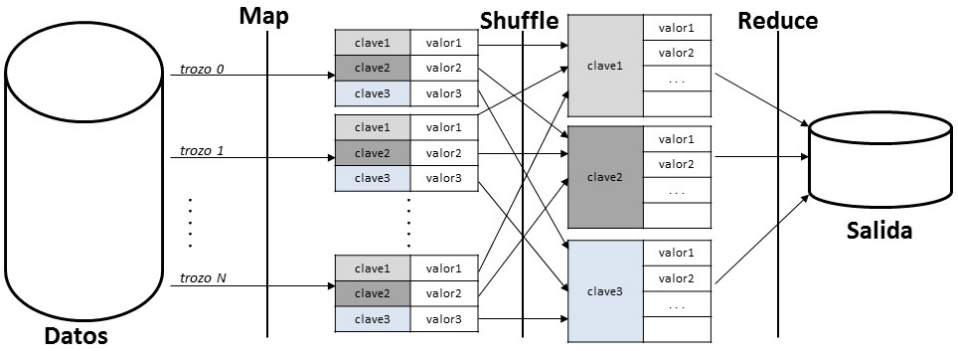



Figure 2: Esquema de Trabajo del Paradigma MapReduce

Está basado en la programación funcional y consiste en dos pasos principales: la fase Map y la fase de Reduce. Cada fase tiene una pareja *(clave; valor)* como entrada y como salida.

La fase Map toma cada pareja y genera un conjunto intermedio de pares clave-valor. Entonces, MapReduce fusiona todos los valores asociados con la misma clave en una lista (etapa shuffle). La fase Reduce toma esta lista como entrada y produce los valores finales. En la figura 2 se ilustran estas tareas: se ejecutan las funciones Map en paralelo y de forma independiente, mientras que las operaciones Reduce esperan hasta que Map haya finalizado.

Una de las tecnologías más destacadas [26] que implementan el paradigma MapReduce para poder manejar grandes volúmenes de datos son los Sistemas Hadoop. Éstos utilizan el almacenamiento distribuido y procesan grandes volúmenes de datos estructurados, semi estructurados y no estructurados con el propósito de extraer información relevante. Entre los framework mas destacados que utilizan esta tecnología se encuentran: Apache Hadoop [27], Spark [28], Haloop [29].

Junto con la aparición de estas plataformas han surgido librerías de aprendizaje de máquinas paralelas, entre las que se encuentran Mahout [30] que se ejecuta sobre Apache Hadoop y MLlib [31] que utiliza Apache Spark. Estas librerías implementan algoritmos para diversas tareas del proceso de Minería de Datos Masivos.

# 3 Objetivos

## Objetivo General

El objetivo de este trabajo es proponer e implementar una aproximación del algoritmo de clasificación C4.5 para datos masivos.

## Objetivos Específicos

Los objetivos específicos que se esperan lograr a partir del desarrollo de este trabajo son:

- Realizar un estudio detallado de implementaciones para grandes conjuntos de datos del algoritmo de clasificación C4.5, el funcionamiento de los mismos, sus ventajas y desventajas.
- Proponer e implementar un algoritmo basado en C4.5 para la clasificación de datos

masivos que mejore las tasas de éxito de al menos uno de los algoritmos ya implementados.

# 4 Metodología

La hipótesis de este trabajo considera que la clasificación de los tipos de datos complejos y de volúmen creciente que se gestionan en la actualidad no puede realizarse con las herramientas tradicionales de análisis de datos, por lo que resulta necesario diseñar algoritmos especiales. En tal sentido se propone el desarrollo de un algoritmo de clasificación para datos masivos inspirado en el tradicional algoritmo C4.5.

Para lograr los objetivos propuestos en este trabajo, hemos dividido el desarrollo de la misma en tres etapas. La primera etapa tiene como objetivo principal una recopilación y estudio de la bibliografía existente en la temática. La segunda etapa consiste en el diseño e implementación de los algoritmos necesarios para lograr los objetivos planteados. Por último, la tercera etapa consiste en seleccionar repositorios de datos para realizar la evaluación experimental de los algoritmos propuestos y poder analizar los resultados obtenidos. Detallamos a continuación las actividades que se realizarán en cada etapa:

**Primera Etapa**

- Recopilación y estudio de la bibliografía sobre algoritmos de clasificación de datos masivos mediante árboles de decisión. Para concretar esta etapa se utilizará el Portal de Bibliotecas de la UTN y se realizarán búsquedas de bibliografía especializada en internet.
- Estudio e instalación de las tecnologías necesarias para el desarrollo sobre un entorno Big Data.

**Segunda Etapa**

- Implementación del algoritmo en estudio en la infraestructura propuesta.
- Diseño de un nuevo algoritmo de construcción de un árbol de decisión para la clasificación de datos masivos.
- Programación de los métodos requeridos para cumplir el objetivo.

**Tercera Etapa**

- Selección de repositorios de datos.
- Análisis experimental de ambos algoritmos sobre los mismos conjuntos de datos.
- Publicación de los resultados obtenidos.

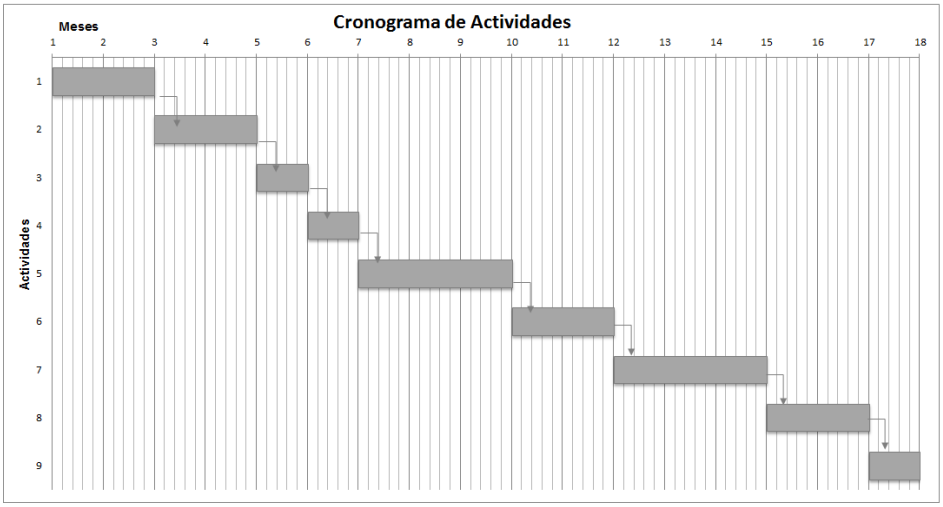
# 5 Cronograma del Plan de Tareas

## 5.1 Actividades

1. Estudio de la bibliografía y los algoritmos de clasificación de datos masivos mediante árboles de decisión, más específicamente aquellos basados en C4.5.

2. Estudio e instalación de las tecnologías necesarias para el desarrollo sobre un entorno Big Data.
3. Análisis del conjunto de datos sobre los que se implementarán los algoritmos y posible selección de atributos.
4. Implementación de al menos uno de los algoritmos estudiados en la infraestructura escogida
5. Diseño de un nuevo algoritmo de construcción de un árbol de decisión para la clasificación de los datos anteriormente seleccionados.
6. Implementación del algoritmo propuesto en el punto anterior.
7. Análisis experimental de ambos algoritmos sobre los mismos conjuntos de datos.
8. Publicación de los resultados obtenidos en Congresos de relevancia.
9. Redacción del informe final.

### 5.2 Diagrama de ejecución




## 6 Bibliografía

[1] Charu C. Aggarwal. Data streams: models and algorithms. *Springer Science & Business Media*, 2007.

[2] IDC. The digital universe of opportunities. *EMC*, 2014.

[3] Douglas Laney. 3D data management: Controlling data volume, velocity, and variety. February 2001.

[4] Mark Van Rijmenam. Why the 3vs are not sufficient to describe big data. *BigData Startups*, 2013.

[5] Wei Fan and Albert Bifet. Mining big data: Current status, and forecast to the future. *SIGKDD Explor. Newsl.*, 14(2):1–5, April 2013.

[6] Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[7] Chun-Wei Tsai, Chin-Feng Lai, Han-Chieh Chao, and Athanasios V. Vasilakos. Big data analytics: a survey. *Journal of Big Data*, 2(1):21, 2015.

[8] Charu C. Aggarwal. *Data Classification: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2014.

[9] V. Suresh Kumar Smitha T. Application of big data in data mining. *International Journal of Emerging Technology and Advanced Engineering*, 2013.

[10] Cristopher Bishop. Pattern recognition and machine learning (information science and statistics). 2007.

[11] Sagar S. Nika. A comparative study of classification techniques in data mining algorithms. *Oriental Journal of Computer Science & Technology*, April 2015.

[12] Carlos G. Figuerola, José L. Alonso Berrocal, Angel F. Zazo Rodríguez, Emilio Rodríguez, and Grupo Reina. Algunas técnicas de clasificación automática de documentos. 2008.

[13] David Goldenberg. Categorización automática de documentos con mapas auto organizados de kohonen. *Universidad Politécnica de Madrid*, 2007.

[14] Hsuan-Hung Lin and Lin-Yu Tseng. Disulfide bonding pattern prediction using support vector machine with parameters tuned by multiple trajectory search. pages 293–298, 2009.

[15] Jair Cervantes Canales, Xiaoou Li Zhang, and Wen Yu Liu. Clasificación de grandes conjuntos de datos vía máquinas de vectores soporte y aplicaciones en sistemas biológicos. 2009.

[16] Shi Cheng, Bin Liu, TO Ting, Quande Qin, Yuhui Shi, and Kaizhu Huang. Survey on data science with population-based algorithms. *Big Data Analytics*, 1(1):3, 2016.

[17] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[18] Luis A Caballero-Cruz, Asdrúbal López-Chau, and Jorge Bautista-López. Arbol de decisión c4. 5 basado en entropía minoritaria para clasificación de conjuntos de datos no balanceados. 2015.

[19] Victor Andres Ayma, Rodrigo Ferreira, Patrick Happ, Dario Oliveira, Raúl Feitosa, Gilson Costa, Antonio Plaza, and Paolo Gamba. Classification algorithms for big data analysis, a map reduce approach. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(3):17, 2015.

[20] John Shafer, Rakesh Agrawal, and Manish Mehta. Sprint: A scalable parallel classifier for data mining. In *Proc. 1996 Int. Conf. Very Large Data Bases*, pages 544–555. Citeseer, 1996.

[21] Joao Gama, Pedro Medas, and Ricardo Rocha. Forest trees for on-line data. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 632–636. ACM, 2004.

[22] Wei Dai and Wei Ji. A mapreduce implementation of c4. 5 decision tree algorithm. 2014.

[23] Yashuang Mu, Xiaodong Liu, Zhihao Yang, and Xiaolin Liu. A parallel c4. 5 decision tree algorithm based on mapreduce. *Concurrency and Computation: Practice and Experience*, 2017.

[24] José A García Gutiérrez. Líneas de investigación en minería de datos en aplicaciones en ciencia e ingeniería: Estado del arte y perspectivas. 2016.

[25] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[26] Sui-Yu Wang. Huan liu and hiroshi motoda: Computational methods of feature selection. *Pattern Analysis and Applications*, 13(2):247–249, 2010.

[27] Vinod Kumar Vavilapalli, Arun C Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, J. Lowe, Hitesh Shah, Siddharth Seth, et al. Apache hadoop yarn: Yet another resource negotiator. page 5, 2013.

[28] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.

[29] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D Ernst. The haloop approach to large scale iterative data analysis. *The International Journal on Very Large Data Bases*, 21(2):169–190, 2012.

[30] Apache Mahout. Scalable machine-learning and data-mining library. *available at mahout. apache. org*, 2008.

[31] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17(34):1–7, 2016.

# 7 Condiciones institucionales para el desarrollo de la tesis. Infraestructura y equipamiento

El desarrollo de la tesis se realizará la mayor parte del tiempo en el Departamento de Ingeniería en Sistemas de Información de la Facultad Regional Concepción del Uruguay de la UTN (FRCU), y con en el asesoramiento del Departamento de Informática de la Facultad de Ciencias Físico Matemáticas y Naturales de la Universidad Nacional de San Luis (UNSL).

Las razones de esta elección se basan en que la tesista ha realizado sus estudios de grado y posgrado en la FRCU, y está iniciando su carrera de docente-investigador en la misma como integrante del grupo de investigación GIBD (Grupo de Investigación sobre Bases de Datos). Este grupo cuenta con dos proyectos homologados en el Programa de Incentivos, uno de ellos es el denominado *Minería de datos: su aplicación a repositorios de datos masivos*, con el que se abre una nueva línea de investigación totalmente relacionada con el tema de la presente tesis.
Respecto a la infraestructura disponible el grupo de investigación posee una oficina propia que cuenta con cuatro PCs con conexión a Internet, 1 notebook, 1 Ultrabook, 1 impresora láser, escritorios y sillas. Desde el departamento de Ing. en Sistemas se cuenta con conexión directa a un servidor propio con las siguientes características: procesador Intel XEON, 32 GB de RAM, puertos Gigabit Ethernet, 2 discos SATA de 2 TB cada uno y 1 UPS de altas prestaciones.
Se cuenta con un servicio de documentación en la biblioteca de la FRCU, que incluye libros, tesis, revistas y acceso a publicaciones electrónicas como Scirus, Citeseer, IEEE, ACM entre otras. Asimismo, a través de la Biblioteca del Ministerio de Ciencia y Técnica se tiene acceso a las revistas científicas más actualizadas en la temática.