Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Programa de Maestría en Computación

Desarrollo de una metodología para la aplicación de la minería de datos en el diseño estructurado de circuitos integrados a muy gran escala (VLSI)

Tesis sometida a consideración del Departamento de Computación para optar por el grado de Magister Scientiae en Computación con énfasis en Ciencias de la Computación

Estudiante: Gustavo Montealegre Castro

Profesor Asesor: Carlos A. González Alvarado, PhD.

Cartago, Costa Rica
Noviembre, 2014

Resumen

Vivimos en un mundo en el que se genera una gran cantidad de datos todos los días, y este volumen de datos aumenta día con día. Sin embargo, nuestra capacidad para analizarlos es limitada, por lo que una gran parte de estos datos no se analizan a fondo y perdemos la oportunidad de tomar mejores *decisiones basadas en datos*.

Este problema se repite en múltiples entornos del quehacer humano, siendo uno de ellos el del diseño de circuitos integrados a muy gran escala. Precisamente en este entorno se da la motivación de crear una metodología de diseño de sistemas de minería de datos optimizados para un entorno específico.

En esta tesis se desarrolla dicha metodología, la cual se explica paso a paso, y luego se aplica en el entorno del diseño de circuitos integrados a muy gran escala. Los resultados obtenidos son presentados en detalle, demostrando tanto la efectividad de la metodología, como el beneficio obtenido al usar técnicas de minería de datos para mejorar el proceso de toma de decisiones.

El documento finaliza con recomendaciones para lograr mejoras en el sistema desarrollado, y como se puede lograr una implementación más efectiva a través de programas de entrenamiento y de modificaciones en las herramientas y bases de datos.

Palabras clave: *minería de datos, circuitos integrados, VLSI, análisis de datos, entorno*

Abstract

We live in a world in which a huge amount of data is generated on a daily basis, where even the amount of data generated daily follows an increasing trend. However, our analysis capabilities are limited, preventing us from analyzing all this data in order to use it to improve our decision making process.

This problem is present in many areas of the human endeavor, one of them being the design of very large scale integrated circuits. Precisely in this area, the idea of creating a methodology to develop data mining systems optimized for a specific area is born.

That methodology is described, step by step, in this thesis. It is then applied in the area of design of very large scale integrated circuits. The results are explained in detail, demonstrating the effectiveness not only of the methodology, but also of applying data mining techniques to improve the decision making process.

The document ends with a list of recommendations to improve the data mining system that was developed. This, through training programs as well as tool and database enhancements.

Key words: *data mining, integrated circuits, VLSI, data analysis, environment*

www.tec.ac.cr

TEC Tecnológico de Costa Rica
Programa de Maestría en Computación
Teléfono: 2550-2402

APROBACIÓN DE LA TESIS

"Desarrollo de una metodología para la aplicación de la minería de datos en el diseño estructurado de circuitos integrados a muy gran escala (VLSI)"

TRIBUNAL EXAMINADOR

Ph.D. Carlos González Alvarado
Profesor Asesor

Ph. D. José Castro Mora
Profesor Lector

Master Isaac Ramírez Herrera
Profesional Externo

Dr. Roberto Cortés Morales
Coordinador del Programa
de Maestría en Computación

Noviembre, 2014

Dedicatoria

A mi esposa Cindy y a mis hijos José Andrés, Juan Ignacio y Valeria

¡Gracias por todo su apoyo y motivación!

Agradecimientos

En primer lugar le doy gracias a Dios por darme la capacidad y empeño para poder completar esta investigación.

Al Dr. Carlos González Alvarado, quién como tutor de esta tesis me proporcionó el apoyo y la retroalimentación necesarios para completarla con la mejor calidad posible.

Al M.Sc. Isaac Ramírez Herrera y al Dr. José Castro, lectores de esta tesis.

Al M.Sc. Edwin Aguilar Sánchez, ex-Director del Programa de Maestría en Computación del TEC, quien dio un gran apoyo al convenio Intel-TEC del Programa de Maestría en Computación.

A la Br. Fabiola Arias Cordero, asistente del Proyecto Intel 6-055-3 de la Maestría en Computación del TEC, por su ayuda en la coordinación de las actividades administrativas relacionadas con esta maestría.

A todos los profesores que impartieron clases en este programa del convenio Intel-TEC por el gran compromiso demostrado al venir a Intel todas las semanas a impartir clases.

A todos mis compañeros del Primer Grupo del convenio Intel-TEC de la Maestría en Computación, en especial a Berny Alvarado Brenes quien a finales del 2010 inició con esta idea de impartir una Maestría en Computación en las instalaciones de Intel.

A mi familia y amigos, quienes con sus preguntas de "¿cómo vas con la tesis?" me motivaron a seguir adelante.

A la empresa Componentes Intel de Costa Rica por su apoya al programa de maestría en la empresa y por el apoyo económico.

Índice General

Índice de Figuras

Índice de Tablas

Capítulo 1: Introducción

1.1 Planteamiento del Problema

Uno de los ámbitos del quehacer humano que más avances ha tenido en los últimos 50 años ha sido el tecnológico. Específicamente, dentro de este campo, se puede resaltar el área de los *circuitos integrados*.

En su expresión más básica, un circuito integrado es un conjunto de transistores conectados entre sí para realizar una función específica. En la Figura 1 se muestra el diagrama esquemático de uno de los circuitos integrados más sencillos que se pueden fabricar, un *circuito inversor*. En esta figura puede observarse cómo este circuito consta de dos transistores (elementos color naranja) y de varios cables que conectan los transistores entre sí. También puede observarse que este circuito tiene una única entrada y una única salida. Por último, en el diagrama esquemático también pueden observarse las alimentaciones de poder (Vcc, Tierra) del circuito inversor.

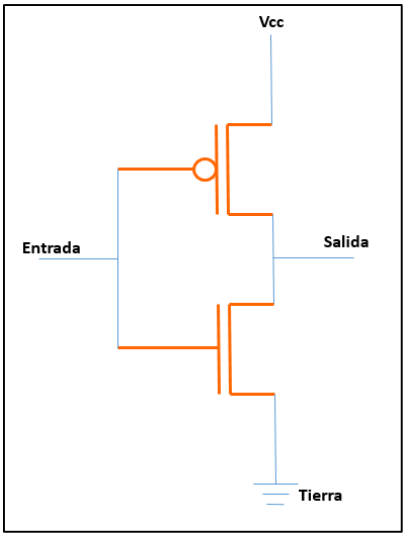

Figura 1: Esquemático de un circuito inversor

En sus inicios, los circuitos integrados contaban con unas cuantas decenas de transistores. En la actualidad, la cantidad de transistores de los circuitos integrados más complejos puede rondar los cientos de millones e incluso llegar a los billones. Existe un enunciado muy famoso conocido como la "*Ley de Moore*" (Colaboradores de Wikipedia, Moore's

Law, s.f.) que dice que "*la cantidad máxima de transistores en los circuitos integrados se duplica aproximadamente cada dos años*". Desde 1965 cuando Gordon E. Moore, co-fundador de Intel Corporation, describió esta tendencia, su predicción se ha cumplido con exactitud, tal y como se puede ver en la Figura 2.

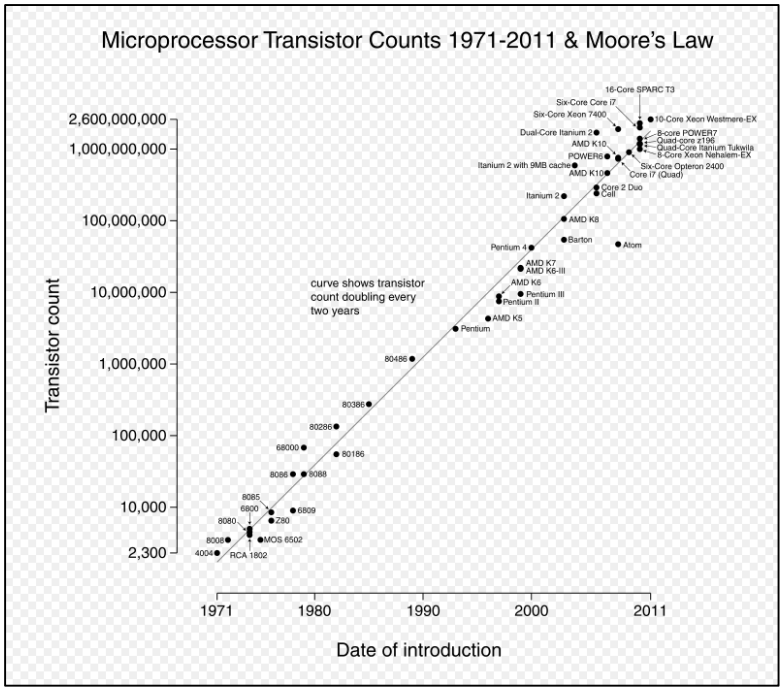

Figura 2: Ley de Moore.
Fuente: Colaboradores de Wikipedia, Moore's Law, s.f.

Es importante mencionar que la complejidad en el proceso de diseño de los circuitos integrados es directamente proporcional a la cantidad de transistores, por lo que se puede decir que la complejidad de los circuitos integrados también ha seguido la Ley de Moore. Por ejemplo, para diseñar un circuito integrado de última tecnología es necesario el uso de herramientas *CAD* (Computer Aided Design, diseño asistido por computadora por sus siglas en inglés) ya que este proceso es tan complejo que hacerlo a mano resultaría imposible.

Estas herramientas utilizan datos a nivel de cada transistor (como por ejemplo ganancia eléctrica, tiempos de retraso, carga eléctrica, consumo de potencia, dopaje y dimensiones internas, entre otros) y de cada línea de metal utilizada para conectar los transistores

(longitud, ancho, resistencia, capacitancia, inductancia) para poder realizar el diseño. Asumamos de forma conservadora que por cada transistor se generan diez datos diferentes, y que por cada línea de metal se generan cinco datos. También podemos asumir que por cada transistor necesitamos por lo menos de tres líneas de metal para conectarlo con otro transistor, de tal forma que por cada transistor (con sus líneas de metal) tendríamos un total de 25 datos diferentes. Ahora, si se está trabajando en el diseño de un circuito integrado de última tecnología, donde se tienen billones de transistores, se estaría generando al menos *25 billones* de datos cada vez que se realiza una simulación del circuito. Y, en el proceso de diseño, se deben realizar múltiples simulaciones para verificar funcionalidad, rendimiento y consumo de potencia. Es precisamente en este contexto donde se genera el problema que se pretende resolver.

El problema reside en que los diseñadores encargados del proceso de diseño de circuitos integrados modernos no le pueden sacar el máximo provecho a la gran cantidad de datos disponibles, debido a la falta de herramientas de minería de datos desarrolladas específicamente para este entorno.

Esto lleva a los ingenieros de diseño a tomar decisiones basadas en análisis parciales de datos, que van a desencadenar en circuitos integrados que podrían no ser los mejores, llevando a la producción de productos que no tendrían el mayor rendimiento posible, o que consumirían más potencia de la mínima requerida, o que utilizarían un área mayor que la menor posible.

El impacto final de esta falta de optimización es que los productos generados no van a tener las mejores características, lo que va a impactar en su competividad. Por ejemplo, si contamos con un circuito integrado que consume más potencia de la mínima necesaria, y este circuito está diseñado para ser utilizado en el mercado de los teléfonos celulares, el impacto es que la batería del teléfono se va a drenar más rápido que lo que se drenaría con un circuito integrado de la competencia, lo que va a traducirse en una desventaja competitiva.

La investigación realizada en esta tesis plantea demostrar que el uso de técnicas de minería de datos efectivamente ayuda a diseñar mejores circuitos integrados.

1.2 Justificación

Tal y como se explicó en la sección anterior, en el proceso de diseño estructurado de circuitos VLSI (Very Large Scale Integration, integración a muy gran escala por sus siglas en inglés) se genera una gran cantidad de datos. Para darse una idea de la magnitud en el volumen de estos datos, usemos como ejemplo un microprocesador actual. El producto Haswell de Intel cuenta con aproximadamente 1.4 billones de transistores (Colaboradores de Wikipedia, Haswell (microarchitecture), s.f.). Para cada uno de estos transistores se pueden generar datos de cientos de parámetros (características eléctricas, dimensiones físicas, propiedades de los materiales), y esto se repite múltiples ocasiones durante el proceso de diseño. Estos datos, si son analizados de la forma correcta, pueden ayudar a los ingenieros en optimizar los procesos de diseño, y obtener así el mejor producto posible. El problema reside en que, por la gran cantidad de datos, este análisis es muy dificil de realizar.

Con este trabajo de investigación se demuestra que al utilizar técnicas de minería de datos se puede hacer el análisis de los datos generados por el proceso de diseño estructurado de circuitos VLSI, de forma tal que el proceso de diseño se ve altamente beneficiado.

Estos beneficios pueden verse reflejados en diferentes parámetros relacionados con el proceso de diseño de circuitos VLSI, como por ejemplo tamaño del diseño (utilización del área de silicio), tiempo de diseño, consumo de potencia y frecuencia de operación. Cada uno de estos parámetros puede tener un impacto muy importante en la competitividad de las empresas dedicadas al diseño y fabricación de circuitos VLSI, especialmente por el alto nivel de competencia que tienen las empresas que se dedican al desarrollo de este tipo de productos.

1.3 Hipótesis

La hipótesis que se planteó es que, al aplicar la metodología desarrollada en este trabajo para el diseño de sistemas de minería de datos en el entorno del diseño estructurado de circuitos VLSI, se puede mejorar el proceso de diseño, dando como resultado mejores circuitos integrados.

Para probar esta hipótesis se siguió un proceso exhaustivo de investigación, desarrollo y validación, el cual está documentado de la siguiente forma:

- *Capítulo 2: Marco Teórico*. En este capítulo se explican conceptos teóricos relacionados con el diseño de circuitos integrados y con la minería de datos.
- *Capítulo 3: Metodología de Desarrollo de un Sistema de Minería de Datos para un Entorno Específico*. Se presenta la metodología desarrollada con el objetivo de crear un sistema de minería de datos optimizado para las características de un entorno específico.
- *Capítulo 4: Desarrollo de la Aplicación de Minería de Datos para el Entorno VLSI*. Se detalla el proceso seguido al implementar la metodología del capítulo anterior en el entorno de diseño estructurado de procesadores gráficos de última tecnología de Intel. Asimismo, se desarrolla un procedimiento de comparación de herramientas de minería de datos que ayuda a decidir si herramientas existentes pueden cumplir con las necesidades específicas del entorno para el cual se está desarrollando la aplicación. Finalmente se propone la arquitectura del sistema de minería de datos que se va a desarrollar.
- *Capítulo 5: Análisis de Resultados*. Aquí se presentan los resultados obtenidos al implementar el sistema de minería de datos en el entorno VLSI. Se utiliza la metodología CRISP-DM (sección 2.2.5) para explicar los análisis realizados, ahondando en el entendimiento del negocio y de los datos, del modelo de minería de datos utilizado y de los resultados obtenidos.
- *Capítulo 6: Recomendaciones y Conclusiones*. Se presentan las recomendaciones de mejora para el sistema de minería de datos así como las conclusiones a las que se llegó luego de completar esta investigación.

Es importante mencionar que por la confidencialidad inherente a las características de los productos que son desarrollados en Intel Costa Rica, sus detalles específicos no pueden ser publicados. Sin embargo, para demostrar que las técnicas de minería de datos efectivamente generan una mejoría, se van a presentar datos relativos. Por ejemplo, en lugar de publicar que el parámetro "X" cambió de un valor inicial de 10 a un valor final de 8, se publica que el parámetro "X" tuvo una mejoría de 20%. De esta forma no se violan los acuerdos de confidencialidad que se tienen con Intel, y esta tesis mantiene su valor académico puesto que se describe la magnitud de la mejoría relativa.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar una metodología para la aplicación de la minería de datos en el diseño estructurado de circuitos VLSI.

1.4.2 Objetivos Específicos

- **Objetivo Específico 1**: Desarrollar una metodología de caracterización de entornos que permita la implementación de un sistema de minería de datos que facilite el proceso de toma de decisiones en dicho entorno.
- **Objetivo Específico 2**: Desarrollar e implementar un sistema de minería de datos utilizando la metodología desarrollada como parte del Objetivo Específico 1 para el entorno de diseño estructurado de circuitos VLSI.
- **Objetivo Específico 3**: Desarrollar una metodología de comparación de herramientas de minería de datos que facilite la escogencia de la herramienta que mejor cumpla con un conjunto de requisitos específicos. Esto con el propósito de poder utilizar herramientas existentes antes de tener que desarrollar de una herramienta nueva.
- **Objetivo Específico 4**: Demostrar que la aplicación de técnicas de minería de datos efectivamente mejoran el proceso de diseño de circuitos VLSI.

- **Objetivo Específico 5**: Generar una lista de recomendaciones necesarias para que el sistema de minería de datos desarrollado pueda implementarse eficientemente en un entorno comercial.

Capítulo 2: Marco Teórico

Los fundamentos teóricos incluyen información sobre el diseño de circuitos integrados VLSI e información de minería de datos, ya que ambos campos se van a intergrar al desarrollar un sistema de minería de datos para el entorno de circuitos VLSI.

2.1 Diseño de Circuitos Integrados VLSI

En esta sección se presenta una descripción de lo que es el diseño de circuitos integrados VLSI, específicamente del diseño basado en celdas. Se inicia dando una breve explicación de lo que son los circuitos integrados y como éstos han evolucionado a través del tiempo. Luego, se describe en detalle las diferentes etapas que se deben seguir para desarrollar circuitos VLSI, desde la definición de la funcionalidad del circuito hasta la producción de los mismos. Se hace énfasis en la complejidad del proceso de diseño de circuitos integrados modernos, y cómo esta complejidad está directamente relacionada con la cantidad de transistores del circuito integrado. También, se resalta la forma en que los diseñadores deben utilizar una gran cantidad de datos para poder tomar decisiones que van a generar un producto altamente competitivo.

2.1.1 Circuitos Integrados

Un *circuito integrado* es un conjunto de componentes electrónicos que se encuentran empotrados en un elemento de material semiconductor, generalmente silicio (Colaboradores de Wikipedia, Integrated Circuits, s.f.). El principal componente electrónico que se incluye en un circuito integrado es el *transistor*, cuyo funcionamiento puede compararse al de un interruptor eléctrico que permite o evita el paso de la corriente eléctrica (Colaboradores de Wikipedia, Transistor, s.f.).

Estos circuitos se utilizan en prácticamente todos los equipos electrónicos modernos, desde un horno de microondas hasta una supercomputadora. Esta gran penetración en nuestro medio se debe en gran parte a las mejoras tan significativas que se han dado en el proceso de manufactura de estos circuitos, las que permiten integrar un número cada vez mayor de transistores por unidad de área.

Los primeros circuitos integrados empezaron a fabricarse a inicios de la década de los sesentas e integraban como máximo decenas de transistores. Estos primeros circuitos integrados fueron llamados **SSI** (*Small Scale Integration*, Integración a Pequeña Escala por sus siglas en inglés). Con el avance en las técnicas de manufactura se logró aumentar la cantidad de transistores, y se pasó de cientos de transistores a finales de los setentas con los circuitos **MSI** (*Medium Scale Integration*, Integración a Mediana Escala por sus siglas en inglés) a miles de transistores en los circuitos **LSI** (*Large Scale Integration,* Integración a Gran Escala por sus siglas en inglés) a mediados de la década de los setentas (Colaboradores de Wikipedia, Integrated Circuits, s.f.).

Actualmente, se habla de **VLSI** (*Very Large Scale Integration,* Integración a Muy Gran Escala por sus siglas en inglés), en donde los circuitos integrados llegan a contar con miles de millones de transistores.

Para alcanzar estos niveles de integración los procesos de manufactura han evolucionado exponencialmente, requiriendo la creación de estándares para el diseño de nuevos productos. Este proceso de diseño se describe en la siguiente sección.

2.1.2 Diseño de Circuitos VLSI

El diseño de circuitos VLSI sigue los mismos pasos básicos que el diseño de cualquier producto, a saber:

1. Definir funcionalidad requerida
2. Diseño de la implementación
3. Validación y pruebas
4. Producción

A continuación, se va a describir cada una de estas etapas, elaborando en las optimizaciones específicas para productos VLSI. También se van a mencionar detalles específicos de los microprocesadores, al ser estos productos el enfoque de este trabajo.

La primera etapa es donde se define la funcionalidad del circuito integrado. Esta es la etapa inicial del diseño de cualquier producto. Generalmente, se tienen que considerar diferentes factores como necesidades del mercado, productos de la competencia, tiempo estimado de desarrollo, complejidad del desarrollo, entre otros; con el fin de desarrollar un producto que pueda salir al mercado en un tiempo aceptable y que genere ganancias para la empresa.

En el caso de los microprocesadores (McFARLAND, 2006), una vez que se define la funcionalidad del mismo, el siguiente paso es definir la *arquitectura* que el microprocesador va a utilizar. Cada microprocesador se diseña para soportar un número finito de instrucciones, las cuales se codifican como números binarios que van a ser interpretados por el procesador. Esta lista de instrucciones, sus funciones y codificación es lo que conforma la arquitectura. Es importante notar que cualquier programa que esté escrito para una arquitectura específica va a poder ejecutarse en todo procesador que utilice esa arquitectura. Por esta razón la selección de la arquitectura es una decisión de suma importancia, ya que dependiendo de la arquitectura que se seleccione se va a definir si el procesador es compatible con programas de software ya existentes en el mercado. Una práctica común es la de agregar nuevas instrucciones a una arquitectura existente, mientras que se mantiene el soporte a todas las instrucciones ya presentes. Esto se conoce como una *extensión* a la arquitectura, y con estas extensiones se logran mejoras en la ejecución de nuevos programas, mientras que se mantiene compatibilidad con los programas anteriores. Algunos ejemplos de arquitecturas existentes son la x86 (procesadores Pentium 4 y Athlon XP), VAX (procesadores Micro VAX 78032) y EPIC (procesadores Itanium 2).

Una vez que la arquitectura está definida, se procede a definir la *micro-arquitectura*. Si la arquitectura define las instrucciones que se pueden ejecutar, la micro-arquitectura define la forma en que esas instrucciones son ejecutadas. De esta forma, los cambios en arquitectura son visibles para los programadores ya que implican nuevas instrucciones, pero los cambios en micro-arquitectura no son visibles para los programadores. Sin embargo, la micro-arquitectura tiene un gran impacto en la eficiencia del procesador, impactando la velocidad de operación, el consumo de potencia y el área.

En la segunda etapa, se completa el diseño del producto final. Para los productos VLSI, el método de diseño más utilizado es conocido como "*Diseño a base de Celdas Estándar*" (Colaboradores de Wikipedia, Application specific integrated circuits, s.f.). Este método consiste de los siguientes pasos:

1. *Diseño Lógico*. Se construye un modelo del producto final utilizando un lenguaje *HDL* (Hardware Description Language, Lenguaje Descriptivo del Hardware por sus siglas en inglés). Este paso es similar a desarrollar un programa de software, y la funcionalidad del producto puede ser verificada por medio de simulaciones. Esta implementación es conocida como *diseño RTL* (Register Transfer Level, Nivel de Transferencia de Registros por sus siglas en inglés). En el caso de los microprocesadores, los HDLs más comúnmente utilizados son *Verilog*, *System Verilog* y *VHDL*.

2. *Diseño Estructurado*, también conocido como Diseño del Circuito (McFARLAND, 2006). En esta etapa se crea una implementación a nivel de transistores de la lógica definida en el RTL. Se utilizan librerías de celdas estándar para pasar de una funcionalidad específica a la mejor implementación a nivel de transistores (por ejemplo, pasar de una Y lógica a una compuerta AND). En esta etapa se utilizan herramientas de software que logran optimizar la implementación en transistores y de esta forma obtener los mejores valores de frecuencia de operación y consumo de potencia. Sin embargo, los diseños generados por estas herramientas siempre presentan problemas que deben ser resueltos por los ingenieros de diseño. La solución a estos problemas requiere del análisis de múltiples variables que ayudarán a los ingenieros a seleccionar las mejores celdas a utilizar, su localización óptima y la mejor forma de interconectarlas. Por su complejidad, esta etapa se puede subdividir en (Colaboradores de Wikipedia, Application specific integrated circuits, s.f.):
    a. *Síntesis*: En esta etapa se escogen las celdas estándar que van a utilizarse para implementar la funcionalidad especificada en el RTL. Además, se

busca minimizar la cantidad de transistores que se van a necesitar para implementar la funcionalidad definida en el RTL.

b. *Colocación*: En este paso se define la topología del circuito, colocando los transistores en la posición física en donde se van a encontrar en el silicio. El objetivo de esta etapa es minimizar el área física del producto.
c. *Ruteo*: Se interconectan las diferentes celdas lógicas tal y como está definido en el RTL. Esta etapa es crítica para el correcto funcionamiento del circuito integrado, y se tienen que respetar múltiples reglas de diseño cuyo objetivo es reducir la cantidad de errores de fabricación del producto. Uno de los objetivos principales de esta etapa es minimizar los recursos de ruteo (i.e. cables), ya que entre más "cables" se utilicen, mayor va a ser la resistencia y capacitancia del circuito, y esto va a tener consecuencias negativas en el rendimiento del mismo.

3. *Diseño del Layout*. En esta etapa se diseñan las máscaras litográficas que se van a utilizar en el proceso de fabricación del circuito integrado. Estas máscaras se usan para revelar material fotoresistente y de esta forma construir los transistores y líneas de metal que eventualmente van a formar el circuito integrado.

Durante las tres etapas del diseño de la implementación se están llevando a cabo simulaciones constantes de la operación del circuito para obtener valores de frecuencia de operación, consumo de potencia y de posibles fallas durante la fabricación de los circuitos. Estos datos son utilizados por los ingenieros de diseño para optimizar el circuito, y de esta forma mejorar el rendimiento del producto final. Sin embargo, por la gran cantidad de datos que se generan, muchas veces los ingenieros de diseño solo pueden analizar una pequeña parte de los mismos, por lo que el diseño final puede no ser el más óptimo, sino simplemente un diseño que cumple con los requisitos mínimos de operación.

También, es importante mencionar que estas simulaciones demandan gran cantidad de recursos computacionales, razón por la cual se corren para simular las condiciones de uso

más comunes, dejando las pruebas más exhaustivas a la validación *post-silicio*, que es cuando ya se tiene un producto fabricado.

La tercera etapa de este proceso es donde se realizan las validaciones y pruebas de funcionalidad, calidad y confiabilidad. Estas pruebas del circuito integrado se pueden hacer en dos momentos diferentes del ciclo de diseño de un producto VLSI:

1. *Pre-Silicio*: Todas las pruebas se basan en simulaciones ya que en este momento todavía no se dispone de un producto físico. Estas simulaciones son muy demandantes desde un punto de vista de recursos computacionales, razón por la cual solo se hacen las pruebas más importantes. Los datos generados por estas pruebas son utilizados por los ingenieros para mejorar sus diseños, y siguiendo un proceso iterativo logran mejorar el producto.

2. *Post-Silicio*: En este momento ya se dispone de un producto terminado, y este producto puede probarse para encontrar problemas en el diseño o en el proceso de fabricación. Muchas veces se logran encontrar problemas en el diseño que se manifiestan solo bajo ciertas condiciones de uso (por ejemplo, a un voltaje y/o temperatura específicos), que llevan a realizar cambios en el diseño. Durante esta etapa también se encuentran problemas en el proceso de fabricación que llevan a definir nuevas reglas de diseño que pueden generar cambios significativos en la implementación y forzar cambios en las etapas de colocación o ruteo.

La última etapa es la de *producción*. Una vez que se ha diseñado un producto que cumple la funcionalidad esperada, y se ha demostrado por medio de validaciones y pruebas que la calidad y confiabilidad del mismo cumple con los requerimientos esperados, se procede a producirlo.

El primer proceso de la etapa de producción se conoce como *fabricación*, y este inicia con una oblea de material semiconductor, generalmente silicio puro. En la actualidad, estas obleas tienen un diámetro máximo de unos 300mm y un espesor de 400-600µm

(Colaboradores de Wikipedia, Fabricación de circuitos integrados, s.f.). Se inicia alterando las propiedades eléctricas del silicio mediante un proceso llamado *dopaje*, mediante el cual se bombardean iones en áreas específicas de la oblea de silicio para aumentar o disminuir su conductividad.

Luego se "imprimen" los transistores y líneas de metal en el silicio por medio del proceso de fotolitografía. Los pasos básicos del proceso de fotolitografía se describen a continuación (Colaboradores de Wikipedia, Fotolitografia, s.f.):

1. *Preparación del sustrato*: se deposita una capa de metal conductor sobre la oblea de silicio (Figura 3a).
2. *Aplicación de resinas fotoresistentes*: Se aplica sobre la capa metálica una capa de resina fotosensible, que es una sustancia que cambia sus características químicas luego de la exposición a la luz, que generalmente es radiación ultravioleta (Figura 3b).
3. *Calentamiento Débil*: La oblea se calienta ligeramente para curar las resinas y fijarlas a la capa de metal.
4. *Exposición a la Luz*: La oblea se expone a una fuente de luz a través de una máscara que tiene los patrones que se desea "imprimir" sobre la resina fotoresistente. De esta forma se cambian las características químicas solamente de ciertas partes de la capa de fotoresistencia (Figura 3c). Cuanto menor es la longitud de onda de la luz utilizada, mayor es la resolución que se puede obtener. Por esta razón es que se utiliza luz ultravioleta. Sin embargo, en los procesos actuales, se está cambiando a utilizar luz láser y otras fuentes de luz con menor longitud de onda por la resolución que se requiere.
5. *Revelado*: Mediante un proceso químico se elimina la fotoresistencia que fue expuesta a la luz (Figura 3d).
6. *Calentamiento Fuerte*: Se cura completamente la resina fotoresistente para fijarla sobre la capa de metal.

7. *Limpieza con agentes químicos*: la oblea se trata con agentes químicos (usualmente ácidos) que reaccionan con el metal pero no con la resina fotoresistente, de tal forma que se elimina la capa de metal en las áreas expuestas (Figura 3e).
8. *Pulido*: Por último, con un proceso de pulido, se elimina la resina fotoresistente dejando expuesta la capa de metal (Figura 3f).

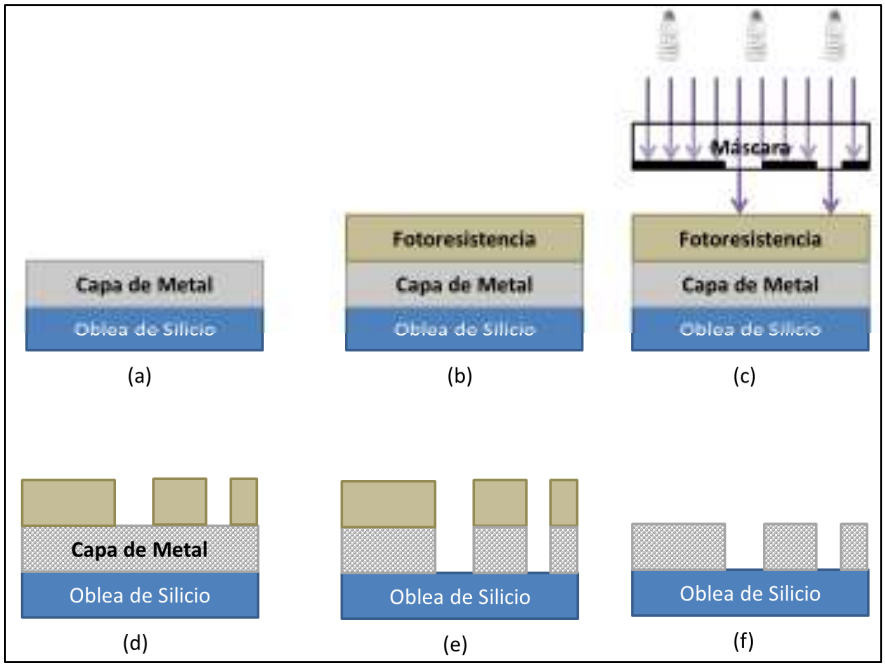

Figura 3: Proceso de fotolitografía

Este proceso se repite múltiples veces hasta formar los transistores y las capas de metal que sirven para interconectar los transistores entre sí.

Una vez que los circuitos integrados están impresos en la oblea, viene el proceso de ensamblaje. El primer paso de este proceso es el de separar cada circuito integrado. Esto se logra al cortar la oblea siguiendo una cuadrícula que separa cada circuito integrado. Una vez separados, se procede a ensamblar el circuito integrado en su paquete respectivo. Esta etapa es de suma importancia ya que los circuitos integrados no pueden utilizarse directamente una vez que han sido separados de la oblea, ya que sería muy difícil

conectarlos con la tarjeta electrónica en la que se tienen que montar y además tienen muy poca resistencia mecánica, lo que los haría muy frágiles.

La última etapa es la de las pruebas eléctricas. Aquí, el circuito integrado se prueba para verificar su correcto funcionamiento. Una vez que se ha verificado que efectivamente funciona como se espera, el circuito integrado está listo para ser vendido.

Es común que incluso durante esta etapa se sigan haciendo cambios al diseño para mejorar el rendimiento o reducir la cantidad de defectos.

Este proceso de diseño de circuitos VLSI se resume en la Figura 4.

Figura 4: Proceso de diseño de circuitos integrados VLSI

La presente investigación involucra tanto la ingeniería electrónica como las ciencias de la computación. Se considera que muchos de los lectores pueden tener una formación más fuerte en la segunda área, razón por la cual se presenta una analogía entre el diseño de circuitos VLSI y el diseño de software. Esta analogía puede facilitar el entendimiento del diseño de circuitos VLSI (Tabla 1).

Diseño circuitos VLSI	Desarrollo de Software	Comentarios
Diseño Funcional	Recolección de requerimientos y selección de lenguaje de programación	*Se define la funcionalidad del producto final. También se selecciona la forma en que estas funciones van a desarrollarse (lenguaje de programación, arquitectura, micro-arquitectura)*
Diseño Lógico	Programación	*Se implementa la funcionalidad en el lenguaje seleccionado (ej. HDL, C#).*
Diseño Estructurado	Compilación	*El producto se transforma a un formato tal que se puede ejecutar*
Diseño del Layout		
Debug Post-Silicio	Validación y pruebas (también conocido como QA)	*Se verifica la funcionalidad del producto*

Tabla 1: Analogía entre diseño de circuitos VLSI y desarrollo de software.

Ahora que se ha explicado la teoría detrás del diseño de circuitos VLSI, se va a proceder a explicar los conceptos básicos de minería de datos.

2.2 Minería de Datos

En esta sección se va a presentar una descripción de alto nivel de lo que es la minería de datos y de lo que se puede lograr con la misma.

2.2.1 Definición

La minería de datos puede definirse como "*el proceso por medio del cual se intentan descubrir **patrones** ventajosos en grandes cantidades de **datos***" (GONZALEZ, 2012). Esta definición tiene dos conceptos importantes:

- *Patrones*: características observables de un elemento o conjunto de elementos.
- *Datos*: los datos pueden estar almacenados en bases de datos, datawarehouses, archivos planos o cualquier otro tipo de repositorio de datos. Los datos pueden ser continuos, discretos, texto, multimedia, estructurados, no estructurados, etc.

Es importante mencionar que la minería de datos es diferente del análisis de datos, ya que en el análisis de datos se busca validar una hipótesis con los datos, mientras que en la minería de datos se busca descubrir patrones a partir de los datos (HAN J. , 1999).

2.2.2 Proceso del Descubrimiento de Conocimiento

La minería de datos es un componente crítico en el proceso del descubrimiento de conocimiento a partir de los datos. Este conocimiento nos ayuda a tomar mejores decisiones.

En general, el proceso del descubrimiento de conocimiento a partir de datos es un proceso iterativo que incluye los siguientes pasos (GONZALEZ, 2012):

1. *Limpieza de los Datos*: se remueve ruido e inconsistencias en los datos.
2. *Integración de Datos*: se combinan varias fuentes de datos en una sola
3. *Selección de los Datos*: se extraen los datos relevantes para el análisis en cuestión.
4. *Transformación de los Datos*: los datos se consolidan por medio de operaciones de resumen y/o agregación para facilitar el proceso de minería de datos
5. *Minado de Datos*: se aplican una serie de métodos por medio de los cuales se pueden extraer patrones de los datos
6. *Evaluación de los Patrones*: se identifican los patrones interesantes para el análisis
7. *Presentación del conocimiento*: se utilizan técnicas de visualización para presentar el conocimiento encontrado y de esta forma ayudar a los usuarios finales a tomar decisiones.

La Figura 5 resume el proceso del descubrimiento de conocimiento.

Figura 5: Proceso del Descubrimiento de Conocimiento

2.2.3 Tareas Principales en el Minado de Datos

Las tareas de minado de datos se pueden clasificar en dos grandes categorías: *descriptivas* y *predictivas*. Las descriptivas generan resúmenes de los datos y presentan las propiedades generales de los datos. Las predictivas se usan para construir modelos que nos pueden ayudar a predecir el comportamiento de nuevos grupos de datos.

Un sistema de minería de datos puede cumplir una o varias de las siguientes tareas (HAN J. , 1999):

- *Descripción de clases*: se proporciona un resumen conciso de los datos de una clase y como esa clase se distingue de las otras clases.
- *Asociación*: es el descubrimiento de relaciones (o correlaciones) entre varios elementos. Un tipo de tarea de asociación es la búsqueda de patrones frecuentes en un conjunto de datos, tales como grupos de ítems o patrones secuenciales.
- *Clasificación*: se utiliza un grupo de datos de entrenamiento para crear reglas de clasificación y de esta forma clasificar los datos nuevos. Por ejemplo, se podría generar un modelo que diagnostique la enfermedad de un paciente basándose en los síntomas que presenta.
- *Predicción*: se genera un modelo que puede predecir valores faltantes en el conjunto de datos. Por ejemplo, se puede predecir el salario de un empleado al conocer los salarios de otros empleados.
- *Agrupación*: se buscan clústeres dentro del conjunto de datos, en donde un clúster es un conjunto de elementos similares. El análisis de datos atípicos o valores

extremos (outliers en inglés) puede considerarse como un tipo de análisis de agrupación.

- *Análisis de series de tiempo*: se buscan señales relacionadas con el tiempo, como por ejemplo tendencias, comportamientos periódicos, etc.

En la Figura 6 se muestra un resumen de estas tareas, y cómo ellas se dividen entre descriptivas y predictivas.

Figura 6: Tareas Principales en el minado de datos

Existen otras tareas que también podrían ser implementadas por un sistema de minería de datos, como por ejemplo *análisis de valores anómalos*, *análisis de variabilidad*, etc.

Otra clase de análisis que comúnmente se realiza durante el proceso de minado de datos es lo que se conoce como *OLAP* (Online Analytical Processing, procesamiento analítico en línea por sus siglas en inglés). Las operaciones OLAP agrupan los datos en diferentes niveles de abstracción. Se utilizan estructuras multidimensionales conocidas como cubos que contienen datos resumidos a diferentes niveles. OLAP ayuda a analizar los datos de una forma rápida siempre y cuando los cubos se hayan implementado de forma correcta y se hayan definido técnicas de pre-calculado de resúmenes (Colaboradores de Wikipedia, OLAP, s.f.) (HAN & KAMBER, 2006).

Como se puede inferir de la lista de tareas, el proceso de minería de datos incorpora múltiples disciplinas, entre las que se pueden mencionar tecnologías de bases de datos, estadística, aprendizaje de máquinas, visualización de datos e inteligencia artificial.

Una vez que se conocen las diferentes tareas del minado de datos, es necesario entender la forma en que estas tareas pueden ser implementadas por medio de un sistema de minado de datos.

2.2.4 Sistema de Minería de Datos

Un sistema de minería de datos consta de varios elementos. En forma general, estos elementos son:

- *Base de Datos*: Es el repositorio en donde se guardan los datos. Puede ser una (o varias) base de datos, un data warehouse, un conjunto de archivos, o cualquier otro tipo de repositorio de datos.
- *Herramienta de Minería de Datos*: Es el conjunto de operaciones que se aplican a los datos para extraer conocimiento. Actualmente, muchos DBMS (Database Management System, sistemas de manejo de bases de datos por sus siglas en inglés) modernos incluyen procedimientos de minería de datos dentro de sus funciones.
- *Interfaz de Usuario*: Como su nombre lo indica, es el componente que permite a los usuarios interactuar con el sistema. Puede ser por medio de una aplicación cliente o de una página web. También, puede permitir a los usuarios implementar diferentes técnicas de visualización de datos para facilitar el proceso de descubrimiento de conocimiento.

Adicionalmente, es deseable que exista un esquema de integración bien definido entre los diferentes elementos del sistema, logrando de esta forma un sistema estrechamente acoplado que va a tener un mejor rendimiento. De esta forma, el proceso de extracción de datos y posterior análisis por parte de la herramienta de minería de datos va a ser altamente eficiente, asegurándose que los datos se encuentran en el formato correcto y la cantidad de transformaciones requeridas sean mínimas o nulas. También, una vez que la herramienta de minería de datos ha completado su análisis, la visualización de los resultados va a realizarse de forma rápida y eficiente.

Con respecto a las bases de datos, algunas de las diferentes tareas de minado de datos (sección 2.2.3) pueden optimizarse si se utiliza un tipo de base de datos específico (HAN & KAMBER, 2006). La Tabla 2 muestra los tipos de bases de datos óptimos para diferentes tareas de minado de datos. Esta tabla incluye únicamente los tipos de bases de datos más comúnmente usados, por lo que no se puede descartar el uso de un tipo diferente de base de datos. Incluso, es posible usar un tipo de base de datos no óptimo para una tarea específica si es lo más recomendable para la aplicación que se está desarrollando. De tal forma que sería posible usar un Data Warehouse para hacer una tarea predictiva si esto fuera lo mejor para la aplicación de minería de datos que se está desarrollando.

Tarea de Minado de Datos	Tipo de Base de Datos Óptimo[1]			
	Relacional	Data Warehouse	Transaccional	NoSQL
Descripción de Clases	X	X		X
Asociación	X		X	X
Clasificación	X			X
Predicción	X			X
Agrupación	X			X
Análisis de Series de Tiempo	X	X		X
OLAP		X		

Tabla 2: Tipos de bases de datos óptimos según la tarea de minado de datos por realizar

Al ser la minería de datos una disciplina relativamente nueva, pueden existir diferentes formas de ejecutar sus tareas. Por esta razón es importante definir estándares que aseguren consistencia. CRISP-DM es uno de estos estándares, y va a ser explicado a continuación.

2.2.5 CRISP-DM

CRISP-DM (CRoss-Industry Standard Process for Data Mining, proceso de minado de datos estándar para múltiples industrias por sus siglas en inglés) es una metodología estándar de minería de datos que puede ser utilizada independientemente de las

[1] Estos tipos de bases de datos se refieren tanto al DBMS como a la funcionalidad de la base de datos. De esta forma, la base de datos transaccional se refiere a una base de datos donde se almacenen principalmente datos transaccionales, sin importar si el DBMS es relacional o no. En el Apéndice 1: Tipos de Bases de Datos Comunes se presentan más detalles sobre estos tipos de bases de datos.

herramientas que se usen. Este proceso fue desarrollado en 1999 por un grupo de compañías lideradas por el fabricante de automóviles Daimler-Benz (NORTH, 2012) (Wikipedia, s.f.)

El proceso consiste de seis pasos, tal y como se puede ver en la Figura 7.

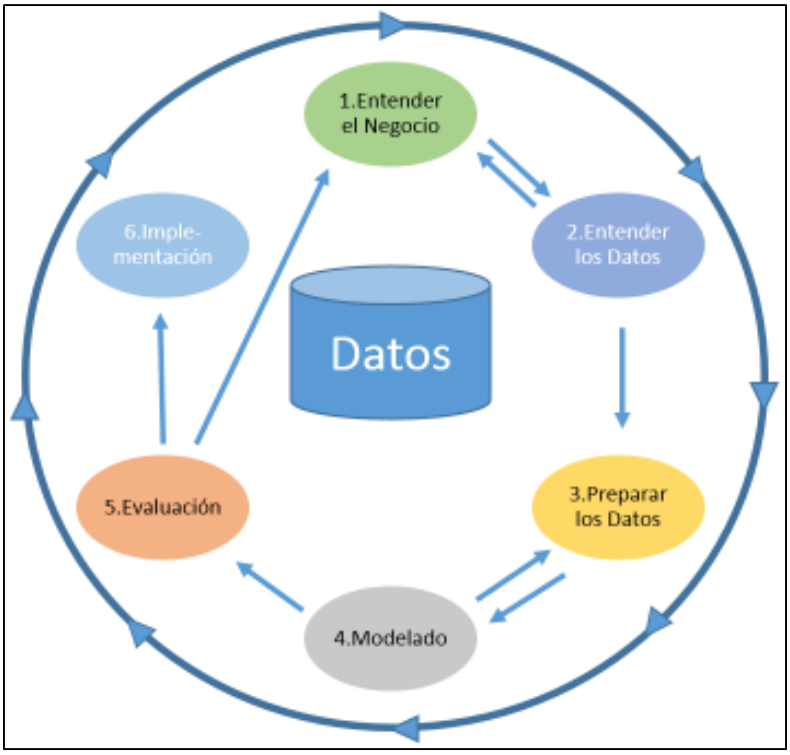

Figura 7: Modelo CRISP-DM

1. *Entender el Negocio*: se busca entender las metas, los objetivos, las preguntas y tratar de predecir los resultados que se van a obtener del proceso de minado de datos. Se quiere entender la razón por la que se va a desarrollar la actividad de minado de datos. Las siguientes preguntas deben ser respondidas después de este paso:
    - ¿Cuál es el objetivo de esta actividad para el negocio?
    - ¿Cómo se realiza esta actividad?
    - ¿Cuáles datos es necesario recolectar, cuáles son deseables y cuáles no son requeridos?
    - ¿Quién va a tener acceso a los datos, y cómo los va a accesar?

2. *Entender los Datos*: se desea lograr un entendimiento de los datos que están disponibles y la forma en que los mismos se recolectan. Se comprende el origen de los datos, la forma en la que se han recolectado y el significado de los mismos. Después de este paso se debe ser capaz de responder las siguientes preguntas:
    - ¿Cuáles son las características de la base de datos que se va a utilizar?
    - ¿Cómo se cargan los datos a la base de datos?
    - ¿Cómo funciona el sistema de seguridad (i.e. control de acceso) para esta base de datos?
    - ¿Cuáles son los mecanismos que se utilizan para asegurar integridad de los datos, y cuál es la integridad actual de los mismos?
    - ¿Qué problemas de consistencia tenemos en los datos? Por ejemplo, un problema de consistencia podría darse si al recolectar información de ciudad se utilizara un campo de texto libre, donde los usuarios podrían digitar datos como S.J., SJ, San Jose, San José, Sna Jose. Algunos de estos problemas pueden arreglarse implementando validaciones durante la generación de los mismos. Otros van a tener que resolverse en el siguiente paso: preparar los datos.

3. *Preparar los Datos*: se arreglan los datos de tal forma que estén listos para el proceso de minado de datos. En este paso se pueden desarrollar tareas como eliminación de valores extremos, manejo de datos faltantes o inconsistentes, cambios de formatos, configuración de tipos de datos, etc.

4. *Modelado*: En esta etapa se desarrollan los modelos de minado de datos que se van a implementar. Se define que tareas de minado se van a usar y como se van a implementar.

5. *Evaluación*: Se revisan los resultados del modelo de minado de datos, se interpretan los resultados y se decide si son útiles. En el caso de obtener resultados inesperados, se debe investigar y entender la razón de los mismos.

6. *Implementación*: Es el paso final. En este se toma alguna acción como resultado de lo que hemos aprendido de nuestro análisis. La implementación puede ocurrir en etapas, como por ejemplo iniciar con un plan piloto para luego proceder a implementarlo en toda la organización.

2.3 Antecedentes Históricos

Basado en la investigación bibliográfica que se realizó, no se encontró ningun trabajo relacionado con la implementación de minería de datos para optimizar el diseño estructurado de circuitos VLSI.

Esto puede deberse a múltiples razones, entre las cuales se considera que las dos más probables son las siguientes:

1. *Confidencialidad*. El mercado de los circuitos integrados es altamente competitivo. Es posible que empresas específicas ya hayan desarrollado aplicaciones de minería de datos pero no las hayan compartido para no perder las ventajas que estas técnicas de análisis de datos y toma de decisiones les dan con respecto a su competencia.

2. *Presión de Tiempo*. La industria de los circuitos integrados tiene cronogramas de desarrollo altamente agresivos para poderse mantener alineados con la "Ley de Moore". Las empresas líderes en este campo, que son las que pueden tener los recursos para desarrollar sistemas avanzados de minería de datos, no disponen del tiempo necesario para desarrollarlos.

En el siguiente capítulo se va a desarrollar una metodología para crear un sistema de minería de datos optimizado para un entorno específico. Luego, esta metodología va a aplicarse al entorno de desarrollo de circuitos VLSI.

Capítulo 3: Metodología de Desarrollo de un Sistema de Minería de Datos

3.1 Desarrollo de un Sistema de Minería de Datos para un Entorno Específico

En este documento, un *entorno* se define como el conjunto de elementos que componen un área específica del quehacer humano y comprende los elementos operacionales que permiten que una tarea específica se lleve a cabo, junto con la información requerida para tomar decisiones relacionadas con el crecimiento y desarrollo de ese entorno.

Ejemplos de entornos podrían ser el entorno médico, el entorno de telecomunicaciones, el entorno de ventas de ropa, etc. En el Apéndice 2: se mencionan varios entornos para los cuales se han desarrollado aplicaciones de minería de datos.

Dentro de un entorno también es posible definir sub-entornos, de tal forma que se puede llegar a formar una jerarquía. De esta forma, dentro del entorno de telecomunicaciones se puede llegar a tener diferentes niveles jerárquicos de sub-entornos, tal y como se puede observar en la Figura 8.

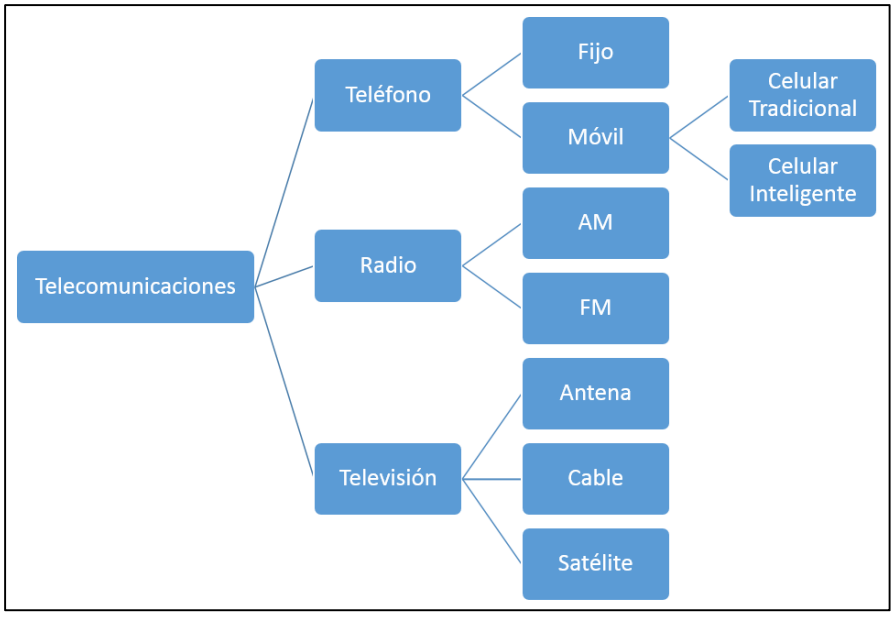

Figura 8: Jerarquía de entornos en el campo de las telecomunicaciones

Para caracterizar un entorno (o un sub-entorno) específico, es necesario contar con un conocimiento detallado de las actividades que se realizan en el mismo, del tipo de decisiones que se llevan a cabo, de los actores que participan en las diferentes actividades específicas de ese entorno, etc.

Sin embargo, antes de iniciar el proceso de caracterización del entorno es necesario completar varias tareas importantes. Estas tareas se aprecian en la Figura 9, que muestra la propuesta metodológica para desarrollar una aplicación de minería de datos para un entorno específico.

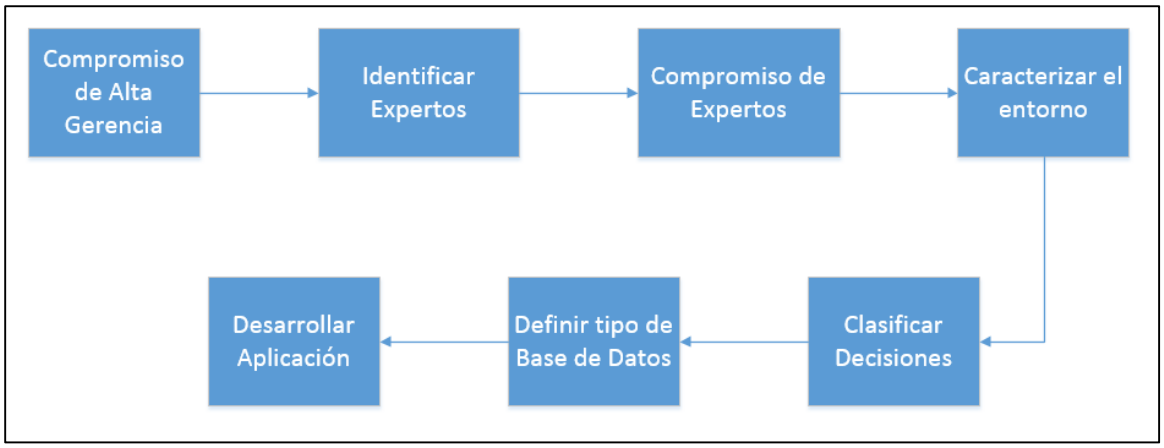

Figura 9: Metodología de Desarrollo de una aplicación de minería de datos para un entorno específico

A continuación, se va a explicar cada uno de los pasos de esta metodología.

3.1.1 Obtener Compromiso de Alta Gerencia

El esfuerzo para desarrollar una aplicación de minería de datos es significativo, tanto desde el punto de vista de tiempo como de recursos. Para lograr el éxito es importante que la alta gerencia apoye la idea.

Para iniciar este proceso es recomendable explicarle a la alta gerencia lo que es la minería de datos y cómo esta puede beneficiar su entorno/negocio.

Existen una gran cantidad de tutoriales en internet en donde se explica la minería de datos y los beneficios que se pueden obtener de ella. Por ejemplo, en el sitio web de zentut.com (Advantages and Disadvantages of Data Mining, s.f.) se mencionan las siguientes ventajas de utilizar minería de datos:

- *Mercadeo y ventas*: se pueden crear modelos predictivos que utilizan datos históricos de ventas para desarrollar campañas de mercadeo más eficientes. Se pueden analizar los patrones de compra de tal forma que se acomoden los productos en los estantes en una forma tal que productos que son comprados juntos (por ejemplo, impresoras y tinta) estén juntos y de esta forma sus ventas se vean incrementadas. También, basados en este análisis, es posible crear "paquetes" de productos.
- *Banca y finanzas*: es posible desarrollar modelos de riesgo para clasificar clientes que están solicitando créditos. También es posible detectar fraudes al analizar los patrones de compra.
- *Manufactura*: al analizar los datos operacionales se pueden detectar equipos dañados y definir los mejores límites de control.

Adicionalmente, en el Apéndice 3: Videos Tutoriales de Minería de Datos, se presentan varios videos que pueden ser utilizados para explicar los conceptos básicos de minería de datos y sus beneficios.

3.1.2 Identificar Expertos en el Entorno

La única forma caracterizar correctamente un entorno es con la ayuda de expertos en el mismo. Generalmente estos expertos son bien conocidos por los diferentes miembros de la organización, por lo que encontrarlos no es una tarea difícil.

Recomendamos los siguientes pasos para encontrar a estos expertos:

1. Obtener de la alta gerencia la lista de las actividades más importantes del entorno.

2. Hacer una encuesta en la cual participe la mayoría de la población, en la cual se va a preguntar, para cada actividad identificada por la alta gerencia, quién es la persona que más conoce sobre esa tarea.
3. Se van a analizar los datos para encontrar los nombres que más se repiten, de tal forma que se logre cubrir, con la menor cantidad de personas posible (se recomienda a lo sumo cinco), la mayor cantidad de tareas. Se desea contar con un grupo pequeño de expertos para facilitar las etapas siguientes.

La lista de personas que obtendríamos luego de ejecutar los tres pasos anteriores sería nuestra lista de expertos.

3.1.3 Obtener Compromiso de los Expertos

De forma similar a como se hizo con la alta gerencia, es importante explicarle a los expertos lo que es la minería de datos y cómo ellos pueden beneficiarse de su uso. Sin embargo, en este caso no se desea justificar el beneficio desde el punto de vista de negocio, sino más bien desde el punto de vista de las diferentes optimizaciones que se pueden lograr en las actividades en las cuales ellos son expertos. Por ejemplo, explicarles que al hacer minería de datos las decisiones que tomen van a estar fundamentadas en datos, y que de esta forma su capacidad de influenciar a la alta gerencia va a aumentar.

El objetivo final de esta etapa es convencer a los expertos que ellos mismos van a ser los más beneficiados con la aplicación de minería de datos.

3.1.4 Caracterizar el Entorno

Este paso es de los más importantes del proceso. Lo primero que se va a hacer es generar una lista de las actividades más importantes. Para esto, se recomienda tener una sesión de lluvia de ideas en donde participen todos los expertos. Durante esta sesión, los expertos van a generar una lista de las diferentes actividades que se realizan en el entorno que se está caracterizando, y luego se van a priorizar en orden de importancia. Se recomienda el siguiente método para generar esta lista priorizada de actividades:

1. Citar a todos los expertos a una reunión de 2 horas
2. Explicarle a los expertos el objetivo de la reunión: Elaborar una lista priorizada de las actividades más importantes que se realizan en ese entorno
3. Darle a cada experto un block de hojas tipo *Post-it*® y un lapicero
4. Pedirle a los expertos que escriban una actividad por hoja, y que las peguen en una pared de la sala de reuniones
5. Una vez que todos los expertos hayan documentado todas las tareas que recuerden, ir agrupando las diferentes hojas en grupos, de tal forma que cada grupo de hojas represente una actividad
6. Escribir todas las actividades en una pizarra
7. Ahora, pedirle a los expertos que voten por las tres actividades que ellos consideren como las más importantes
8. Luego de que todos los expertos hayan votado, contabilizar el total de votos que tiene cada actividad.
9. Seleccionar las quince[2] actividades con mayor cantidad de votos. Estas van a ser las actividades prioritarias.[3]
10. Como último paso, para cada una de estas actividades, identificar a los expertos que más conocen de esa actividad. Tratar de no contar con más de tres expertos en cada actividad.

Una vez que se tiene la lista de las 15 actividades más importantes, se procede a llenar la Tabla 3. Para hacer esto de forma efectiva, es importante que se cite a los expertos, que previamente fueron identificados, de dicha actividad. Puede programarse una reunión de entre 30 minutos a 1 hora para completar la información de dicha actividad.

Característica	**Descripción**
Actividad	Descripción de una actividad que se realiza en el entorno

[2] El número quince fue seleccionado arbitrariamente para tener una cantidad suficiente de actividades que sea manejable y permita generar una buena caracterización del entorno.
[3] Es importante que una vez que se tenga la lista con las quince actividades más importantes, ésta sea validada por la alta gerencia.

Característica	Descripción
Actores	Elementos que interactúan durante esa actividad. Estos elementos pueden ser personas, herramientas, etc.
Experto(s)	Nombre de los expertos en esta actividad
Decisiones	Decisiones claves que se toman durante esa actividad
Tomador de decisiones	Actor encargado de tomar las decisiones
Datos necesarios para tomar la decisión	Datos que necesita el actor para tomar la decisión

Tabla 3: Caracterización de un entorno

Como se ve en la Tabla 3, para cada actividad se van a identificar las decisiones más importantes relacionadas con dicha actividad. Esta lista de decisiones va a ser de suma utilidad para definir las características de la aplicación de minería de datos que queremos desarrollar.

El último paso de este punto es validar las tablas con la información de las quince actividades con todos los expertos. Para hacer esto, se recomienda programar una reunión de por lo menos medio día. En esta reunión se van a revisar las tablas de todas las actividades. Es recomendable enviar las tablas con la información de todas las actividades a los expertos por lo menos un día antes de la reunión, para que ellos puedan revisarlas *a priori*.

3.1.5 Clasificar las Decisiones

Primero tenemos que tratar de determinar, para cada una de las decisiones, el tipo de tarea de minado de datos (sección 2.2.3) que se tiene que realizar. Este proceso va a requerir participación tanto de los expertos del entorno como de un científico de datos.

Para facilitar este proceso se desarrolló la Tabla 4. Al ir completando la tabla para cada una de las decisiones previamente identificadas, es posible que para una decisión dada, sea necesario llevar a cabo varias tareas, con lo que se tendrían que marcar varias columnas de la tabla. También puede darse el caso que las tareas tradicionales del minado de datos no

sean suficientes para tomar una decisión específica, caso en el cual se tendría que marcar la columna "Otra".

Luego de completar los datos de todas las decisiones se procede a calcular los totales para cada columna. De esta forma vamos a saber cuáles son las tareas de minería de datos más importantes para el entorno en el que se está trabajando. Esta información nos va a ayudar a definir el tipo de base de datos que se necesita, así como las características que tienen que cumplir las herramientas de minado y de visualización de datos.

Decisión	Tareas del minado de datos							
	Descripción	Asociación	Clasificación	Predicción	Agrupación	Tiempo	OLAP	Otra
Decisión 1								
Decisión 2								
…								
TOTAL								

Tabla 4: Clasificación de las decisiones en las tareas del minado de datos

3.1.6 Definir el Tipo de Base de Datos

Una vez que la Tabla 4 se ha completado, puede utilizarse la Tabla 2 para determinar el tipo de base de datos óptimo para el sistema que se quiere desarrollar. En el caso de que, luego de completar este proceso, no se logre definir un tipo específico de base de datos, por simplicidad se recomienda iniciar con una base de datos relacional.

Es importante recalcar que la recomendación dada por la Tabla 2 no necesariamente tiene que tomarse como la única opción, sino que más bien tiene que tratarse como una sugerencia. Otros factores como tipos de bases de datos existentes, acoplamiento con otros elementos del sistema de minado de datos, etc.; pueden llevarnos a seleccionar un tipo de base de datos diferente al recomendado por la Tabla 2.

3.1.7 Desarrollar la Aplicación de Minería de Datos

La aplicación de minería de datos incluye tanto la base de datos como las herramientas de análisis y visualización de datos, según se mencionó anteriormente.

Con respecto a las herramientas de minería y visualización de datos, en el mercado existen múltiples aplicaciones comerciales disponibles. La decisión entre utilizar una aplicación comercial o desarrollarla desde cero va a depender de varios factores, entre los cuales se pueden mencionar:

- *Funcionalidad*: se tiene que comparar la funcionalidad requerida con la que proveen las herramientas comerciales. Si existe una diferencia significativa es muy probable que se tenga que desarrollar la aplicación. Algunas funcionalidades que se deben verificar incluyen: tipo de bases de datos que soportan, tareas de minería de datos que pueden realizar, funcionalidades de visualización de datos.
- *Tiempo*: si existe un fuerte presión de tiempo y es necesario implementar el sistema lo antes posible, las aplicaciones comerciales tienen una gran ventaja ya que el tiempo para ponerlas a funcionar es mucho menor que el tiempo requerido para desarrollar una nueva aplicación.
- *Costo*: generalmente desarrollar una aplicación desde cero conlleva un mayor costo. Sin embargo, también hay que considerar los costos de las licencias de las aplicaciones comerciales.

En el Apéndice 4: Herramientas de Minería de Datos, se presentan detalles de varias herramientas de minería de datos existentes en la actualidad. Se recomienda conocer bien las opciones que están disponibles en el mercado, tanto para bases de datos como para herramientas de minería de datos. De esta forma se puede decidir si una herramienta ya existente puede cumplir con los requerimientos del sistema que se desea implementar, o si por el contrario, es necesario desarrollar una aplicación nueva.

Capítulo 4: Desarrollo de la Aplicación de Minería de Datos para el Entorno VLSI

En este capítulo se va a presentar el proceso que se siguió para desarrollar la aplicación de minería de datos. Se va a utilizar la metodología desarrollada en el capítulo anterior. El entorno que se va a caracterizar es el de diseño estructurado de circuitos integrados a muy gran escala de los GPU (Graphical Processing Unit, unidades de procesamiento de gráficos por sus siglas en inglés) de Intel Costa Rica.

A continuación se va a describir el proceso seguido.

4.1 Paso 1: Obtener Compromiso de Alta Gerencia

En el caso del departamento de diseño de GPUs, tanto la gerencia local como la gerencia en el extranjero tienen un buen conocimiento de lo que es la minería de datos, razón por la cual el compromiso se obtuvo sin mayor complicación.

4.2 Paso 2: Identificar Expertos en el Entorno

Al ser el autor de esta investigación miembro del departamento de diseño estructurado de circuitos VLSI de GPUs en Intel Costa Rica, la identificación de los expertos fue una tarea que él realizó basado en su conocimiento interno del departamento. Es importante aclarar que en la mayoría de los casos los científicos de datos no van a conocer a fondo a la población del entorno, razón por la cuál va a ser necesario seguir las recomendaciones dadas en la sección 3.1.2.

Sin embargo, para el caso de esta investigación, este proceso no fue necesario gracias al conocimiento que se tiene del departamento.

4.3 Paso 3: Obtener Compromiso de los Expertos

En el caso de los expertos, y de forma similar a cómo ocurrió con la alta gerencia, ellos tenían un buen conocimiento de lo que es la minería de datos. Por esta razón se logró obtener su compromiso para apoyar esta investigación. Adicionalmente, por la experiencia

que ha adquirido el autor de esta investigación en el entorno de diseño de circuitos VLSI, él mismo es considerado como parte del grupo de expertos.

4.4 Paso 4: Caracterizar el Entorno

En este paso se tiene que completar la Tabla 3, identificando las actividades más importantes para el entorno. El resultado de este ejercicio se muestra en la Tabla 5.

Es importante aclarar que por la naturaleza de confidencialidad del proceso de diseño estructurado de Intel, se van a utilizar acrónimos cuyo significado no se va a explicar. El objetivo de hacer esto es asegurarse que las características específicas del proceso de Intel no sean divulgadas. Sin embargo, aún con esto, se logra ejemplificar el proceso que se siguió para completar esta etapa de caracterización del entorno.

Característica	**Detalles**
Actividad 1	Definir área del diseño
Actores	FPO, UO
Experto(s)	FPO
Decisiones	Área asignada a cada nivel jerárquico del diseño
Tomador de decisiones	PM
Datos necesarios para tomar la decisión	Estimados de GC por unidad GC y área de diseños anteriores
Actividad 2	Definir localización de los elementos al nivel jerárquico más alto
Actores	FPO
Experto(s)	FPO
Decisiones	Localización de los elementos a nivel de jerarquía 1 y nivel de jerarquía 2
Tomador de decisiones	FPO
Datos necesarios para tomar la decisión	Tamaño estimado de elementos de niveles de jerarquía 1 y 2 Bits de comunicación entre cada elemento de jerarquía

Característica	Detalles
Actividad 3	Desarrollo del entorno de automatización
Actores	RLS, SynTool, APRTool
Experto(s)	RLS
Decisiones	Orden de las operaciones del proceso de diseño estructural y parámetros de configuración
Tomador de decisiones	RLS
Datos necesarios para tomar la decisión	Características del diseño Funcionalidad de las herramientas de diseño asistido por computadora Detalles de entornos de proyectos previos
Actividad 4	Síntesis
Actores	PEO, SEO, RLS, SynTool
Experto(s)	PEO, SEO
Decisiones	Aceptar el esquemático generado por la síntesis
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Detalles de los errores en el proceso de síntesis Resultados de GC Resultados de Timing Resultados de DRVs
Actividad 5	Path Groups
Actores	PEO, SynTool, APRTool
Experto(s)	PEO
Decisiones	¿Dónde y cómo se implementan los Path Groups?
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Detalles de los timing paths que no cumplen requerimientos
Actividad 6	Placement blockages
Actores	PEO, SynTool, APRTool
Experto(s)	PEO
Decisiones	¿Dónde y cómo se implementan los placement blockages?

Característica	Detalles
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Datos de DRVs
Actividad 7	Routing blockages
Actores	PEO, SynTool, APRTool
Experto(s)	PEO
Decisiones	¿Dónde y cómo se implementan los routing blockages?
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Datos de DRVs
Actividad 8	Repeaters
Actores	PEO, APRTool, PVTool
Experto(s)	PEO, STO
Decisiones	¿Dónde y cómo se implementan los repeaters?
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Datos de timing Detalles de los timing paths
Actividad 9	Colocación de puertos
Actores	PEO, SEO, SFPO, FPO
Experto(s)	FPO
Decisiones	Localización final de los puertos
Tomador de decisiones	FPO
Datos necesarios para tomar la decisión	Localización inicial de puertos Detalles de los timing paths Datos de timing Bits de comunicación entre elementos de nivel de jerarquía 2 y 3
Actividad 10	Colocación de DOPs
Actores	PEO, SEO, SpineO
Experto(s)	SEO
Decisiones	Localización final de DOPs
Tomador de decisiones	SpineO

Característica	Detalles
Datos necesarios para tomar la decisión	Localización inicial de DOPs Área de cobertura de DOP Detalles de red de relojes Datos de timing
Actividad 11	Definir celdas de no-uso
Actores	RLS, LibT
Experto(s)	LibT
Decisiones	Lista de celdas de no-uso
Tomador de decisiones	LibT
Datos necesarios para tomar la decisión	Detalles de las celdas Detalles de la tecnología Uso de celdas en el diseño
Actividad 12	Retroalimentación a UO
Actores	PEO, SEO, RLS, UO
Experto(s)	UO
Decisiones	Cambios en RTL
Tomador de decisiones	UO
Datos necesarios para tomar la decisión	Datos de timing Datos de DRVs Datos de tiempo de ejecución de SynT y APRT
Actividad 13	Colocación de EBBs
Actores	PEO, SFPO, FPO
Experto(s)	PEO
Decisiones	Localización final de EBBs
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Localización inicial de EBBs Datos de Timing Datos de DRVs Detalles de jerarquía nivel 2 y nivel 3
Actividad 14	Optimización de tiempos de ejecución de SynT y APRT

Característica	Detalles
Actores	RLS, SynT, APRT
Experto(s)	RLS
Decisiones	Cambios en los parámetros de SynT y APRT
Tomador de decisiones	RLS
Datos necesarios para tomar la decisión	Tiempos de ejecución de SynT y APRT Datos de Timing Datos de DRVs
Actividad 15	Cambios de celdas
Actores	PEO, APRT, STO
Experto(s)	PEO, STO
Decisiones	Cambios de celdas
Tomador de decisiones	PEO
Datos necesarios para tomar la decisión	Características de las celdas Datos de timing Datos de DRVs

Tabla 5: Caracterización del entorno de diseño estructurado de un GPU

Con las quince actividades y decisiones documentadas, se puede continuar con el siguiente paso de la metodología.

4.5 Paso 5: Clasificar las Decisiones

En este paso se tiene que completar la Tabla 4 con las decisiones identificadas en el paso anterior (Tabla 5). Para que este paso se complete correctamente, es importante contar con la participación de expertos en el entorno y de científicos de datos, de tal forma que ambos grupos de expertos van a revisar las diferentes decisiones del entorno y proceder a categorizarlas en una o varias tareas de minería de datos. La Tabla 6 muestra el resultado de clasificar las quince decisiones.

Decisión	Tareas del minado de datos							
	Descripción	Asociación	Clasificación	Predicción	Agrupación	Tiempo	OLAP	Otra
Actividad 1: Área asignada a cada nivel jerárquico del diseño				X				
Actividad 2: Localización de los elementos a nivel de jerarquía 1 y nivel de jerarquía 2	X	X						
Actividad 3: Orden de las operaciones del proceso de diseño estructural y parámetros de configuración	X							X
Actividad 4: Aceptar el esquemático generado por la síntesis	X						X	
Actividad 5: ¿Dónde y cómo se implementan los Path Groups?	X	X					X	
Actividad 6: ¿Dónde y cómo se implementan los placement blockages?	X	X					X	
Actividad 7: ¿Dónde y cómo se implementan los routing blockages?	X	X					X	
Actividad 8: ¿Dónde y cómo se implementan los repeaters?	X	X					X	
Actividad 9: Localización final de los puertos								X
Actividad 10: Localización final de DOPs								X
Actividad 11: Lista de celdas de no-uso	X	X	X					
Actividad 12: Cambios en RTL	X		X	X				X
Actividad 13: Localización final de EBBs		X						X
Actividad 14: Cambios en los parámetros de SynT y APRT	X	X		X			X	X
Actividad 15: Cambios de celdas	X	X	X				X	X
TOTAL	**11**	**9**	**3**	**3**	**0**	**0**	**7**	**7**

Tabla 6: Clasificación de decisiones del diseño de GPUs en las tareas del minado de datos

Se puede concluir que para el entorno de diseño estructurado de GPUs, las tareas de minado de datos predominantes son las descriptivas, las de asociación, los análisis OLAP y otros

tipos de tareas. Conocer esta información es de suma importancia para escoger el tipo de base de datos y las herramientas de minado y visualización, de tal forma que se alíneen con las necesidades del entorno, tal y como se va a ver en las siguientes secciones.

4.6 Paso 6: Definir el Tipo de Base de Datos

Utilizando los resultados de la Tabla 6 y comparándolos con las recomendaciones dadas en la Tabla 2, se puede concluir que la hay dos posibles tipos de base de datos que se alinean de manera ideal para el entorno de diseño estructurado de GPUs: la relacional y la NoSQL.

Luego de completar este paso se procede a desarrollar la aplicación de minería de datos, procedimiento que se va a presentar en detalle en la siguiente sección.

4.7 Desarrollo de la Aplicación de Minería de Datos para Entorno VLSI

La metodología de desarrollo que se propone es innovadora y puede ser utilizada para desarrollar otras aplicaciones de software, razón por la cual va a ser explicada. Esta metodología es una combinación de dos métodos usados en el desarrollo de software: el método cascada (también conocido como waterfall) y el método Kanban (PATTON, s.f.) (DENNING, s.f.) (SAHOTA, 2012) (BECK & ANDRES, 2004).

El método cascada se utiliza en el nivel más alto del desarrollo, el cual se divide en cinco etapas, tal y como se puede observar en la Figura 10.

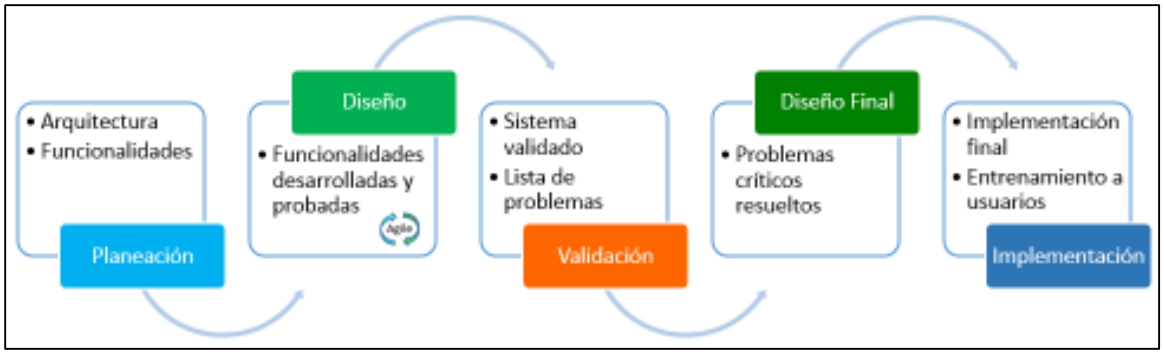

Figura 10: Metodología de Desarrollo

La Tabla 7 presenta una descripción de cada una de estas etapas, al igual que los objetivos que se deben lograr al final de cada una de ellas.

Etapa del Desarrollo	Descripción	Objetivos
Planeación	Se define la arquitectura del sistema Se definen las características y funcionalidades que va a tener el sistema	Arquitectura del sistema Lista priorizada de funcionalidades del sistema
Diseño	Se desarrolla cada una de las funcionalidades del sistema. Durante esta etapa se utilizan prácticas Agile para desarrollar las funcionalidades, de tal forma que cada funcionalidad del sistema se valide completamente	Todas las funcionalidades del sistema se han desarrollado y probado, tanto a nivel individual como a nivel integrado con otras funcionalidades
Validación	Se procede a hacer una validación del sistema completo con un grupo reducido de usuarios.	Validar la funcionalidad completa del sistema Generar una lista de los problemas encontrados, con una descripción clara del impacto de cada problema Análisis y disposición de cada uno de los problemas encontrados. Posibles disposiciones que se podrían tener incluyen: resolver inmediatamente, resolver en el futuro, no resolver, problema inválido
Diseño Final	Se analizan los datos de la validación y se definen y desarrollan los cambios necesarios para que el sistema cumpla con las expectativas de los clientes (funcionalidad, rendimiento, etc.)	Todos los problemas categorizados como “resolver inmediatamente” están implementados en la aplicación.

Etapa del Desarrollo	Descripción	Objetivos
Implementación	Se hace la implementación final a toda la población de usuarios de la aplicación.	Implementación de la aplicación Entrenamientos a los usuarios Definición e implementación del modelo de soporte

Tabla 7: Etapas del Desarrollo

Tal y como se menciona en la Tabla 7, durante la etapa de diseño se siguen prácticas Agile para el desarrollo de cada funcionalidad. Específicamente, se siguió la metodología Kanban durante esta etapa, de tal forma que cada funcionalidad que se tenía que desarrollar se seleccionaba de acuerdo a la importancia de la misma.

Se van a cubrir las primeras tres etapas del proceso de desarrollo descrito en la Tabla 7. Los problemas encontrados durante la etapa de validación van a ser explicados en detalle en el Capítulo 5:, y en el Capítulo 6: se van a dar sugerencias de cómo resolver estos problemas.

A continuación, se va a describir en detalle el proceso seguido para desarrollar la aplicación de minería de datos.

4.7.1 Planeación

En esta etapa inicial se define la arquitectura del sistema y las funcionalidades principales. Es importante que al definir la lista de funcionalidades se asignen prioridades para ayudar al equipo de desarrollo a priorizar su trabajo de acuerdo a las necesidades de los usuarios.

Si partimos de una implementación típica de un sistema de minería de datos, la arquitectura a alto nivel se muestra en la Figura 11.

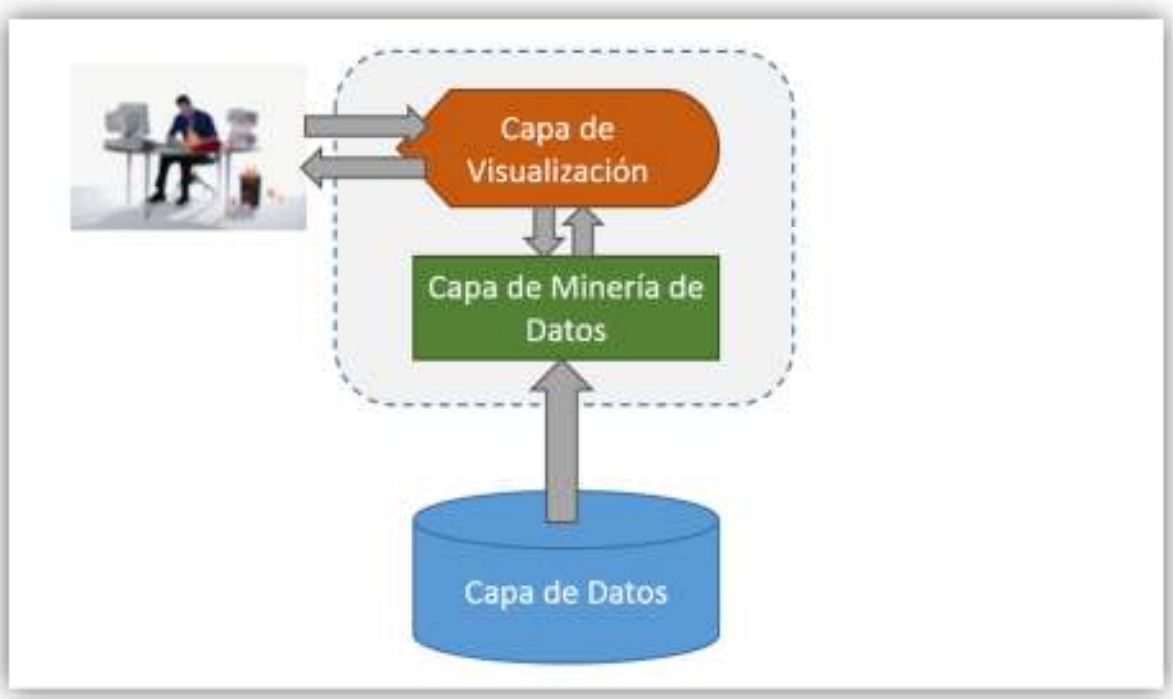

Figura 11: Arquitectura de una aplicación típica de minería de datos

Como se puede ver, la arquitectura del sistema consta de tres componentes principales: *capa de datos*, *capa de minería de datos*, *capa de visualización*.

La capa de datos incluye la base de datos que se va a utilizar al igual que todos los elementos requeridos para cargar datos en la misma.

En el entorno de diseño VLSI, la gran mayoría de los datos del sistema son generados por las herramientas CAD, las cuales crean archivos planos. De esta forma, va a ser necesario desarrollar elementos que analicen estos archivos planos, extraigan los datos y luego los carguen en la base de datos.

La capa de base de datos también incluye los procesos que se encargan de limpiar y transformar los datos ((HAN & KAMBER, 2006, págs. 47-97), al igual que los procesos que se encargan de verificar la calidad de los mismos.

La capa de minería de datos incluye todas las herramientas responsables de extraer datos de la capa de datos, analizarlos y hallar patrones ventajosos (sección 2.2.1). Idealmente, la capa de minería de datos y la capa de visualización deberían estar integradas en una aplicación única, sin embargo, existe la posibilidad de que sean aplicaciones independientes.

Por último, la capa de visualización incluye todas las herramientas que van a permitir a los usuarios interactuar con el sistema. Esta es una capa de "entrada-salida", ya que el usuario va a entrar información en el sistema (ej. configuraciones, solicitudes de análisis, etc.) y también va a sacar información del sistema (ej. gráficos, tablas, resúmenes, etc.).

Es muy importante que el usuario tenga acceso a múltiples técnicas de visualización de datos, para de esta forma interactuar de una forma más eficiente con los datos y tomar las mejores decisiones (HERNANDEZ-CASTRO, 2012).

Luego de completar la caracterización del entorno y de analizar las necesidades del mismo, se completó la lista de funcionalidades que se resume en la Tabla 8.

Capa	Funcionalidad	Descripción	Prioridad
Datos	Retención de datos	Se tienen que mantener datos históricos de por lo menos los últimos cinco proyectos anteriores.	Alta
	Almacenamiento de datos	Los datos requeridos para hacer los análisis descritos en la Tabla 5 deben estar disponibles en la base de datos	Alta
Minado	Tareas de minado de datos	Las siguientes tareas de minado de datos tienen que estar disponibles: descripción, asociación, OLAP, Clasificación, Predicción.	Alta
	Tareas de minado de datos	Las siguientes tareas de minado de datos tienen que estar disponibles: agrupación, análisis de series de tiempo	Baja
Visualización	Gráficos básicos	La herramienta tiene que ser capaz de generar los siguientes tipos de gráficos: barras, línea, X-Y, dispersión.	Alta

Capa	Funcionalidad	Descripción	Prioridad
	Gráficos avanzados	La herramienta tiene que ser capaz de generar los siguientes tipos de gráficos: conos, árbol, burbujas, cuerdas, redes.	Baja
	Tablas pivote	El usuario tiene que tener la capacidad de generar e interactuar con tablas pivote	Alta

Tabla 8: Funcionalidades Principales del Sistema de Minería de Datos

4.7.2 Diseño

En esta sección se va a describir el proceso que se siguió para diseñar y desarrollar los diferentes elementos que componen el sistema de minería de datos.

Actualmente, en el mercado hay una gran variedad de opciones disponibles para bases de datos, herramientas de minería y de visualización de datos. Así, durante esta etapa de diseño, se van a evaluar varias opciones comerciales antes de iniciar a desarrollar una herramienta desde cero. Esto con la esperanza de encontrar una solución comercial que cumpla con los requisitos del sistema que se quiere desarrollar y de esta forma acelerar el proceso de desarrollo.

A. Capa de Datos

De la sección 4.6 se concluyó que la base de datos ideal puede ser del tipo relacional o del tipo NoSQL.

Al investigar más a fondo sobre el entorno de diseño VLSI de Intel, se encontró que actualmente existe una base de datos en la cual se cargan diariamente cientos de miles de registros que contienen datos del proceso de diseño. Esta base de datos es conocida como *MDP* (Metrics for Design Process, métricas del proceso de diseño por sus siglas en inglés). De esta forma es fácil intuir que es necesario investigar sobre esta base de datos para determinar si puede ser utilizada como la base de datos del sistema de minado de datos que se desea desarrollar.

Se encontró que MDP es una base de datos relacional, sin esquema, que ha sido implementada en el entorno de diseño de Intel. MDP tiene dos tablas principales: la Tabla_1 contiene datos estándar que son comunes para cada registro, como por ejemplo nombre del proyecto, versión del proyecto, etc. La Tabla_2 se utiliza para almacenar los datos específicos del proceso de diseño, en un formato de duplas llave-valor, y contiene cientos de miles de llaves diferentes. Estas dos tablas están relacionadas entre sí por medio de una serie de columnas de identificación común.

Esta estructura permite almacenar una gran variedad de datos sin necesidad de modificar la configuración de la base de datos (por ejemplo, agregar nuevas columnas a tablas ya existentes).

La Figura 12 muestra de forma simplificada la manera en que las dos tablas principales de MDP trabajan juntas para permitir gran flexibilidad en el tipo y variedad de datos que se almacenan. Es fácil observar que agregar un tipo de dato nuevo es sumamente sencillo, y se logra con sólo cargar un registro llave-valor en la Tabla_2, donde el valor de la llave contiene el nombre del nuevo dato que se desea cargar.

Es importante mencionar que la estructura real de MDP es mucho más compleja que la que se presenta en la Figura 12, ya que la Tabla_2 tiene varios niveles de jerarquía en su estructura de llave-valor, y la unión entre la Tabla_1 y la Tabla_2 se logra al usar varias columnas, no únicamente una como se muestra en la figura. Sin embargo, por razones de confidencialidad, la estructura completa de esta base de datos no puede describirse en detalle. Aún con esta limitante, para efectos de este trabajo la estructura presentada es suficiente para entender la forma en que MDP está diseñada.

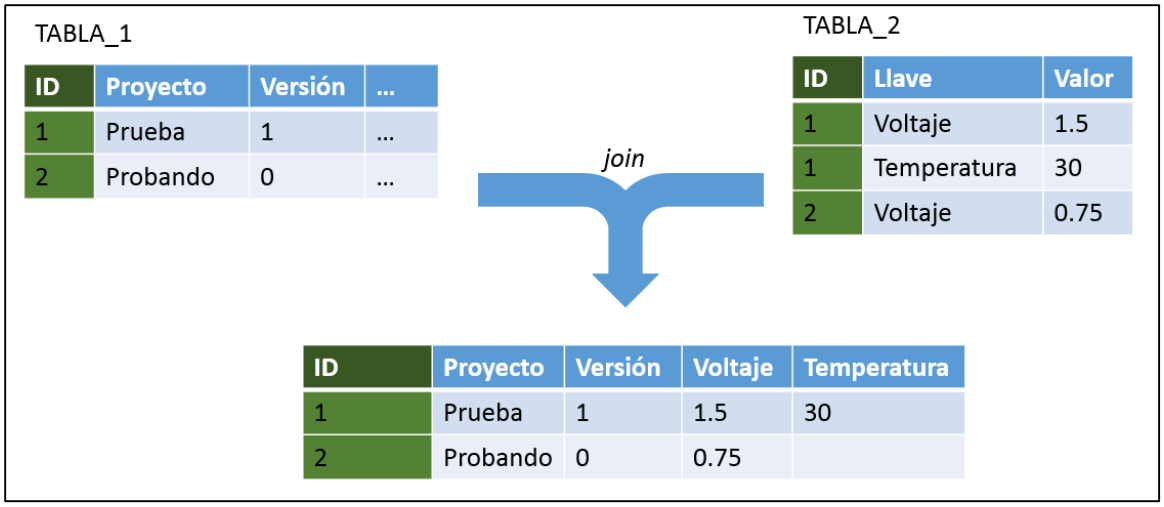

Figura 12: Estructura (simplificada) de MDP

MDP tiene una gran cantidad de datos, que incluye múltiples proyectos que se han desarrollado en los últimos años.

Si revisamos la Tabla 8 y comparamos las funcionalidades requeridas con las características de MDP, se puede concluir que MDP cumple con las características deseadas. De esta forma, se tomó la decisión de usar MDP como la base de datos del sistema de minería de datos por desarrollar.

B. Capa de Minería de Datos

El primer paso que se va a realizar es mostrar cómo evaluar varias opciones de herramientas de minería de datos disponibles. Para lograr esto se va a seguir un proceso similar al usado en (GOEBEL & GRUENWALD, 1999).

El proceso de selección va a considerar las características generales de las diferentes aplicaciones, la interacción con bases de datos y las funciones de minería de datos y de visualización de datos disponibles.

Los criterios que se van a analizar desde el punto de vista de *características generales* incluyen:

- Capacidad de la herramienta para manejo de datos. Es una medida de la cantidad de registros que la herramienta puede manejar eficientemente.
- Costos de licencia para instalar la aplicación.
- Sistema Operativo en el que corre la aplicación.
- Facilidad de uso de la herramienta.

Desde el punto de vista de *interacción con bases de datos*, se va a evaluar:

- Capacidad de la aplicación para conectarse a una base de datos relacional (MDP es una base de datos relacional)
- Tipos de datos que puede manejar (ej. Numéricos, texto, boléanos, etc.)
- Capacidad de accesar la base de datos directamente por medio de comandos SQL.

Con respecto a las funciones de minería de datos y de visualización, se va a medir la capacidad de las herramientas de cumplir con lo que se requiere de acuerdo a la Tabla 8.

Para este análisis se identificaron las siguientes aplicaciones: Weka, Tanagra, RapidMiner CE y Cloudera CDH. En el Apéndice 4: Herramientas de Minería de Datos se da una breve descripción de cada una de estas aplicaciones, y en la Tabla 9 se presentan los resultados comparativos entre todas las opciones[4]. Algunos de los datos de la Tabla 9 son basados en estimaciones puesto que las páginas de internet de las herramientas no incluían información completa. Para determinar si un dato de esta tabla es completamente confiable o es estimado, se siguió la siguiente convención:

- *Mayúscula*: Dato confiable. Ejemplo: S (significa que en la página de la herramienta se menciona explícitamente que esa funcionalidad es soportada por la herramienta)
- *Minúscula*: Dato estimado. Ejemplo: n (en la página de la herramienta no se hace mención de esa funcionalidad específica).

[4] Basados en la información disponible en las páginas de internet de cada una de las herramientas.

Herramienta	Características Generales				Interacción con BD			Tareas de Minado							Visualización										Comentarios
	Capacidad de Datos	Costo Licencia	Corre en Windows	Facilidad de Uso	BD Relacional	Tipos de Datos	SQL	Descripción	Asociación	Clasificación	Predicción	OLAP	Agrupación	Series de Tiempo	Barras	Línea	X-Y	Dispersión (scatter)	Tablas Pivote	Conos	Árbol	Burbujas	Cuerdas	Redes	
Weka	¿?	0	S		S	7	S	S	S	S	S	N	S	N	S	S	S	S	N	N	S	n	n	n	
TANAGRA	¿?	0	S		N	7	N	S	S	S	S	N	S	n	N	S	S	S	N	N	S	n	n	n	Entrada de datos por medio de archivos de texto.
RapidMiner CE	¿?	0	S		s	7	s	S	S	S	S	s	S	S	S	S	S	S	s	s	S	s	s	s	En la página de RapidMiner hay una gran cantidad de material de entrenamiento. RapidMiner es usado por algunos grupos en Intel.
Cloudera CDH	¿?	0	N		s	7	s	s	s	s	s	n	s	s	n	n	n	n	n	n	n	n	n	n	No existe mucha información disponible sobre las funciones de minado y visualización.

Tabla 9: Opciones de herramientas de minería de datos

En la Tabla 9, la columna "Tipos de Datos" va a seguir la codificación que se muestra en la Tabla 10.

Código	Datos Continuos	Datos Categóricos	Datos Simbólicos (texto)
0	No	No	No
1	No	No	Si
2	No	Si	No
3	No	Si	Si
4	Si	No	No
5	Si	No	Si
6	Si	Si	No
7	Si	Si	Si

Tabla 10: Codificación de los tipos de datos

Al analizar los datos de la Tabla 9 es claro que la herramienta que presenta las mejores características es RapidMiner CE. Esta herramienta no solo cumple los requisitos para la capa de minería, sino que también cubre los requisitos de la capa de visualización.

De esta forma, la herramienta que se va a utilizar tanto para la Capa de Minería como para la Capa de Visualización va a ser RapidMiner CE. En el Apéndice 5: Uso Básico de RapidMiner se presenta un resumen breve de las características de RapidMiner.

4.7.3 Validación

El diseño final de la aplicación de minería de datos se muestra en la Figura 13.

Figura 13: Diseño final de la aplicación de minería de datos

Para demostrar que esta aplicación tiene resultados positivos en el proceso de diseño de circuitos VLSI, se van a realizar los siguientes análisis:

1. Análisis jerárquico de violaciones de tiempo máximo
2. Predicción de la complejidad de las particiones
3. Análisis de predictores en la tasa de crecimiento de la cantidad de celdas
4. Análisis de predictores para variables del *QoR* (Quality of Results, resultados de calidad por sus siglas en inglés).

Los resultados obtenidos se presentan en el siguiente capítulo.

Capítulo 5: Análisis de Resultados

En este capítulo se van a presentar los resultados obtenidos al implementar técnicas de minado de datos en el entorno de diseño de circuitos VLSI para procesadores gráficos de Intel.

Es importante recordar que para respetar los acuerdos de confidencialidad con Intel, los nombres de los elementos del diseño van a ser generalizados (por ejemplo, en lugar de usar el nombre real de una unidad jerárquica del diseño, se va a usar un nombre genérico como Unidad_1). También, los datos van a ser modificados para prevenir una fuga de información. De esta forma, en lugar de presentar los datos reales de los diseños, éstos van a ser modificados mediante una fórmula matemática, de tal forma que pueden ser multiplicados, sumados, restados, divididos, etc.

Al presentar los resultados se va a utilizar un formato común basado en CRISP-DM (sección 2.2.5).

Para el entorno de diseño de circuitos VLSI se van a realizar tres tipos de análisis principales:

1. *Situación actual*: Se realizan resúmenes de los datos para lograr un mejor entendimiento de los mismos y de la situación actual.

2. *Modelo Predictivo*: Se genera un modelo que puede predecir el valor de una variable de interés a partir de un grupo de predictores.

3. *Análisis de Predictores*: Este tipo de análisis va a tener una variable de interés y un grupo de posibles predictores. El objetivo del análisis es determinar cuáles de estos posibles predictores son en realidad significativos y tienen un impacto importante en el valor final de la variable de interés.

En los casos en los que se tenga un conjunto de variables de interés, se recomienda hacer un análisis de clústeres de tal forma que se pueda crear una variable única de interés, de tipo nominal. Esta variable se utiliza como la variable de interés. De esta forma, en lugar de tener múltiples variables de interés, se puede trabajar con únicamente una variable de interés.

La Tabla 11 muestra los análisis de minería de datos recomendados dependiendo del tipo de variables predictivas y de interés.

Variable de Interés	Variables Predictivas	Análisis Recomendado
Nominal	Continua	Discriminante
		Regresión logística
Continua	Continua	Matriz de correlación
		Regresión lineal
Nominal	Cualquiera	Árbol de decisión
Cualquiera	Cualquiera	Red Neuronal

Tabla 11: Tipos de análisis recomendados

A continuación se van a presentar los resultados de cada uno de los análisis realizados.

5.1 Análisis Jerárquico de Violaciones de Tiempo Máximo

5.1.1 Entender el Negocio

Uno de los objetivos de los diseñadores estructurales es lograr que sus diseños cumplan con los requerimientos de tiempo, los cuales definen la frecuencia de operación del circuito. Entre más rápido viajen las señales eléctricas, más alta va a ser la frecuencia de operación y se obtiene un producto más competitivo.

Sin embargo, este no es un proceso sencillo puesto que en las etapas iniciales, los datos de tiempo se basan únicamente en simulaciones que estiman el tiempo que tardan las diferentes señales lógicas en viajar de una compuerta a la siguiente, y comparaciones de estos tiempos con los requerimientos. Y, por la gran cantidad de compuertas lógicas que

forman parte de un circuito VLSI, la cantidad de datos generados por estas simulaciones es muy extensa. Estos datos incluyen detalles de las celdas (ej. tipo, jerarquía) y de los cables que componen el circuito que está violando. También, incluyen detalles del tiempo de propagación de las señales eléctricas en ese circuito. Los datos se guardan en archivos de texto que luego son cargados a la base de datos MDP.

Para facilitar el proceso de convergencia de tiempos se define una estructura jerárquica para poder dividir el diseño en partes, y de esta forma asignar diferentes partes a diferentes diseñadores. En la Tabla 12 se muestran datos aproximados de la cantidad de compuertas que cada nivel jerárquico tiene.

Nivel Jerárquico	Cantidad aproximada de compuertas
Bloque Lógico	100-1000 millones
Sección	10-100 millones
Partición	1-2 millones
Unidad	100-500 miles
Bloque Funcional	1-50 miles

Tabla 12: Cantidad aproximada de compuertas lógicas por nivel jerárquico

Un diseñador estructural generalmente es dueño de varias particiones, y es responsable de asegurarse que los requerimientos de temporización de sus particiones se cumplan. Existen requerimientos tanto de tiempos máximos como de tiempos mínimos, y si cualesquiera de estos requerimientos no se cumple se da una violación. Las violaciones de tiempo máximo son conocidas como *violaciones de setup* o de *max*, y las de tiempo mínimo como *violaciones de hold* o de *min*.

Adicionalmente, dependiendo de los puntos de inicio y fin del circuito lógico, se pueden generar cuatro tipos diferentes de violaciones:

- *Intra*: Si las unidades a las que pertenecen las celdas de inicio y fin del circuito lógico son la misma
- *Inter*: Si tenemos dos unidades diferentes

- *IO*: Si el punto de inicio o el punto de fin es externo a la partición.
- *FF*: Si tanto el punto de inicio como el punto de fin son externos a la partición.

La Figura 14 muestra una representación gráfica de los diferentes tipos de violaciones de tiempo, dependiendo de sus puntos de inicio y fin.

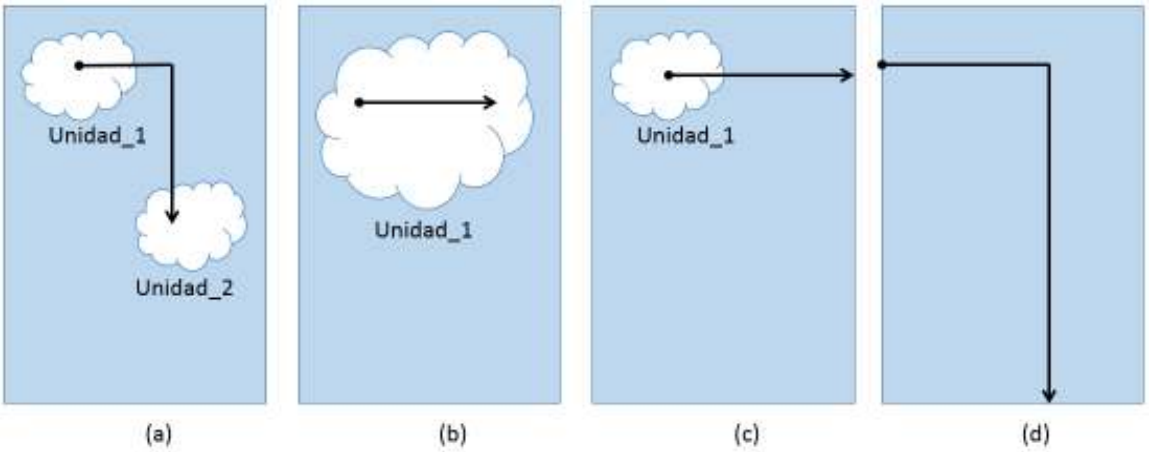

Figura 14: Tipos de violaciones de tiempo (a) inter, (b) intra, (c) IO, (d) FF.

Es responsabilidad de los diseñadores estructurales revisar cada una de estas violaciones y encontrar formas para eliminarlas. Hay diferentes técnicas que se pueden seguir para resolver estas violaciones dependiendo de la clase de violación (max o min).

En el caso de las violaciones de max, una técnica de resolución se basa en definir la localización de las diferentes unidades dentro de la partición de tal forma que se reduzca la distancia entre compuertas y de esta forma se reduzca el tiempo de propagación, logrando cumplir los requerimientos de tiempo. Por ejemplo, en la Figura 15 se puede observar cómo cambiando la localización de las unidades se logra eliminar la violación de tiempos máximos.

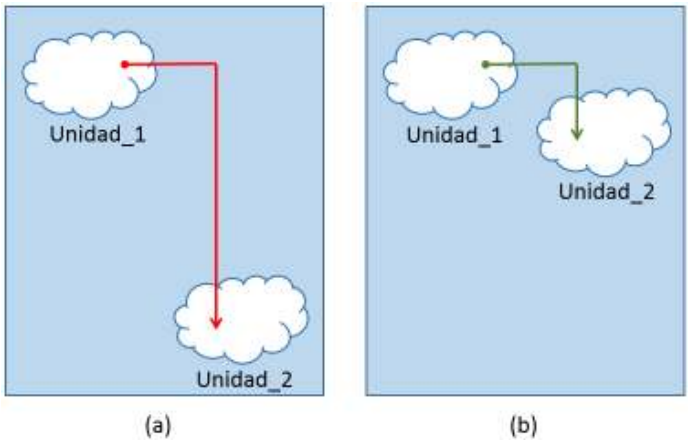

Figura 15: Circuito que (a) no cumple y (b) si cumple requerimientos de tiempo máximo

El análisis que vamos a hacer nos va a ayudar a dar recomendaciones sobre la localización que deberían tener las unidades dentro de la partición de tal forma que se reduzcan las violaciones de tiempos máximos.

5.1.2 Entender los Datos

Los datos necesarios para poder hacer el análisis propuesto incluyen:

- *Jerarquía a nivel de unidad de la celda inicial del circuito*. Es el nombre de la unidad en la cual se encuentra la celda de inicio del circuito que está violando la condición de tiempo máximo. Si el punto de inicio es externo a la partición, entonces el punto de inicio se considera que es un puerto.
- *Jerarquía a nivel de unidad de la celda final del circuito*. Es el nombre de la unidad en la cual se encuentra la última celda del circuito que está violando la condición de tiempo máximo. Si el punto final es externo a la partición, se considera que es un puerto.
- *Tipo de violación*. Ya sea inter, intra, IO o FF.
- *Tiempo*. Es la cantidad de tiempo por el cual se está violando el requerimiento de ese circuito

Estos datos actualmente no se están cargando con el nivel de detalle necesario en MDP, razón por la cual va a ser necesario extraer la información directamente del archivo de texto en el cual se guarda el reporte de la simulación de temporización.

5.1.3 Preparar los Datos

Se desarrolló una aplicación que extrae los datos requeridos del reporte de violaciones de tiempo máximo y genera un archivo .csv con la siguiente información: unidad de inicio, unidad de fin, tipo de violación, tiempo.

Adicionalmente, se crea un dato adicional que concatena los nombres de las unidades de inicio y final en forma ordenada. Por ejemplo, este dato va a ser "Unidad_1 | Unidad_2" sin importar si la unidad de inicio es la Unidad_1 y la de fin la Unidad_2 o si la unidad de inicio es la Unidad_2 y la de fin la Unidad_1.

Una vez que se tiene este archivo, se procede a cargarlo en RapidMiner.

5.1.4 Modelado

En este caso vamos a hacer un análisis tipo 3: situación actual. El objetivo es entender los detalles de las violaciones de tiempo máximo, separándolo por tipo de violación. De esta forma se puedan tomar decisiones de como acomodar las celdas de las unidades y así reducir la cantidad de violaciones.

Se va a proceder a generar una tabla resumen donde se vea la cantidad de violaciones y la suma total del tiempo de violación, conocido como *TNS* (Total Negative Slack, margen negativo total por sus siglas en inglés), para todas las posibles combinaciones de unidad de inicio y unidad de fin por cada tipo de violación.

El proceso de RapidMiner desarrollado para generar estas tablas se muestra en la Figura 16.

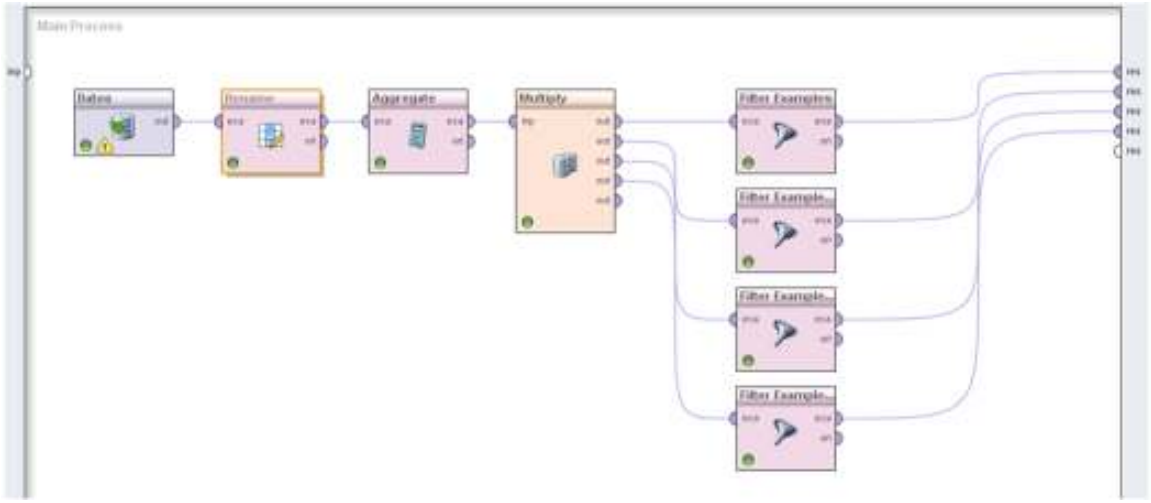

Figura 16: Proceso de RapidMiner para el análisis jerárquico de violaciones de tiempo máximo

Se va a dar una breve descripción de cada etapa del flujo:

- *Datos*: Este operador se usa para importar los datos del archivo .csv a RapidMiner.
- *Rename*: Se cambia el nombre de la columna donde se tiene la información de la violación de tiempo de "Tiempo de violación" a "Violacion". Este cambio no tiene ningún fin práctico, es únicamente para facilitar la lectura de las tablas resultantes.
- *Aggregate*: Este es el operador principal en este análisis. Se utiliza para generar resúmenes por tipo de violación y combinación de unidades.
- *Multiply*: Se utiliza para generar multiples copias de los datos. Esto con el propósito de tener cuatro tablas separadas al final del proceso,en lugar de una única tabla con todos los datos.
- *Filter Examples*: Se utiliza para filtrar los datos por tipo de violación, y de esta forma poder generar cuatro tablas separadas, una para cada tipo de violación.

Las tablas resumen resultantes se presentan a continuación, ordenadas de forma descendente con base en la magnitud del TNS.

ExampleSet (66 examples, 0 special attributes, 4 regular attributes)

Row No.	Merged sorted	Type	sum(Violacion)	count(Violacion)
61	Unidad_6 | Unidad_7	Inter	29.006	42
13	Unidad_10 | Unidad_12	Inter	27.146	30
46	Unidad_3 | Unidad_4	Inter	26.596	41
41	Unidad_2 | Unidad_5	Inter	25.719	32
11	Unidad_1 | Unidad_9	Inter	25.282	38
63	Unidad_6 | Unidad_9	Inter	23.966	36
26	Unidad_11 | Unidad_5	Inter	23.718	35
60	Unidad_5 | Unidad_9	Inter	23.448	31
31	Unidad_12 | Unidad_2	Inter	23.324	37
53	Unidad_4 | Unidad_6	Inter	22.899	28
2	Unidad_1 | Unidad_11	Inter	22.548	39
10	Unidad_1 | Unidad_8	Inter	22.438	37
4	Unidad_1 | Unidad_2	Inter	21.868	27
29	Unidad_11 | Unidad_8	Inter	21.657	38
28	Unidad_11 | Unidad_7	Inter	21.008	35
7	Unidad_1 | Unidad_5	Inter	20.750	22
54	Unidad_4 | Unidad_7	Inter	20.369	32
23	Unidad_11 | Unidad_2	Inter	20.187	30
59	Unidad_5 | Unidad_8	Inter	20.120	28
34	Unidad_12 | Unidad_5	Inter	20.088	30
40	Unidad_2 | Unidad_4	Inter	20.058	29
30	Unidad_11 | Unidad_9	Inter	19.830	28

Tabla 13: Violaciones de tiempo máximo de tipo Inter

ExampleSet (12 examples, 0 special attributes, 4 regular attributes)

Row No.	Merged sorted	Type	sum(Violacion)	count(Violacion)
4	Unidad_12 | Unidad_12	Intra	25.465	37
11	Unidad_8 | Unidad_8	Intra	25.032	35
1	Unidad_1 | Unidad_1	Intra	23.843	34
10	Unidad_7 | Unidad_7	Intra	22.644	37
6	Unidad_3 | Unidad_3	Intra	22.483	36
8	Unidad_5 | Unidad_5	Intra	21.491	34
2	Unidad_10 | Unidad_10	Intra	20.765	36
12	Unidad_9 | Unidad_9	Intra	19.302	36
3	Unidad_11 | Unidad_11	Intra	17.947	34
7	Unidad_4 | Unidad_4	Intra	17.162	29
9	Unidad_6 | Unidad_6	Intra	16.749	30
5	Unidad_2 | Unidad_2	Intra	11.614	27

Tabla 14: Violaciones de tiempo máximo de tipo Intra

ExampleSet (12 examples, 0 special attributes, 4 regular attributes)

Row No.	Merged sorted	Type	sum(Violacion)	count(Violacion)
8	Port | Unidad_5	IO	25.943	41
1	Port | Unidad_1	IO	25.206	31
6	Port | Unidad_3	IO	23.381	35
2	Port | Unidad_10	IO	21.120	31
7	Port | Unidad_4	IO	18.946	34
9	Port | Unidad_6	IO	18.234	32
12	Port | Unidad_9	IO	17.793	27
3	Port | Unidad_11	IO	17.755	26
5	Port | Unidad_2	IO	17.539	34
4	Port | Unidad_12	IO	16.849	27
10	Port | Unidad_7	IO	15.766	31
11	Port | Unidad_8	IO	10.001	19

Tabla 15: Violaciones de tiempo máximo de tipo IO

ExampleSet (1 example, 0 special attributes, 4 regular attributes)

Row No.	Merged sorted	Type	sum(Violacion)	count(Violacion)
1	Port | Port	FF	15.549	25

Tabla 16: Violaciones de tiempo máximo de tipo FF

5.1.5 Evaluación

Al analizar los datos obtenidos se pueden dar las siguientes recomendaciones:

- Inter (Tabla 13): Debe hacerse un esfuerzo por acercar los siguientes pares de unidades: 6 y 7, 10 y 12, 3 y 4, 2 y 5, 1 y 9.
- Intra (Tabla 14): Deben configurarse restricciones de localización de tal forma que las celdas de las unidades 12 y 8 se posicionen más cerca entre sí.
- IO (Tabla 15): Las unidades 5 y 1 deben colocarse más cerca de los extremos de la partición, más cerca de los puertos.

5.1.6 Implementación

Al compartir los datos obtenidos y las recomendaciones con el diseñador de esta partición se decidió modificar el diseño de tal forma que la localización de las unidades se alinee, en lo posible, con las recomendaciones dadas. Se consideraron otros elementos del diseño de la partición, y al final se logró obtener una localización similar a la mostrada en la Figura 17.

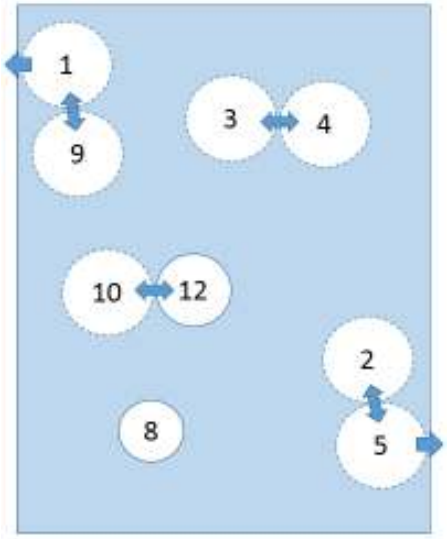

Figura 17: Topología de la partición luego de implementar los cambios sugeridos

5.2 Predicción de la Complejidad de las Particiones

5.2.1 Entender el Negocio

El proceso de diseño estructurado de circuitos VLSI se basa en el uso de herramientas de diseño asistido por computadoras. Estas herramientas ejecutan algoritmos de optimización en las etapas de síntesis, de colocación y de ruteo. Por la gran cantidad de celdas y por la complejidad de los diseños, estas herramientas pueden tardar de varias horas a varios días en completar sus procesos.

Intuitivamente, se puede deducir que entre más complejo es un circuito integrado, más tiempo va a durar ejecutando la herramienta de diseño. En la práctica se ha comprobado que esta relación directa entre complejidad y tiempo de ejecución efectivamente existe. Adicionalmente, también se podría intuir que para poder completar las tareas de diseño en una partición compleja, las herramientas van a requerir más recursos de máquina, específicamente memoria. De esta forma, se podría pensar que el consumo de memoria también podría ser una buena medida de la complejidad de una partición.

El primer paso de esta investigación fue el de tratar de verificar que efectivamente ambas métricas (tiempo de ejecución y consumo de memoria) se encuentran correlacionadas entre sí, de tal forma que cualquiera de ellas podría ser usada como medida de la complejidad de una partición. En la Figura 18 se puede observar que efectivamente se tiene una muy buena correlación entre ambas mediciones (factor de correlación = 0.926), con lo que se puede concluir que cualquiera de las dos podría utilizarse.

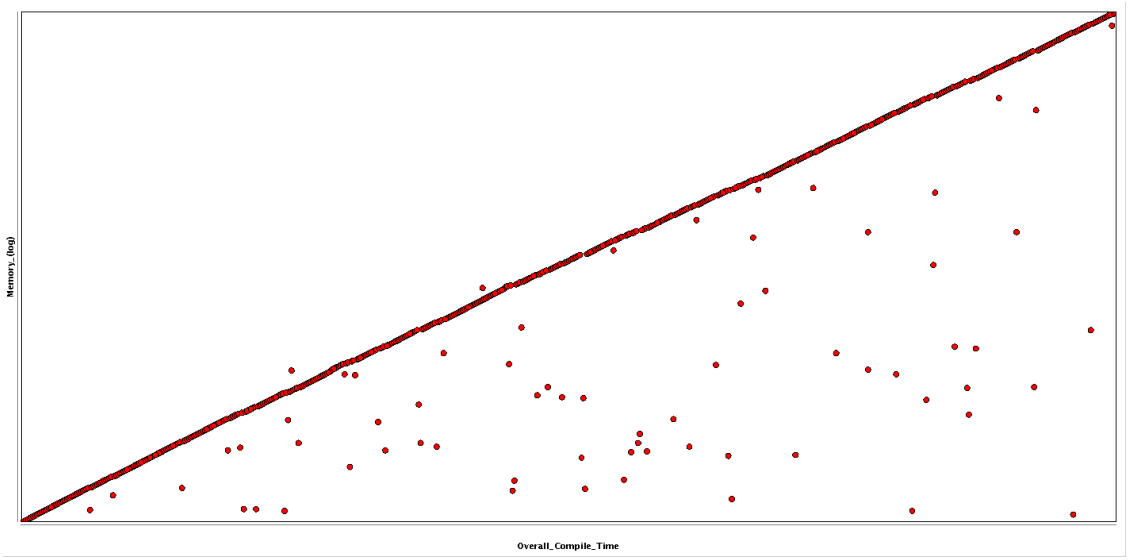

Figura 18: Uso de memoria de las herramientas de diseño versus tiempo de ejecución

El siguiente paso es validar que efectivamente el tiempo de ejecución y el consumo de memoria se relacionan con la complejidad de la partición. Para lograr esto, se verificó la correlación entre estas dos medidas e indicadores de calidad de una partición, como lo son el TNS y la cantidad de DRVs (Design Rule Violations, violaciones a las reglas de diseño por sus siglas en inglés). La Tabla 17 muestra que en general se tiene una buena correlación, siendo un poco mejor en el caso del tiempo de ejecución. La Figura 19 muestra claramente que aún y cuando ambas variables tienen una buena correlación, el tiempo de ejecución tiene menos ruido.

	Tiempo de Ejecución	Consumo de Memoria
TNS	0.924	0.852
Violaciones a Reglas de Diseño	0.680	0.641

Tabla 17: Correlación entre métricas de complejidad e indicadores de calidad

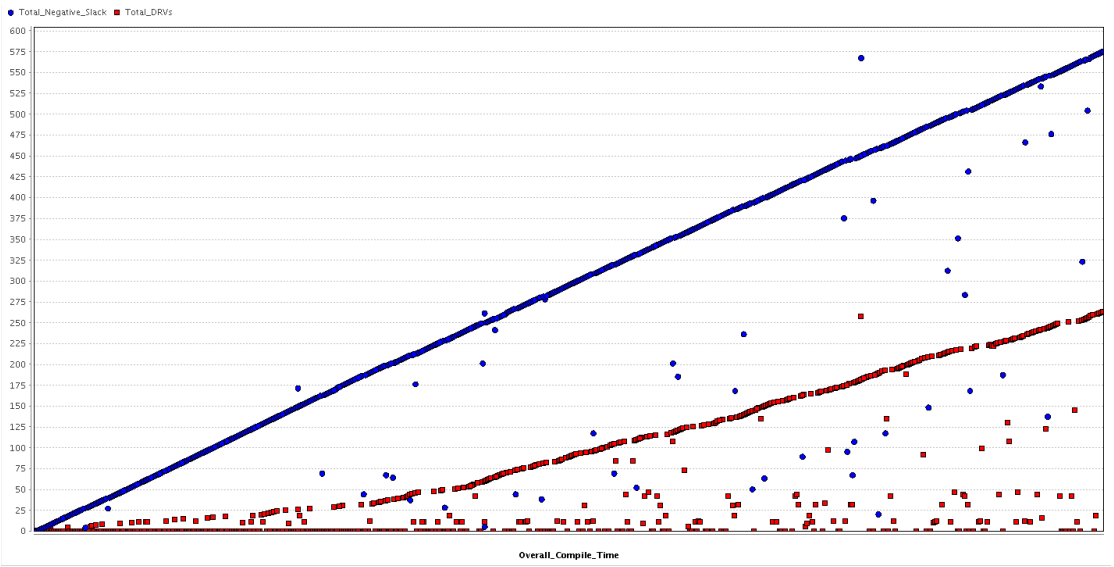

Figura 19: Correlación entre TNS (azul), violaciones a reglas de diseño (rojo) y el tiempo de ejecución

De esta forma se seleccionó el tiempo de ejecución como la forma de medir la complejidad de una partición.

Ahora, si el tiempo de ejecución es una forma de medir la complejidad de una partición, ¿qué elementos intrínsecos de la partición nos podrían ayudar a predecirla? Y, ¿de qué nos serviría poder predecir la complejidad esperada para una partición?

Nosotros planeamos contestar la primer interrogante al desarrollar un modelo predictivo de complejidad de particiones. Con respecto a la segunda pregunta, hay una gran cantidad de beneficios que se pueden obtener de un modelo como el propuesto. Por ejemplo:

- Se pueden tomar mejores decisiones en las etapas iniciales del diseño de tal forma que se homogenice la complejidad de las particiones, y de esta forma no tengamos particiones muy sencillas y particiones de mucha complejidad. Esto nos ayudaría a tener una menor variabilidad en los tiempos de ejecución de las herramientas, con lo que se logra reducir el tiempo total del proceso de diseño.

- Se puede optimizar la asignación de particiones a los diseñadores, de tal forma que los diseñadores más experimentados pueden enfocarse en las particiones de mayor

complejidad, mientras que los diseñadores con menos experiencia pueden trabajar en las particiones más sencillas.

- Se pueden generar mejores estimados de tiempos de ejecución de las particiones al saber su complejidad. De esta forma, se puede lograr una mejor planeación de la duración del proyecto.

- También, al poder identificar desde el inicio las particiones más sencillas (que por ende son las que tienen un menor tiempo de ejecución), se pueden utilizar como "vehículos de prueba" para validar que el entorno de ejecución funcione correctamente.

5.2.2 Entender los Datos

Para poder desarrollar el modelo predictivo se escogieron los siguientes datos a nivel de partición como posibles predictores.

- *Área de buffers e inversores* (Buf/Inv_Area): es el área total dentro de la partición asignada a celdas tipo buffer o inversos.
- *Área total de las celdas* (Cell_Area): es el área de todas las celdas dentro de la partición.
- *Área de celdas combinacionales* (Combinational_Area): es el área de todas las celdas combinacionales dentro de la partición.
- *Área total del diseño* (Design_Area): es el área total de la partición
- *Área de macros* (Macro/Black_Box_Area): Los macros son elementos de diseño que para efectos prácticos funcionan como cajas negras para los diseñadores.
- *Área de los cables* (Net_Area): el área cubierta por los cables de interconexión
- *Área de no combinacionales* (Noncombinational_Area): es el área total cubierta por celdas no combinacionales.
- *Área de buffers* (Total_Buffer_Area): el área total de celdas tipo buffer en la partición.

- *Área de inversores* (Total_Inverter_Area): el área total de celdas tipo inversor en la partición.
- *Cantidad de buffers e inversores* (Buf/Inv_Cell_Count): cantidad de celdas tipo buffer e inversor en la partición.
- *Cantidad de buffers* (Buf_Cell_Count): cantidad de celdas tipo buffer en la partición.
- *Cantidad de celdas combinacionales* (Combinational_Cell_Count): cantidad de celdas combinacionales dentro de la partición.
- *Cantidad de celdas jerárquicas* (Hierarchical_Cell_Count): cantidad de celdas dedicadas a poder realizar cambios de jerarquía dentro de la partición.
- *Cantidad de inversores* (Inv_Cell_Count): cantidad de celdas tipo inversor en la partición
- *Cantidad de celdas estándar* (Leaf_Cell_Count): cantidad de celdas que pertenecen a las librerías estándar.
- *Cantidad de celdas secuenciales* (Sequential_Cell_Count): cantidad de celdas secuenciales dentro de la partición.
- *Ancho total de canal para transistores tipo P* (Total_PWidth_Zp_____): ancho total de todos los canales para los transistores tipo P.
- *Ancho total de canal para transistores tipo N* (Total_NWidth_Zn_____): ancho total de todos los canales para los transistores tipo N.
- *Cantidad de macros* (Macro_Count): Cantidad de macros dentro de la partición.
- *Cantidad de puertos* (Hierarchical_Port_Count): Cantidad total de puertos en la partición. Un puerto es un pin que comunica la partición con otra partición.
- *Niveles de lógica máximos IO* (Levels_of_Logic_(IO)): cantidad máxima de niveles de lógica en los circuitos lógicos tipo IO
- *Niveles de lógica máximos Inter-Intra* (Levels_of_Logic_(Reg2Reg)): cantidad máxima de niveles de lógica en los circuitos lógicos tipo Inter o Intra.
- *Niveles de lógica máximos* (Levels_of_Logic): cantidad máxima de niveles de lógica en la partición.

- *Utilización* (Utilization): utilización de la partición. La utilización se define como el área total de las celdas dividida por el área total de la partición.
- *Periodo del reloj para el circuito más crítico* (Critical_Path_Clk_Period): el periodo del reloj del circuito lógico más lento de la partición.
- *Optimización de mapeo* (Mapping_Optimization): Es una medida del nivel de optimización en el mapeo de celdas.
- *Tamaño promedio del canal* (Avg_Z_per_gate_____): promedio de los anchos de los canales de todos los transistores dentro de la partición.
- *Porcentaje nominal* (NOM_PER): porcentaje de celdas nominales dentro de la partición. Las celdas nominales son las que usan transistores nominales.
- *Porcentaje UV1* (UV1_PER): porcentaje de celdas UV1 dentro de la partición. Las celdas UV1 son las que usan transistores UV1.
- *Porcentaje UV2* (UV2_PER): porcentaje de celdas UV2 dentro de la partición. Las celdas UV2 son las que usan transistores UV2.
- *Ancho de canal normalizado* (Normalized_Z): es una medida normalizada del ancho de canal promedio dentro de la partición.

Todos estos datos se extrajeron directamente de la base de datos MDP, para la etapa de síntesis del proceso de diseño estructurado.

5.2.3 Preparar los Datos

Debido a la estructura de la base de datos MDP, donde los datos se almacenan en pares llave-valor, cuando se extraigan los datos tenemos que asegurarnos que la consulta se haga de tal forma que se obtenga una columna por cada llave. De esta forma, se obtiene un conjunto de datos con tantas columnas como categorías de datos se tengan, en lugar de obtener únicamente dos columnas (llave, valor). Este formato de tabla es el que se necesita para poder realizar los análisis de datos.

Adicionalmente se va a categorizar el tiempo de ejecución en cuatro grupos: baja complejidad, mediana complejidad, alta complejidad y muy alta complejidad.

5.2.4 Modelado

Para este análisis hicieron dos ejercicios de modelado. El primero se llevó a cabo para generar los grupos de complejidad y el modelo usado en RapidMiner se muestra en la Figura 20.

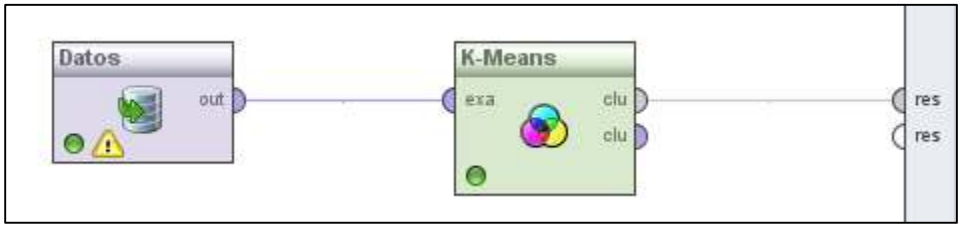

Figura 20: Proceso para generar grupos de complejidad

Aquí vemos que hay dos operadores. El primer operador (Datos) se utiliza para cargar los datos de tiempo de ejecución por partición en RapidMiner. El segundo operador utiliza el algoritmo de K-Means para generar los grupos especificados. En este caso, se generaron cuatro grupos. Al ejecutar este proceso se obtiene que de las 918 particiones de las que se extrajeron datos, 504 fueron clasificadas como de baja complejidad, 314 de mediana complejidad, 87 de alta complejidad y 13 de muy alta complejidad. La Figura 21 muestra el resultado de la aplicación del algoritmo.

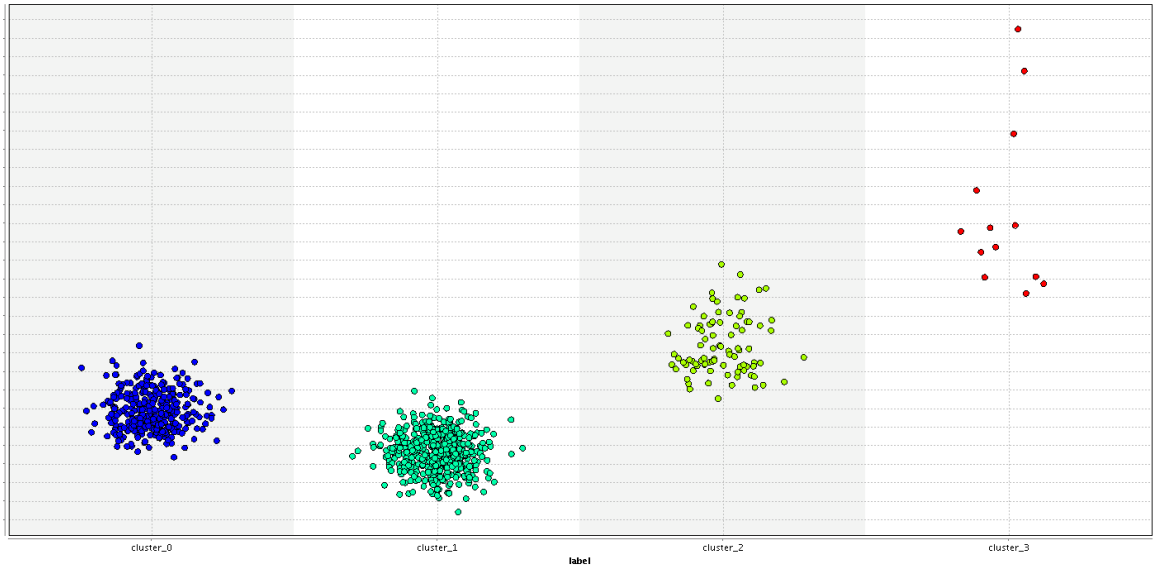

Figura 21: Particiones separadas por grupos de complejidad

Con estos datos podemos proceder a generar el modelo predictivo. El primer paso que se debe hacer es revisar los datos de los predictores. La Tabla 18 muestra los metadatos de las variables predictoras. Al analizar estos datos, es importante resaltar lo siguiente:

- Las variables predictoras son de múltiples tipos: reales, enteras, polinomiales.
- La variable que se desea predecir es de tipo polinominal.
- Hay una gran cantidad de datos faltantes, lo que se puede observar en la columna "Missings". Por ejemplo, en el caso de la variable "Cell_Area" hay 17 registros con este dato faltante.

También es importante mencionar que aunque los valores estadísticos y de rango se ocultaron por razones de confidencialidad, esto no impacta el análisis que se desea realizar.

Con esta información, y usando como referencia la Tabla 11 y la Tabla 24, se puede concluir que podemos usar los algoritmos de árbol de decisión o el de red neuronal.

ExampleSet (898 examples, 2 special attributes, 27 regular attributes)

Role	Name	Type	Statistics	Range	Missings
id	Partition_Milestone_Model_Label	polynominal			0
label	cluster	nominal			0
regular	Bufinv_Area	real			17
regular	Cell_Area	real			17
regular	Combinational_Area	real			17
regular	Design_Area	real			17
regular	MacroBlack_Box_Area	integer			17
regular	Net_Area	integer			17
regular	Noncombinational_Area	real			17
regular	Total_Buffer_Area	real			17
regular	Total_Inverter_Area	real			17
regular	Bufinv_Cell_Count	integer			17
regular	Buf_Cell_Count	integer			17
regular	Combinational_Cell_Count	integer			17
regular	CT_Bufinv_Cell_Count	integer			17
regular	Hierarchical_Cell_Count	integer			17
regular	Inv_Cell_Count	integer			17
regular	Leaf_Cell_Count	integer			17
regular	Sequential_Cell_Count	integer			17
regular	Total_PWidth_Zp_____	integer			18
regular	Total_NWidth_Zn_____	integer			18
regular	Macro_Count	integer			17
regular	Hierarchical_Port_Count	integer			17
regular	Levels_of_Logic_IO	integer			18
regular	Levels_of_Logic_Reg2Reg	integer			17
regular	Levels_of_Logic	integer			17
regular	Critical_Path_Clk_Period	integer			26
regular	Mapping_Optimization	real			17
regular	Avg_Z_per_gate_____	integer			18

Tabla 18: Metadatos de las variables predictoras.

El proceso que se presenta en la Figura 22 se utilizó con dos objetivos:

1) Determinar cuál modelo predictivo es más exacto entre el árbol de decisión y la red neuronal.

2) Determinar cuáles de las variables predictoras son más importantes.

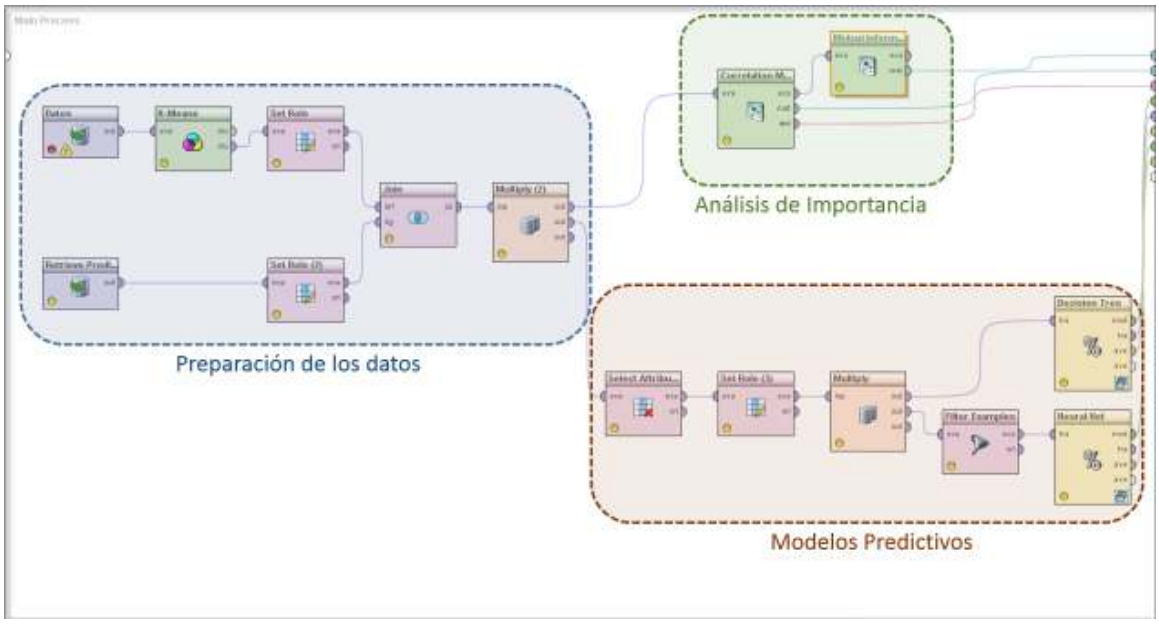

Figura 22: Proceso para generar modelo predictivo de complejidad de particiones

Se va a explicar cada una de las etapas del proceso de la Figura 22.

- *Preparación de los datos*: se generó un conjunto de datos donde se unió el análisis de grupos de complejidad (que se hizo por medio del algoritmo K-Means como se mencionó anteriormente) con los datos de las variables predictoras.
- *Análisis de importancia*: para determinar cuáles de las variables predictoras son más importantes, se utilizaron dos técnicas, una de ellas basada en una matriz de correlación y la otra en una matriz de información mutua.
- *Modelos predictivos*: se implementan los operadores de árbol de decisión y de red neuronal. Es importante mencionar que para el caso de la red neuronal, fue necesario eliminar los datos vacíos con el uso del operador “Filter Examples”.

5.2.5 Evaluación

De este proceso se genera una gran cantidad de información. Lo primero está relacionado con la exactitud de los dos modelos predictivos: el árbol de decisión y la red neuronal. En la Figura 23 se aprecia el resultado de evaluar el rendimiento del modelo de red neuronal, y en la Figura 24 el del árbol de decisión. Ambos modelos son bastante buenos, con precisiones superiores al 70%. Sin embargo, el modelo basado en la red neuronal tiene una mayor precisión.

accuracy: 75.52% +/- 5.29% (mikro: 75.52%)

	true cluster_0	true cluster_1	true cluster_2	true cluster_3	class precision
pred. cluster_0	196	56	35	1	68.06%
pred. cluster_1	91	411	1	0	81.71%
pred. cluster_2	18	1	46	8	63.01%
pred. cluster_3	2	0	0	4	66.67%
class recall	63.84%	87.82%	56.10%	30.77%	

Figura 23: Vector de rendimiento del modelo de red neuronal.

accuracy: 71.83% +/- 4.86% (mikro: 71.83%)

	true cluster_0	true cluster_1	true cluster_2	true cluster_3	class precision
pred. cluster_0	276	118	84	13	56.21%
pred. cluster_1	37	369	1	0	90.66%
pred. cluster_2	0	0	0	0	0.00%
pred. cluster_3	0	0	0	0	0.00%
class recall	88.18%	75.77%	0.00%	0.00%	

Figura 24: Vector de rendimiento del modelo de árbol de decisión.

En la Tabla 19 se presenta un resumen de los factores de correlación e importancia mutua. Vemos que en general ambas técnicas identifican las mismas variables como las más importantes (letras blancas en la tabla). Se puede concluir que en general, entre mayor es la cantidad de celdas, mayor va a ser la complejidad de la partición. También es interesante notar que al tener una optimización de mapeo mayor, también aumenta la complejidad de la partición.

	Factor de Correlación	Factor de Información Mutua
Mapping_Optimization	0.797	0.205
Combinational_Area	0.578	0.380
Leaf_Cell_Count	0.534	0.293
Combinational_Cell_Count	0.532	0.304
Cell_Area	0.527	0.308
Design_Area	0.527	0.308
Total_Buffer_Area	0.521	0.290

	Factor de Correlación	Factor de Información Mutua
Levels_of_Logic	0.521	0.248
BufInv_Area	0.516	0.277
Levels_of_Logic_Reg2Reg	0.514	0.238
BufInv_Cell_Count	0.479	0.243
Buf_Cell_Count	0.463	0.239
Total_Inverter_Area	0.462	0.230
Inv_Cell_Count	0.450	0.238
Sequential_Cell_Count	0.399	0.175
Noncombinational_Area	0.359	0.185
Hierarchical_Port_Count	0.269	0.162
Macro_Count	0.232	0.155
Total_NWidth_Zn_____	0.166	0.127
Total_PWidth_Zp_____	0.163	0.127
Levels_of_Logic_IO	0.152	0.057
Avg_Z_per_gate_____	0.140	0.133
Hierarchical_Cell_Count	0.104	0.079
MacroBlack_Box_Area	-0.003	0.002
Critical_Path_Clk_Period	-0.017	0.041
CT_BufInv_Cell_Count	-0.074	-0.001

Tabla 19: Factores de correlación e información mutua

5.2.6 Implementación

Se logró generar un modelo predictivo para complejidad de partición. Este modelo utiliza datos de las primeras corridas de síntesis para predecir la complejidad de la partición, por lo que se puede usar desde las primeras etapas del proceso de diseño estructurado.

También, se logró identificar una fuerte correlación entre la cantidad de celdas de una partición y la complejidad de la misma. Basados en este hallazgo se van a dar recomendaciones con respecto a la cantidad máxima de celdas por partición en proyectos futuros.

5.3 Análisis de Predictores en la Tasa de Crecimiento de la Cantidad de Celdas

5.3.1 Entender el Negocio

En las diferentes etapas que comprenden el proceso de diseño estructurado de circuitos VLSI se da un crecimiento en la cantidad de celdas que puede estar entre un 10 y un 20 por ciento al comparar la cantidad final de celdas con la cantidad de celdas en la etapa de síntesis.

Esto se debe a que durante el proceso de diseño y convergencia se agregan celdas que ayudan a mejorar las condiciones de temporización de los circuitos, también hay celdas que se agregan para resolver violaciones de reglas de diseño, y así hay otras razones por las cuales se agregan celdas. Esta tendencia se puede apreciar en la Figura 25, en la cual se omite la escala del eje Y por razones de confidencialidad. Es importante mencionar que aún y cuando la tendencia general es de crecimiento, si existen pasos en los cuales la cantidad de celdas se reducen, como por ejemplo los pasos 4 y 5 en la Figura 25.

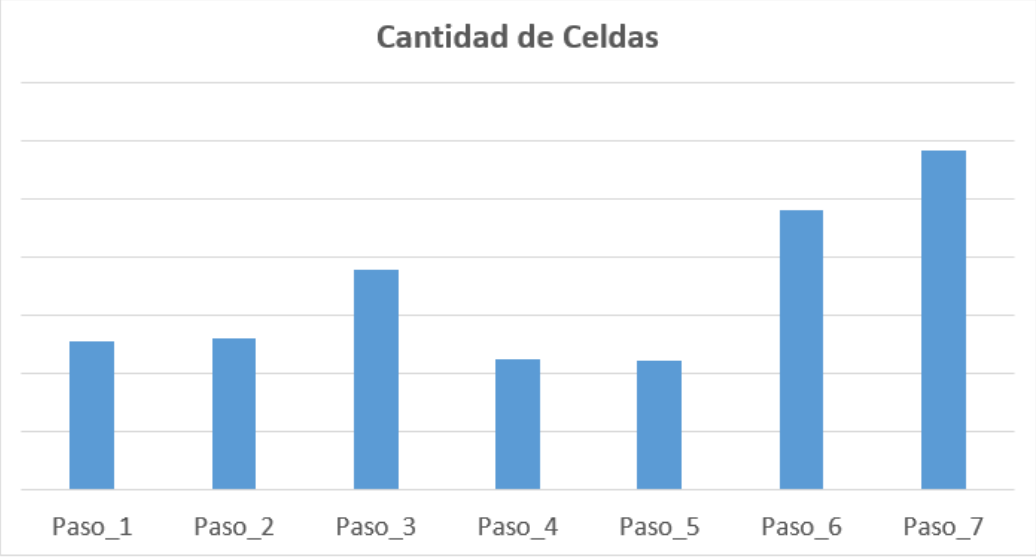

Figura 25: Crecimiento en cantidad de celdas en diferentes etapas del flujo de diseño estructurado

Al calcular la tasa de crecimiento entre el Paso_7 y el Paso_1 se obtiene la distribución de la Figura 26. De esta figura es importante resaltar la gran cantidad de valores extremos que se tienen.

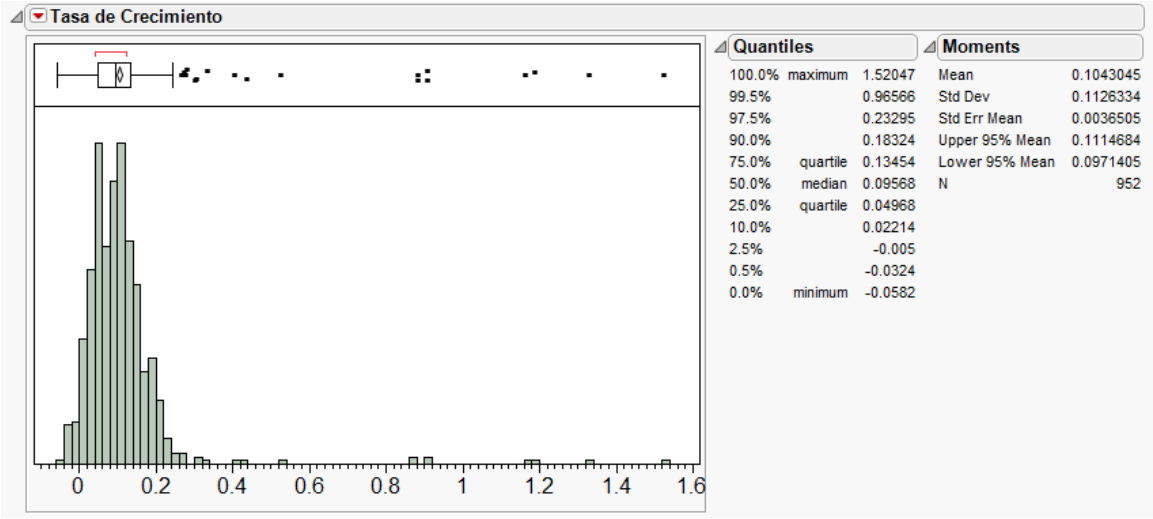

Figura 26: Distribución de tasa de crecimiento de celdas

Así, es deseable que esta tasa de crecimiento sea lo menor posible, de tal forma que se minimice la complejidad de la partición (tal y como se pudo concluir del análisis anterior).

Si se logra entender cuáles son los elementos que modulan este crecimiento de cantidad de celdas, se pueden hacer esfuerzos para controlarlos y de esta forma minimizar el crecimiento.

También, con un modelo predictivo de crecimiento en cantidad de celdas, se podrían hacer mejores estimados de área por asignar a cada partición, de tal forma que evitemos particiones que terminen con una congestión muy alta por tener una cantidad mayor de celdas que las que se pueden soportar en un área determinada.

5.3.2 Entender los Datos

Para este análisis se van a utilizar los mismos datos que se usaron en el análisis anterior. Una descripción detallada de los mismos se puede obtener de la sección 5.2.2.

5.3.3 Preparar los Datos

Los metadatos pueden observarse en la Figura 27. Es importante notar que:

- Para poder realizar el análisis de datos correctamente, es necesario cambiar el rol del dato "Key" por ID.
- Existen muchas columnas que tienen datos faltantes. Dependiendo del tipo de análisis que se desee realizar, es posible que las tuplas con datos faltantes tengan que descartarse. Sin embargo, esto no es un problema ya que estas tuplas representan únicamente un 2.8% del total de tuplas.
- En el caso de la tasa de crecimiento de celdas, hay valores extremos que se tienen que remover del conjunto de datos.
- Al igual que en el análisis anterior, por razones de confidencialidad los datos estadísticos y de rango no se muestran.

Role	Name	Type	Statistics	Range	Missings
regular	Key	polynominal			0
regular	Cell Growth Rate	real			0
regular	Buffinv_Area	real			17
regular	Cell_Area	real			17
regular	Combinational_Area	real			17
regular	Design_Area	real			17
regular	MacroBlack_Box_Area	integer			17
regular	Net_Area	integer			17
regular	Noncombinational_Area	real			17
regular	Total_Buffer_Area	real			17
regular	Total_Inverter_Area	real			17
regular	Buffinv_Cell_Count	integer			17
regular	Buf_Cell_Count	integer			17
regular	Combinational_Cell_Count	integer			17
regular	CT_Buffinv_Cell_Count	integer			17
regular	Hierarchical_Cell_Count	integer			17
regular	Inv_Cell_Count	integer			17
regular	Leaf_Cell_Count	integer			17
regular	Sequential_Cell_Count	integer			17
regular	Total_PWidth_Zp_____	integer			18
regular	Total_NWidth_Zn_____	integer			18
regular	Macro_Count	integer			17
regular	Hierarchical_Port_Count	integer			17
regular	Levels_of_Logic_IO	integer			18
regular	Levels_of_Logic_Reg2Reg	integer			17
regular	Levels_of_Logic	integer			17
regular	Cntical_Path_Clk_Period	integer			27
regular	Mapping_Optimization	real			17
regular	Avg_Z_per_gate_____	integer			18

Figura 27: Metadatos para el análisis de crecimiento en cantidad de celdas

5.3.4 Modelado

Para este caso se quieren identificar las variables predictoras más importantes y también generar un modelo predictivo para el crecimiento de celdas

El primer objetivo se puede lograr con una matriz de correlaciones, y para el segundo se van a generar dos modelos, uno utilizando el algoritmo de redes neuronales y otro con una

regresión lineal. El proceso desarrollado para generar estos modelos se puede observar en la Figura 28.

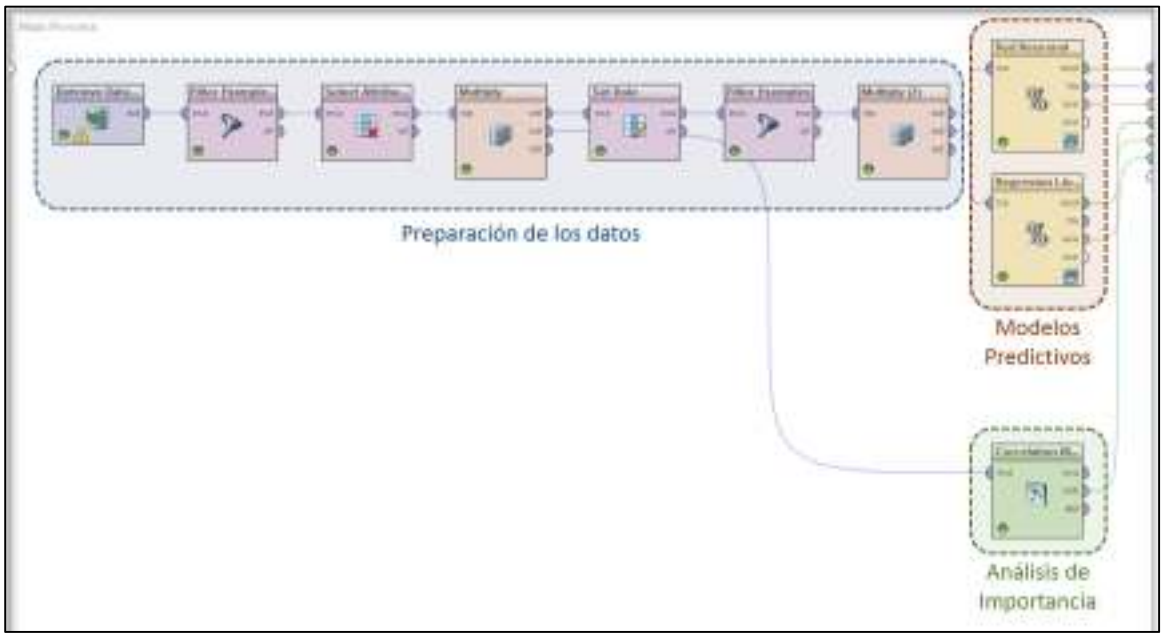

Figura 28: Proceso de RapidMiner para el análisis de tasa de crecimiento de celdas

Las etapas principales de este proceso se explican a continuación:

- *Preparación de los datos*: se eliminan los valores extremos de la tasa de crecimiento, usando como límite superior el percentil 97.5. También se modifica el rol del attributo "Key" y por último se remueven las tuplas con valores faltantes. La Figura 29 muestra la distribución de las tasas de crecimiento luego de completar esta etapa. Es importante notar en esta figura que ya no se tiene valores extremos.

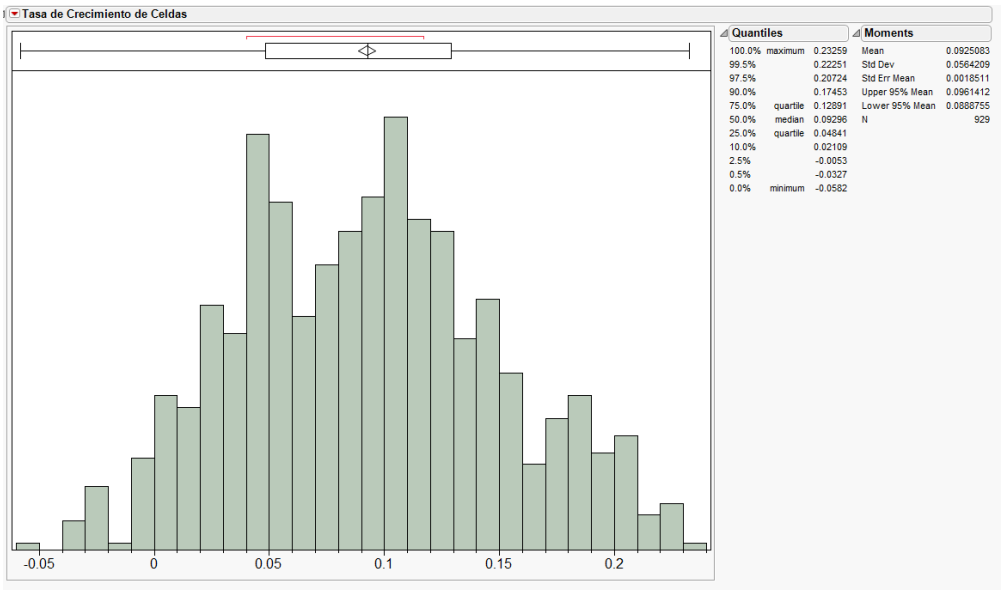

Figura 29: Distribución de tasa de crecimiento de celdas luego de eliminar valores extremos y tuplas con valores faltantes

- *Modelos Predictivos*: Se calcula la efectividad de los diferentes modelos implementados por medio de un operador compuesto, el cual requiere de un operador para generar el modelo predictivo, un operador para aplicar el modelo y un operador para medir el rendimiento del modelo. En la Figura 30 se muestra el detalle interno de cómo está construido este operador compuesto para el modelo de regresión lineal. Un operador compuesto similar se utilizó para el modelo de red neuronal.

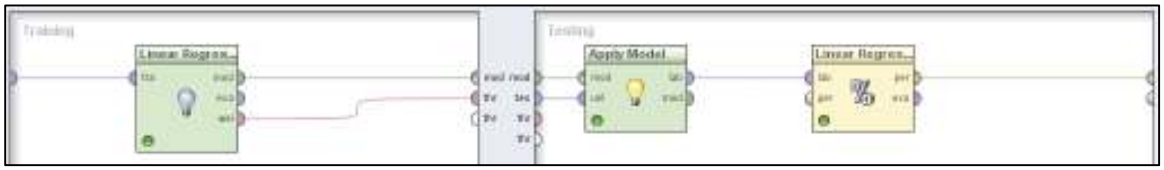

Figura 30: Detallen interno del operador compuesto para el modelo de regresión lineal

- *Análisis de Importancia*: se generó una matriz de correlación.

5.3.5 Evaluación

Los factores de correlación obtenidos se muestran en la Tabla 20. Se puede observar que ninguna variable presenta un factor de correlación alto, por lo que no se puede considerar que ninguna de estas variables sea un buen predictor de la tasa de crecimiento de celdas.

	Factor de correlación
Total_PWidth_Zp_____	-0.34976
Total_NWidth_Zn_____	-0.34906
Avg_Z_per_gate_____	0.213305
MacroBlack_Box_Area	-0.16715
Critical_Path_Clk_Period	-0.12673
Noncombinational_Area	0.12169
Combinational_Cell_Count	-0.11937
Combinational_Area	-0.10964
Inv_Cell_Count	-0.09536
Leaf_Cell_Count	-0.09433
Levels_of_Logic_Reg2Reg	0.07145
Total_Inverter_Area	-0.07108
Levels_of_Logic	0.069779
BufInv_Cell_Count	-0.05339
Sequential_Cell_Count	0.048365
BufInv_Area	-0.02911
Buf_Cell_Count	0.02846
Levels_of_Logic_IO	-0.02528
Macro_Count	0.023192
Total_Buffer_Area	0.014667
CT_BufInv_Cell_Count	-0.01162
Hierarchical_Cell_Count	-0.01139
Mapping_Optimization	-0.00879
Cell_Area	0.003538
Design_Area	0.003538

Tabla 20: Factores de Correlación para el análisis de crecimiento de cantidad de celdas

En la Tabla 21 se presentan los errores de los modelos predictivos. Como se puede ver de esta tabla, el modelo basado en la regresión lineal presenta un error menor que el modelo basado en la Red Neuronal.

Error	Red Neuronal	Regresión Lineal
Raíz cuadrada del error cuadrático medio	0.055 ± 0.006	0.047 ± 0.001
Error cuadrático medio	0.003 ± 0.001	0.002 ± 0.000

Tabla 21: Errores de los modelos predictivos

Al comparar estos errores con el promedio (0.093) y la desviación estándar (0.056) de la distribución de tasas de crecimiento de celdas filtradas (Figura 29), se puede observar que la raíz cuadrada del error cuadrático medio representa un 50.5% del promedio de tasas de crecimiento, lo que es sumamente alto.

Con estos errores tan altos y con los factores de correlación tan bajos, podemos concluir que las variables seleccionadas no son buenos predictores para la tasa de crecimiento de celdas.

5.3.6 Implementación

En este caso es necesario buscar otras variables predictoras, ya que las que se seleccionaron inicialmente no tienen una buena correlación con la tasa de crecimiento de celdas. Adicionalmente, los modelos de predicción generados a partir de estas variables no son precisos, ya que tienen errores demasiado altos.

5.4 Análisis de Predictores para Variables del QoR

5.4.1 Entender el Negocio

Durante el proceso de diseño de circuitos integrados se generan múltiples indicadores que miden la calidad del circuito. El proceso de diseño en sí trata de que muchos de estos indicadores se lleven a cero, momento en el cual se dice que el diseño ha convergido.

El lograr entender la relación que estos indicadores tienen con otros elementos del circuito es de suma importancia ya que puede ayudar a los diseñadores a converger su partición en menor tiempo.

El objetivo de este análisis es determinar si esas relaciones existen, y de ser así, determinar cuáles elementos del circuito tienen mayor impacto en las variables de QoR.

Para efectos de este análisis, se seleccionaron las siguientes variables del QoR:

- *TNS*. Esta variable representa la suma total del tiempo por el que no se cumple con los requisitos de temporización de todos los circuitos lógicos. Se desea que el TNS llegue a cero una vez que el diseño ha convergido.

- *DRVs*. Es la suma de todas las violaciones a reglas de diseño. Al igual que en el caso del TNS, se desea que esta variable llegue a cero una vez que el diseño ha convergido.

- *Cortos*. Es la suma de la cantidad total de corto-circuitos que se tienen en el diseño. También, se desea que llegue a cero en el momento en que el diseño ha convergido.

- *Longitud total del alambrado del circuito*. Este indicador representa la longitud total de líneas de metal usadas para interconectar los transistores del diseño. Se desea minimizar el valor de este indicador.

- *Violaciones de carga máxima por celda*. Cada celda tiene una carga máxima que puede manejar. Esta carga máxima viene dada por las características de los transistores que forman la celda. Se desea que la cantidad de violaciones llegue a cero una vez que el diseño ha convergido.

- *Violaciones de transición máxima*. El tiempo de transición de una celda es el tiempo que la celda tarda en pasar de un cero lógico a un uno lógico, o viceversa.

Este tiempo es una función de la carga que está conectada a la salida de una celda. Para cumplir con los requerimientos de temporización del circuito, se define un tiempo máximo de transición. Una violación de tiempo máximo de transición se da cuando una celda tiene un tiempo de transición mayor al tiempo máximo de transición. Se desea que la cantidad de violaciones de transición máxima sea cero al converger el diseño.

5.4.2 Entender los Datos

Para este análisis se va a usar el mismo conjunto de variables predictoras que se han usado en los últimos análisis. En la sección 5.2.2 se da una descripción detallada de las mismas.

En el caso de las variables del QoR, es importante entender sus distribuciones para saber si tenemos valores extremos que se tengan que remover. En la Figura 31 se pueden ver las distribuciones de todas estas variables. Se puede observar que en todos los casos tenemos valores extremos que van a tener que ser removidos.

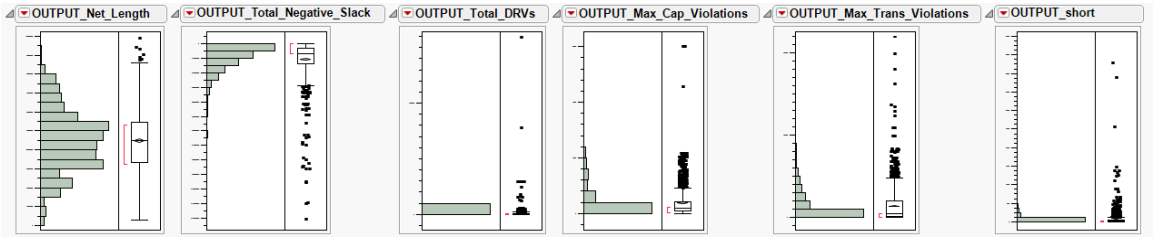

Figura 31: Distribuciones de las variables de QoR

5.4.3 Preparar los Datos

En el caso de este análisis se van a realizar las siguientes operaciones en los datos:

1. Remover las tuplas que tienen valores faltantes
2. Remover los valores extremos en las variables de QoR. Se va a utilizar un valor máximo de la media más dos desviaciones estándar como límite superior. Cualquier valor que esté por encima de este límite va a ser removido.

5.4.4 Modelado

Se va a calcular la matriz de correlaciones para cada una de las variables de QoR. El proceso utilizado para este análisis se muestra en la Figura 32.

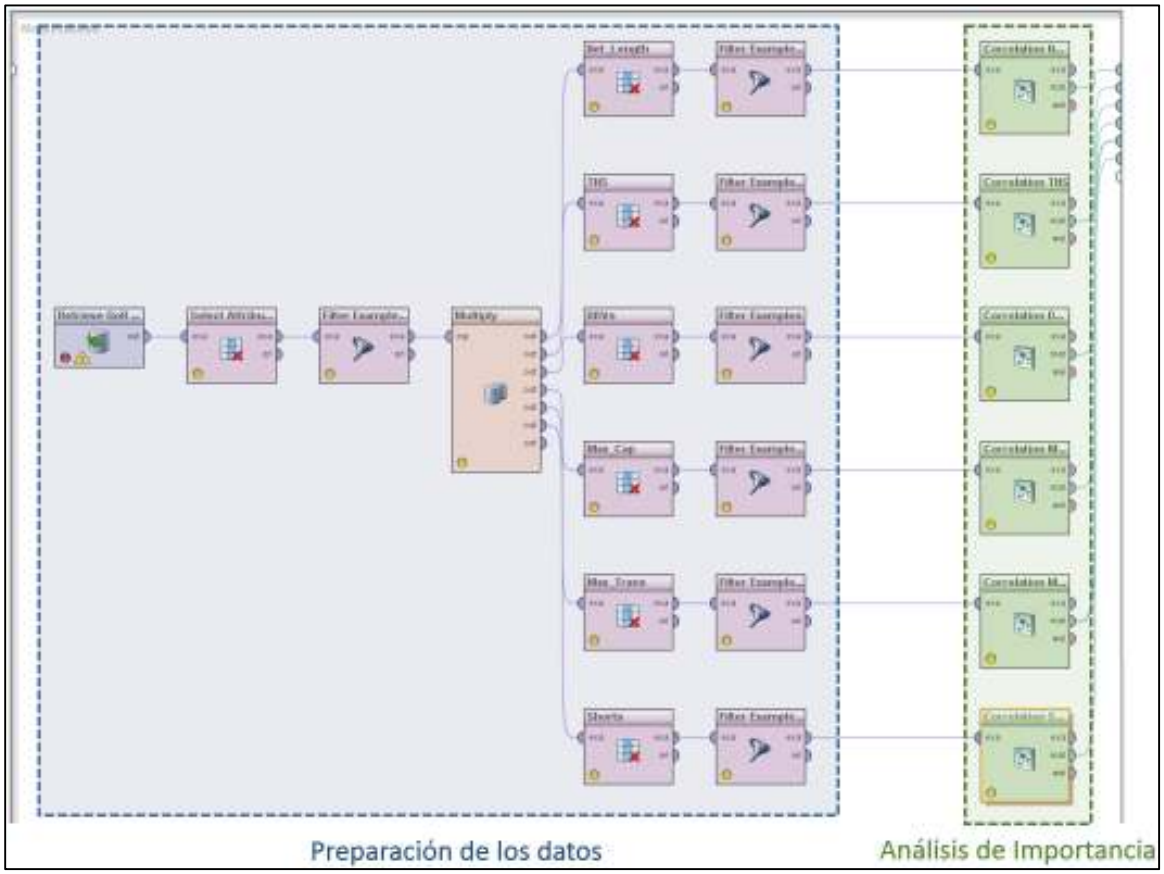

Figura 32: Proceso de análisis de correlación para variables de QoR

Al igual que en los casos anteriores, se tiene una etapa de preparación de datos y una etapa de análisis. En la etapa de preparación de datos se remueven los valores extremos y las tuplas con datos faltantes, mientras que en la etapa de análisis de importancia se generan las matrices de correlación para cada variable de QoR.

5.4.5 Evaluación

La Tabla 22 muestra los factores de correlación para las variables de QoR. A continuación se resume el análisis de los resultados para cada una de las variables de QoR.

- *TNS*. Existe una leve correlación con el área total de inversores.
- *DRVs*. No existe correlación fuerte con ninguna de las variables predictoras.
- *Cortos*. No existe correlación fuerte con ninguna de las variables predictoras.
- *Longitud total del alambrado del circuito*. Existe una fuerte correlación con las variables relacionadas a la cantidad de celdas. Este resultado es esperado, puesto que entre más celdas tenga en mi diseño, mayor va a ser la cantidad de recursos de ruteo que voy a necesitar para poder interconectar las celdas entre sí.
- *Violaciones de carga máxima por celda*. No existe correlación fuerte con ninguna de las variables predictoras.
- *Violaciones de transición máxima*. No existe correlación fuerte con ninguna de las variables predictoras.

	Net_Length	TNS	DRVs	Max_Cap	Max_Trans	Shorts
Total_Inverter_Area	0.92	-0.54	0.15	0.25	0.08	0.05
BufInv_Area	0.92	-0.48	0.28	0.31	0.29	0.17
BufInv_Cell_Count	0.89	-0.40	0.22	0.29	0.19	0.09
Total_PWidth_Zp_____	0.88	-0.49	0.18	0.21	0.09	0.02
Cell_Area	0.87	-0.46	0.17	0.18	0.08	0.00
Design_Area	0.87	-0.46	0.17	0.18	0.08	0.00
Normalized_Z	0.87	-0.46	0.17	0.20	0.08	0.02
Total_NWidth_Zn_____	0.85	-0.48	0.16	0.20	0.08	0.01
Inv_Cell_Count	0.81	-0.40	0.10	0.22	0.00	-0.02
Leaf_Cell_Count	0.80	-0.35	0.15	0.16	0.05	-0.01
Combinational_Area	0.79	-0.34	0.17	0.19	0.09	0.02
Noncombinational_Area	0.77	-0.48	0.14	0.12	0.05	-0.03
Total_Buffer_Area	0.77	-0.35	0.35	0.32	0.44	0.25
Combinational_Cell_Count	0.76	-0.31	0.14	0.15	0.05	-0.01
Buf_Cell_Count	0.76	-0.30	0.32	0.31	0.39	0.20
Hierarchical_Port_Count	0.70	-0.47	0.18	0.19	0.07	0.02
MacroBlack_Box_Area	0.60	-0.36	0.20	0.08	0.12	-0.02
Sequential_Cell_Count	0.56	-0.40	0.14	0.10	0.05	0.00
Mapping_Optimization	0.42	-0.14	0.18	0.15	0.10	0.11
Levels_of_Logic	0.33	-0.21	0.01	0.18	0.01	0.01

	Net_Length	TNS	DRVs	Max_Cap	Max_Trans	Shorts
Levels_of_Logic_IO	0.28	-0.18	-0.07	0.06	-0.05	-0.03
Utilization	0.28	-0.06	0.02	0.04	0.12	-0.03
CT_BufInv_Cell_Count	0.24	0.04	0.24	0.00	0.29	0.19
Macro_Count	0.22	-0.07	0.11	0.25	0.00	0.00
Levels_of_Logic_Reg2Reg	0.16	-0.09	0.10	0.20	0.07	0.04
Hierarchical_Cell_Count	0.12	-0.22	0.11	0.08	0.08	0.00
UV1_PER	0.01	-0.16	-0.02	0.03	-0.01	-0.04
NOM_PER	-0.01	0.16	0.02	-0.03	0.01	0.04
Critical_Path_Clk_Period	-0.06	-0.04	-0.02	0.00	0.04	-0.03
Avg_Z_per_gate_____	-0.06	0.01	-0.03	-0.02	-0.02	-0.01

Tabla 22: Matriz de Correlación para variables de QoR

5.4.6 Implementación

Se encontró una fuerte correlación entre la longitud total del alambrado del circuito y la cantidad de celdas. Sin embargo, esta correlación no revela información nueva ya que es esperado que entre más celdas se tengan, mayor sea la cantidad de alambrado necesario para poder conectarlas. Otras posibles variables predictoras no presentan factores de correlación altos con las variables de QoR incluidas en este análisis. En este caso necesitamos buscar otras variables predictoras y rehacer el análisis para buscar patrones de interés.

Las recomendaciones hechas luego de completar los diferentes análisis realizados en esta investigación incluyeron:

- Recomendaciones de cambios en la ubicación de las unidades dentro de una partición para mejorar el rendimiento en las medidas de temporización del diseño.

- Recomendaciones relacionadas a la cantidad máxima de celdas por partición con el objetivo de reducir la complejidad de las particiones y de esta forma lograr

disminuir los tiempos de ejecución de las herramientas CAD. El resultado final es una reducción en el tiempo total de diseño de un circuito integrado VLSI.

- Recomendaciones de datos adicionales para ser incluidos en MDP de tal forma que se puedan encontrar mejores predictores para la tasa de crecimiento de celdas y las variables de QoR.

Se puede observar que estas recomendaciones tienen un impacto real y positivo en el proceso de diseño de circuitos integrados VLSI, que va desde reducciones en el tiempo requerido para diseñar un producto hasta mejoras en las características de rendimiento del producto. Siguiendo un procedimiento similar al seguido en los cuatro análisis realizados como parte de este trabajo, es posible encontrar otras relaciones de interés que mejoren el proceso de diseño. Los beneficios de implementar técnicas de minería de datos en el entorno de diseño de circuitos integrados VLSI son claros.

Capítulo 6: Conclusiones y Recomendaciones

6.1 Conclusiones

En la actualidad, la cantidad de datos que se generan y almacenan diariamente, supera la capacidad que tenemos para analizarlos. Se dice que en los últimos dos años se ha generado un 90% de todos los datos del mundo (SINTEF, 2013) y esta tendencia por generar más y más datos continúa. Existen muchas áreas del quehacer humano (*entornos* en este documento) que no tienen sistemas de minería de datos que les permitan sacar provecho de los datos que poseen.

En este trabajo se desarrolló una metodología para diseñar e implementar un sistema de minería de datos para un entorno específico. Esta metodología fue validada en el entorno de diseño de circuitos VLSI, que por su complejidad y gran volumen de datos generados, resultó ideal para demostrar la efectividad de la misma.

Los resultados obtenidos demostraron la validez de la metodología y la forma en la que un sistema de minería de datos puede ayudar a tomar mejores decisiones. Incluso, se llegó a dar recomendaciones de otros tipos de datos que deben recolectarse para poder aportar todavía más valor.

Esta metodología se desarrolló en forma genérica, de tal forma que no está atada ni a un entorno específico ni a un conjunto de herramientas específicas. Esa es la gran diferencia -y ventaja- de esta metodología con otras que puedan existir: la gran mayoría de procesos de implementación de sistemas de minería de datos han sido desarrollados por las empresas que comercializan herramientas de minería, por lo que están fuertemente ligados a esas herramientas.

La metodología desarrollada en este trabajo es independiente de las herramientas que se utilicen. Además, como parte de la metodología se desarrolló un procedimiento de comparación de herramientas, lo que facilita la selección de la que mejor llene las necesidades del entorno por analizar. Siguiendo esta metodología se desarrolla un sistema

de minería de datos en donde la base de datos, la herramienta de minería y la herramienta de visualización son seleccionadas de acuerdo a las necesidades del entorno.

Se encontró que la participación de expertos en el entorno y de científicos de datos es de suma importancia para lograr resultados exitosos. Esto por cuanto el primer paso de la metodología es realizar una caracterización del entorno, y durante este paso se necesita contar con la participación de expertos en el entorno. Luego, en las etapas posteriores del proceso de diseño, la participación de científicos de datos es requerida para desarrollar los modelos de análisis y preparar los datos por analizar. Finalmente, ambos expertos deben participar en la interpretación los resultados.

Adicionalmente, también se llegó a las siguientes conclusiones:

- El proceso de minería de datos es un proceso continuo. El completar un análisis y llegar a una conclusión no significa que esa conclusión es una verdad absoluta. Es necesario implementar un proceso de actualización, tanto para los datos como para los análisis. Cada vez que hay cambios a nivel de datos, proceso, etc., es necesario repetir el análisis para verificar que las conclusiones se mantienen.
- No todos los análisis de minería de datos van a detectar patrones de interés. Sin embargo, aún y cuando no se detecte un patrón de interés, siempre se puede llegar a conclusiones valiosas como por ejemplo: "el conjunto de variables predictoras usadas para este análisis no contiene ninguna variable que pueda predecir la variable de interés, razón por la cual es necesario buscar otras variables predictoras".
- Es preferible utilizar herramientas de minería de datos disponibles, que desarrollar una herramienta desde cero.
- Para proliferar el uso de un sistema de minería de datos, es necesario capacitar a los usuarios en las tareas de minería y análisis de datos.

En resumen, la metodología de desarrollo de sistemas de minería de datos para entornos específicos demostró su efectividad al ser validada en el entorno de diseño de circuitos

VLSI. Adicionalmente, es importante subrayar que la participación de expertos tanto en el entorno como en el análisis de datos es necesaria para que esta metodología sea exitosa.

6.2 Recomendaciones

A continuación se van a dar una serie de recomendaciones para facilitar la adopción del sistema de minería de datos desarrollado.

6.2.1 Recomendaciones Generales

- Los usuarios deben recibir un *entrenamiento básico de minería de datos*. Este entrenamiento debe incluir, como mínimo, los siguientes tópicos:
    - Definición de minería de datos.
    - Tipos de datos (i.e. continuos, nominales).
    - Explicación de las tareas principales de minería de datos.
    - Preparación de datos (ex. Remover valores extremos, manejo de datos faltantes).
    - Uso básico de la herramienta de minado de datos.
- Se debe crear una tabla donde se explique el *uso correcto* de las tareas de minado de datos dependiendo del tipo de variables que se tengan y de las características de los datos (como por ejemplo si tienen datos faltantes). Esta tabla es especialmente importante si se utilizan herramientas como RapidMiner, ya que una herramienta como esta tiene más de 500 operadores. Sería irreal esperar que los usuarios lleguen a conocer todos los operadores y sepan cuál se debe usar en qué condición.
- La arquitectura del sistema se tiene que modificar para que la herramienta de minería de datos pueda *accesar archivos de texto*, como por ejemplo archivos .csv. Esto es importante ya que es altamente probable que no todos los datos de interés se encuentren en la base de datos, por lo que los usuarios tienen que tener la posibilidad de cargar datos adicionales en la herramienta. De esta forma, la arquitectura recomendada para el sistema de minado de datos es la que se muestra en la Figura 33.

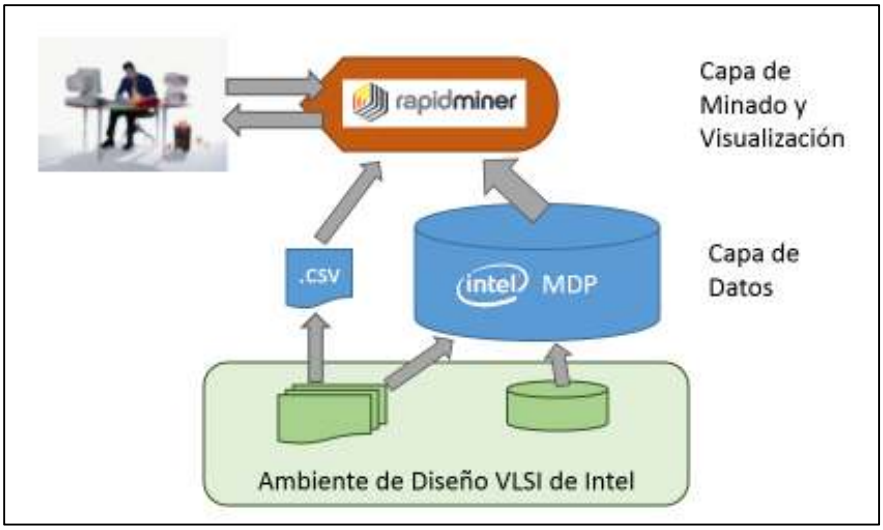

Figura 33: Arquitectura recomendada para el Sistema de Minado de Datos

- Se debe generar un documento de *casos de estudio*, y el mismo debe estar disponible para los usuarios. Este documento le enseñaría a los usuarios ejemplos reales de análisis hechos en el entorno de su interés. El hecho de que sean ejemplos reales facilita significativamente el proceso de aprendizaje para los usuarios.
- Entre la población de expertos en el entorno, se debe tener por lo menos una persona que también sea experta en técnicas de minado y análisis de datos. Esta persona con *conocimiento experto en ambas áreas* va a ser de mucho valor en la identificación de análisis importantes y en la correcta interpretación de los resultados.

6.2.2 Recomendaciones para la Base de Datos

- Es importante tener un *diccionario de datos*. Esto es especialmente importante para bases de datos "schema-less" (como por ejemplo MDP), ya que es muy fácil que la variedad de llaves en los pares llave-valor llegue a ser inmanejable y sea sumamente difícil para los usuarios determinar cuál es el par llave-valor que deben utilizar.
- Se deben implementar procesos automáticos que verifiquen la integridad de los datos y hagan limpieza y transformación de los mismos. Un ejemplo de una herramienta que se podría evaluar es Potter's Wheel A-B-C (RAMAN, s.f.).

- En el caso específico de MDP, es necesario tener más datos a nivel de RTL. Estos datos pueden ser mejores predictores que los datos de síntesis usados en esta investigación.

6.2.3 Recomendaciones para la Herramienta de Minería de Datos

- Se tiene que documentar el proceso de instalación y configuración de la herramienta, para que los nuevos usuarios no tengan problemas. En el caso específico de este trabajo, se tuvieron problemas al inicio para lograr conectar RapidMiner a MDP. Estos problemas se lograron resolver al modificar la configuración de RapidMiner. Sin embargo, un usuario nuevo probablemente se vería desmotivado si se enfrenta con este tipo de problemas, lo que podría poner en riesgo la implementación del sistema.
- Si se llega a un punto donde el rendimiento de la herramienta no es bueno, se pueden tratar de implementar algoritmos de reducción o transformación de datos para reducir su tamaño y así mejorar el rendimiento. Se puede buscar más información al respecto en (HAN & KAMBER, 2006, pág. 101).

Referencias Bibliográficas

Advantages and Disadvantages of Data Mining. (s.f.). Recuperado el 29 de August de 2014, de Zentut: http://www.zentut.com/data-mining/advantages-and-disadvantages-of-data-mining/

AKTHAR, F., & HAHNE, C. (2012). *RapidMiner 5 Operator Reference.* Rapid-I. Obtenido de http://rapidminer.com/documentation/

BECK, K., & ANDRES, C. (2004). *Extreme Programming Explained* (2nd ed.). Boston, Massachusetts, USA: Addison-Wesley.

CDH. (s.f.). Recuperado el 20 de Setiembre de 2014, de Cloudera: http://www.cloudera.com/content/cloudera/en/products-and-services/cdh.html

Colaboradores de Wikipedia. (s.f.). *Almacen de datos*. Recuperado el 9 de Setiembre de 2014, de Wikipedia, La enciclopedia libre: http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos

Colaboradores de Wikipedia. (s.f.). *Application specific integrated circuits*. Recuperado el 10 de agosto de 2013, de Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/ASIC

Colaboradores de Wikipedia. (s.f.). *Bases de Datos Relacionales*. Recuperado el 5 de Setiembre de 2014, de Wikipedia, La enciclopedia libre: http://es.wikipedia.org/wiki/Base_de_datos_relacional

Colaboradores de Wikipedia. (s.f.). *Big Data*. Recuperado el 20 de Setiembre de 2014, de Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Big_data

Colaboradores de Wikipedia. (s.f.). *Cloudera*. Recuperado el 20 de Setiembre de 2014, de Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Cloudera

Colaboradores de Wikipedia. (s.f.). *Fabricación de circuitos integrados*. Recuperado el 20 de agosto de 2013, de Wikipedia, La enciclopedia libre: http://es.wikipedia.org/wiki/Fabricaci%C3%B3n_de_circuitos_integrados

Colaboradores de Wikipedia. (s.f.). *Fotolitografia*. Recuperado el 20 de agosto de 2013, de Wikipedia, La enciclopedia libre: http://es.wikipedia.org/wiki/Fotolitograf%C3%ADa

Colaboradores de Wikipedia. (s.f.). *Haswell (microarchitecture)*. Recuperado el 24 de August de 2014, de Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Haswell_(microarchitecture)

Colaboradores de Wikipedia. (s.f.). *Integrated Circuits*. Recuperado el 17 de agosto de 2013, de Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Integrated_circuits

Colaboradores de Wikipedia. (s.f.). *Moore's Law*. Recuperado el 7 de Octubre de 2014, de Wikipedia: http://en.wikipedia.org/wiki/Moore's_law

Colaboradores de Wikipedia. (s.f.). *NoSQL*. Recuperado el 5 de Setiembre de 2014, de Wikipedia, La enciclopedia libre: http://es.wikipedia.org/wiki/NoSQL

Colaboradores de Wikipedia. (s.f.). *OLAP*. Recuperado el 5 de Setiembre de 2014, de Wikipedia, La enciclopedia libre: http://es.wikipedia.org/wiki/OLAP

Colaboradores de Wikipedia. (s.f.). *Transistor*. Recuperado el 12 de Octubre de 2014, de Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Transistor

Colaboradores de Wikipedia. (s.f.). *Weka (machine learning)*. Recuperado el 19 de Setiembre de 2014, de Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Weka_(machine_learning)

DENNING, S. (s.f.). *What Exactly is Agile? Is Kanban Agile?* Recuperado el 19 de Setiembre de 2014, de Forbes.com: http://www.forbes.com/sites/stevedenning/2012/09/25/what-exactly-is-agile-is-kanban-agile/

GOEBEL, M., & GRUENWALD, L. (1999). A survey of data mining and knowledge discovery software tools. *ACM SIGKDD (Association for Computing Machinery's Special Interest Group on Knowedge Discovery and Data Mining)*, *1*, págs. 20-33.

GONZALEZ, C. A. (2012). Curso MC-6007 Bases de Datos Avanzadas. Cartago, Costa Rica: Programa de Maestria en Ciencias de la Computacion, Instituto Tecnologico de Costa Rica.

HAN, J. (1999). Data Mining. En J. URBAN, & P. DASGUPTA (Edits.), *Encyclopedia of Distributed Computing*. Kluwer Academic Publishers.

HAN, J., & KAMBER, M. (2006). *Data Mining Concepts and Techniques* (2 ed.). San Francisco, CA: Elsevier Inc.

HERNANDEZ-CASTRO, F. (2012). MC8832 - Curso de Visualización de la Información. Cartago, Costa Rica: Programa de Maestria en Ciencias de la Computación, Instituto Tecnologico de Costa Rica.

IlliMine, E. d. (s.f.). *IlliMine*. Recuperado el 20 de Setiembre de 2014, de IlliMine: http://illimine.cs.uiuc.edu/

INMON, W. H. (2005). *Building the Data Warehoulse* (1 ed.). Indiana: Wiley Publishing Inc.

INMON, W. H., STRAUSS, D., & NEUSHLOSS, G. (2008). *DW2.0 The Architecture for the Next Generation of Data Warehousing*. USA: Elsevier Science.

KINBALL, R., & ROSS, M. (2002). *The Data Warehouse Toolkit: The complete guide to dimensional modeling* (2 ed.). USA: Wiley Computer Publishing.

KUSIAK, A., KERNSTINE, K., KERN, J., McLAUGHLIN, K., & TSENG, T. (2000). Data Mining: Medical and Engineering Case Studies. *Proceedings of the Industrial Engineering Research Conference.* Cleveland, Ohio. Obtenido de http://user.engineering.uiowa.edu/~ankusiak/Res-in-Prog/IIE-0.pdf

LEE, J. H., YU, S. Y., & PARK, S. C. (2001). Design of intelligent data sampling methodology based on data mining. *IEEE Transactions of Robotics and Automation, 17*(5), 637-649. Recuperado el 10 de junio de 2013, de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=964664&tag=1

LIU, H., SINGHEE, A., RUTENBAR, R. A., & CARLEY, L. R. (2002). Remembrance of Circuits Past: Macromodeling by Data Mining in Large Analog Design Spaces. *IEEE Design Automation Conference*, (págs. 437-442). Recuperado el 10 de junio de 2013, de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1012665&tag=1

McFARLAND, G. (2006). *Microprocessor Design: A Practical Guide from Design Planning to Manufacturing*. USA: McGraw-Hill Professional Publishing.

MEINEL, C., & THEOBALD, T. (1998). *Algorithms and Data Structures in VLSI Design. OBDD - Foundations and Applications*. Berlin: Springer-Verlag.

MISHRA, P., PADHY, N., & PANIGRAHI, R. (2012). The survey of data mining applications and feature scope. En G. I. Technology (Ed.), *Asian Journal of Computer Science and Information Technology*, *4*, págs. 68-77. Gunupur, India. Obtenido de http://www.innovativejournal.in/index.php/ajcsit

NORTH, D. M. (2012). *Data Mining for the Masses.* Global Text. Obtenido de http://rapidminer.com/documentation/

PATEL, T., & THOMPSON, W. (s.f.). *Data Mining from A to Z.* Recuperado el 29 de August de 2014, de SAS: http://www.sas.com/en_us/offers/sem/data-mining-2273479/register.html?gclid=CI2HnKnKucACFQ-DfgodmY4AVg

PATTON, J. (s.f.). *Kanban Development Oversimplified*. Recuperado el 19 de Setiembre de 2014, de AgileProductDesign.com: http://agileproductdesign.com/blog/2009/kanban_over_simplified.html

RAKOTOMALALA, R. (s.f.). *TANAGRA*. Recuperado el 19 de Setiembre de 2014, de Université Lyon 2: http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

RAMAN, V. (s.f.). *Potter's Wheel A-B-C: An Interactive Tool for Data Analysis, Cleansing, and Transformation*. Recuperado el 12 de Setiembre de 2014, de Berkeley, University of California: http://control.cs.berkeley.edu/abc/

Rapid-I GmbH. (2010). *RapidMiner 5.0 Manual.* Rapid-I. Obtenido de http://rapidminer.com/documentation/

Rapid-I GmbH. (s.f.). *RapidMiner 5.2 Advanced Charts.* Rapid-I. Obtenido de http://rapidminer.com/documentation/

RapidMiner Resources. (25 de Mayo de 2014). Aggregate. *RapidMiner YouTube Channel*. Recuperado el 16 de Octubre de 2014, de http://www.youtube.com/watch?v=Pao5GSohrK8&list=PLADDF95B445B2E26F&index=2

RapidMiner Resources. (25 de Mayo de 2014). RapidMiner GUI Overview. Recuperado el 19 de Octubre de 2014, de http://www.youtube.com/watch?v=YTMwhK705QA&list=PLADDF95B445B2E26F

SAHOTA, M. (2012). *An Agile Adoption and Transformation Survival Guide: Working with Organizational Culture.*

SINTEF. (22 de mayo de 2013). *Big Data, for better or worse: 90% of world's data generated over last two years*. Recuperado el 3 de noviembre de 2014, de Sciencedaily: http://www.sciencedaily.com/releases/2013/05/130522085217.htm

Weka 3. (s.f.). Recuperado el 19 de Setiembre de 2014, de University of Waikato: http://www.cs.waikato.ac.nz/ml/weka/index.html

Wikipedia, C. d. (s.f.). *Cross Industry Standard Process for Data Mining*. Recuperado el 15 de octubre de 2014, de Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining

XU, J., LIU, B., & XU, R. (s.f.). Design and Implementation of Data Mining based IDS. *Computing Center, Institute of High Energy Physics, CAS.* Beijing. Recuperado el 10 de junio de 2013, de http://en.cnki.com.cn/Article_en/CJFDTOTAL-JSJC200206007.htm

Apéndices

Apéndice 1: Tipos de Bases de Datos Comunes

Base de Datos Relacional

Consiste de un conjunto de tablas que pueden estar relacionadas entre sí por medio de elementos comunes (de ahí su nombre). Cada tabla tiene un nombre único y está compuesta por una o varias columnas y una o varias filas. Los datos guardados en una fila se conocen como una tupla. Cada tupla puede ser identificada por un llave única (donde la llave puede estar formada por los elementos de una o varias columnas).

Los datos en una base de datos relacional pueden ser extraídos por medio de búsquedas de datos, implementadas por medio de lenguajes relacionales, como por ejemplo SQL.

Data Warehouse

Es un repositorio de información de diferentes fuentes, cuyos datos están relacionados por medio de un entorno común. Para facilitar el proceso de toma de decisiones, los datos en un data warehouse están organizados por temas comunes, como por ejemplo clientes, suplidores, artículos, ventas, etc. Los datos generalmente son almacenados por largos periodos de tiempo (múltiples años), y se almacenan en forma resumida (por ejemplo, en lugar de guardar los detalles de cada venta, se resumen las ventas por día o por semana.

Base de Datos Transaccional

Es un tipo de base de datos donde cada record representa una transacción. Generalmente, cada transacción se puede identificar por medio de un número de identificación único.

Base de Datos NoSQL

Bases de datos que difieren del modelo clásico de las bases de datos relacionales, en las cuales los datos no se encuentran almacenados en estructuras fijas como tablas y no se garantiza ACID (Atomicidad, Consistencia, Aislamiento (Isolation en inglés) y Durabilidad). Una implementación popular de este tipo de bases de datos es la conocida como key-value, donde los datos se almacén de acuerdo a pares clave-valor. Este tipo de bases de datos están altamente optimizadas para las tareas de recuperar y agregar, y se

caracterizan por tener una muy buena escalabilidad y rendimiento en estructuras de datos altamente horizontales (Colaboradores de Wikipedia, NoSQL, s.f.).

Apéndice 2: Aplicaciones de la Minería de Datos

La minería de datos todavía puede ser considerada como un área en desarrollo. Existen múltiples implementaciones de minería de datos que han sido optimizadas para entornos y/o industrias específicas. A continuación se van a mencionar algunas de estas implementaciones con ejemplos del tipo de análisis que se puede realizar (HAN & KAMBER, 2006).

- **Entornos financieros**: predicción de pago de préstamos, clasificación de clientes en base al riesgo, clasificación de clientes para actividades de mercadeo, detección de fraudes y otros crímenes financieros.
- **Industria de ventas de productos**: análisis multidimensional de clientes, ventas, productos, tiempo y regiones. Análisis de efectividad de campañas de mercadeo, análisis de lealtad y retención de los clientes, predicción de compras de productos para hacer recomendaciones a los clientes.
- **Industria de telecomunicaciones**: identificación de patrones fraudulentos, predicción de servicios que un cliente específico puede solicitar, técnicas de visualización para para el análisis de los datos.
- **Análisis de datos biológicos**: integración semántica de bases de datos genómicas. Alineación, indexación y búsqueda de similitudes en secuencias de ADN y proteínicas. Análisis de enfermedades.

Como se puede observar, las aplicaciones de la minería de datos son muy variadas y pueden incluir una gran variedad de áreas de interés para el ser humano. Esta investigación se enfoca en encontrar aplicaciones de minería de datos para el desarrollo estructurado de circuitos integrados VLSI.

Apéndice 3: Videos Tutoriales de Minería de Datos

En la Tabla 23 se presentan los detalles de videos cortos que pueden ser utilizados para explicar los conceptos básicos y beneficios de la minería de datos.

Descripción	Idioma	Duración	URL
Visión general de minería de datos dada por un profesor de administración de negocios	Inglés	3:22	http://youtu.be/R-sGvh6tI04
Explicación (cómica) de como la minería de datos puede ayudar a los comercios a aumentar sus ventas	Inglés	5:42	http://youtu.be/f2Kji24833Y
Uso de la minería de datos en las empresas	Español	10:56	http://youtu.be/P5oxLVOu8qU
Explicación más técnica de los conceptos de minería de datos	Español	11:51	http://youtu.be/mx3vv--tn9o
Minería de datos para mercadeo y ventas	Español	12:35	http://youtu.be/LfauNEEAt1Y

Tabla 23: Videos instructivos de minería de datos

Apéndice 4: Herramientas de Minería de Datos

A continuación se presenta información básica de las herramientas de minería de datos que fueron analizadas.

Weka

Weka (Waikato Environment for Knowledge Analysis, ambiente Waikato para el análisis de conocimiento por sus siglas en inglés) es una aplicación de minería de datos desarrollada en la Universidad de Waikato, en Nueva Zelandia (Colaboradores de Wikipedia, Weka (machine learning), s.f.) (Weka 3, s.f.).

Weka puede descargarse de http://www.cs.waikato.ac.nz/ml/index.html

TANAGRA

TANAGRA es una aplicación libre de minería de datos para aplicaciones académicas y de investigación. TANAGRA es un software de código abierto. El objetivo principal de TANAGRA es el de darle a los estudiantes e investigadores una aplicación de minería de datos sencilla de usar (RAKOTOMALALA, s.f.).

TANAGRA puede descargarse de http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html

RapidMiner CE

RapidMiner es una aplicación comercial que soporta minado de datos, minado de texto, análisis predictivos, inteligencia de negocio y aprendizaje de máquina.

Adicional a la versión comercial, también existe una edición para la comunidad de código abierto de RapidMiner. Esta versión es de descarga libre y está enfocada a desarrolladores e investigadores.

RapidMiner CE (Community Edition, edición comunitaria por sus siglas en inglés) se puede descargar de http://community.rapidminer.com/

Cloudera CDH

Cloudera CDH (Cloudera Distribution including Apache Hadoop, distribución de Cloudera que incluye Apache Hadoop por sus siglas en inglés) es la aplicación de código abierto de Cloudera (Colaboradores de Wikipedia, Cloudera, s.f.). Es básicamente el sistema para analizar datos que se encuentran almacenados en una base de datos Apache Hadoop (CDH, s.f.).

Cloudera CDH puede descargarse de
http://www.cloudera.com/content/cloudera/en/products-and-services/cdh.html

Apéndice 5: Uso Básico de RapidMiner

En este apéndice se va a presentar un resumen breve de las funcionalidades más comunes de RapidMiner. La información presentada se obtuvo del libro (NORTH, 2012).

Antes de iniciar, se recomienda ver el video (RapidMiner Resources, RapidMiner GUI Overview, 2014) para familiarizarse con la interfaz de usuario.

En la Tabla 24 se puede ver una lista de los operadores de RapidMiner más comunes y el tipo de análisis de minería de datos para el que se pueden utilizar.

Tarea del Minado de Datos	Análisis	Capítulo	Operador de RapidMiner	Comentarios
Asociación	Matriz de correlación	4	Correlation Matrix	Genera una matriz en la que se presentan los coeficientes de correlación entre todos los datos del modelo Necesita datos enteros o reales
Asociación	Reglas de asociación	5	FP-Growth + Create Association Rules	Busca los FP (Frequent Patterns, patrones frecuentes por sus siglas en inglés) Todos los valores deben ser binominales Es un análisis de "carrito de mercado"
Agrupación	K-Means	6	K-Means	Busca los grupos en los que se pueden acomodar los datos al analizar los promedios de los atributos en un conjunto de datos. Necesita datos enteros o reales
Predicción	Discriminante	7	Linear-Discriminant Analysis	Genera un modelo para predecir la categoría en base a datos históricos. La categoría en el conjunto de datos de entrenamiento debe configurarse como "Label". Al igual que K-Means, usa datos enteros o reales

Tarea del Minado de Datos	Análisis	Capítulo	Operador de RapidMiner	Comentarios
Predicción	Regresión lineal	8	Linear Regression	Modelo predictivo que utiliza la ecuación de una recta para predecir donde una observación va a caer dentro de esa línea. Al igual que para la matriz de correlación, necesita datos enteros o reales. Es importante verificar que los datos que se van a usar como entrada para el modelo no tienen rangos mayores que los datos de entrenamiento. Por ejemplo, si tengo una variable de temperatura en el conjunto de datos de entrenamiento con valores entre 15 y 40, entonces el conjunto de datos que va a ser utilizado para predecir, no puede tener valores de temperatura que estén por fuera de ese rango. Si los tuviera, es necesario arreglar los datos (ej. eliminar los datos que están fuera de rango, cambiar el valor, etc.).
Predicción	Regresión logística	9	Logistic Regression	Se utiliza para predecir la probabilidad de que un evento ocurra. Utiliza la fórmula matemática de una función cuadrática. Los datos deben ser reales o continuos. Label tiene que ser de tipo nominal Como en el caso de la regresión lineal, el modelo predictivo solo es válido en el rango de los datos del conjunto de entrenamiento.
Agrupación	Árboles de decisión	10	Decision Tree	Genera un árbol de decisión para predecir el Label. Puede manejar cualquier tipo de datos, incluso, puede manejar datos faltantes. Si el tipo de modelado se cambia a geni_index se aumenta significativamente la granularidad del modelo.

Tarea del Minado de Datos	Análisis	Capítulo	Operador de RapidMiner	Comentarios
Predicción	Redes neuronales	11	Neural Net	Similar a el árbol de decisiones, pero es mejor para detectar relaciones ente variables predictoras. Permite predecir valores aun y cuando el rango sea mayor al rango del conjunto de entrenamiento. Esto lo logra ya que utiliza algoritmos de lógica difusa.

Tabla 24: Comandos comunes de RapidMiner

Otros operadores de uso frecuente en RapidMiner incluyen:

- **Aggregate**: Se puede utilizar para generar resúmenes. Esto es, puede calcular sumas, promedios, etc. Con este indicador se pueden hacer análisis muy similares a los que se hacen con las tablas dinámicas de Microsoft Excel. Por la gran utilidad que tiene este operador, se recomienda ver el siguiente video (RapidMiner Resources, Aggregate, 2014).
- **Apply Model**: Se utiliza para aplicar un modelo predictivo en un conjunto de datos. Es necesario que la variable que se desea predecir en el conjunto de datos de entrenamiento tenga el rol de "label".
- **Filter examples**: Se pueden utilizar para remover datos que no cumplan una regla específica (ej. decisión_number>2)
- **Multiply**: se utiliza para duplicar un dato, de tal forma que pueda ser utilizado por dos procesos diferentes luego de que el dato sea "multiplicado"
- **Replace**: Se utiliza para cambiar el valor de un atributo específico.
- **Replace Missing**: Se utiliza para arreglar un conjunto de datos que tiene datos faltantes.
- **Select Attribute**: Se utiliza para obtener un subset del conjunto de datos.
- **Set Role**: Se utiliza para cambiar el rol de una variable. Label es utilizado por los modelos predictivos, mientras que id se usa para mantener variables que no van a ser usadas como predictores.

- **X-Validation**: Se utiliza para validar la exactitud de un modelo predictivo, ayudando a calcular el porcentajes de fallas de predicción (conocidos como falsos positivos). En la Figura 34 se muestra el flujo recomendado para generar un modelo predictivo y al mismo tiempo poder validar la efectividad del mismo. En este flujo, el operador de X-Validation es un operador de tipo sub-proceso. En la Figura 35 se muestran los dos subprocesos que componen este operador: Training y Testing.

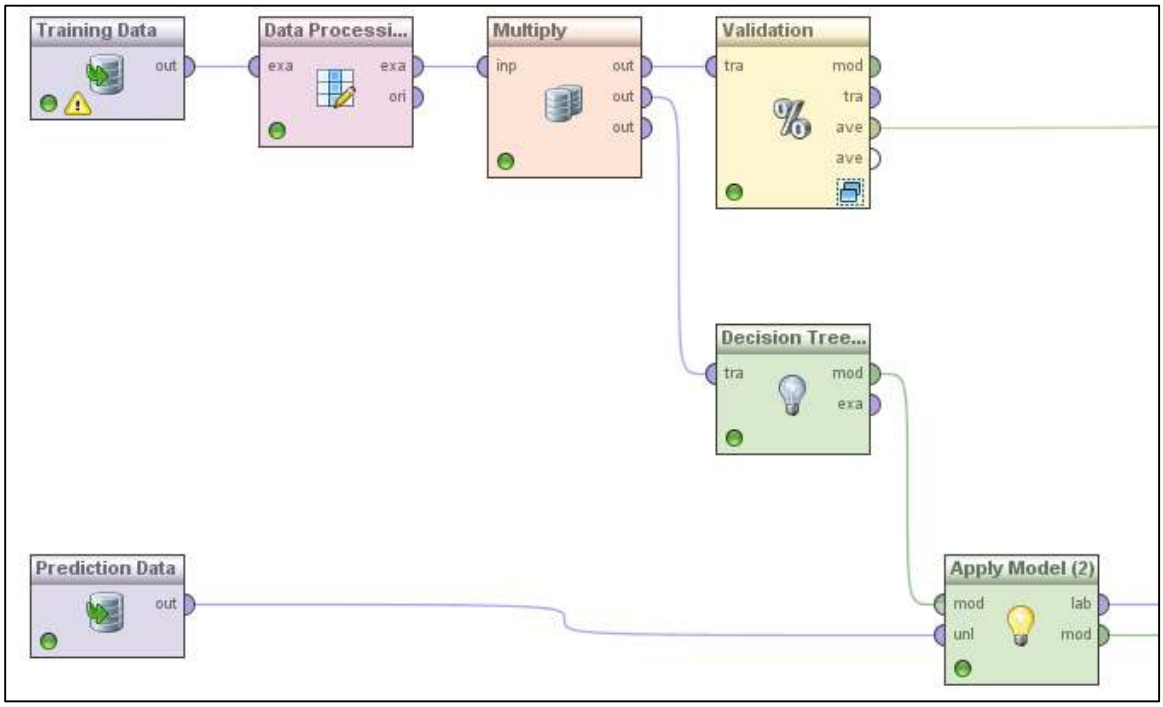

Figura 34: Flujo recomendado de RapidMiner para modelos predictivos

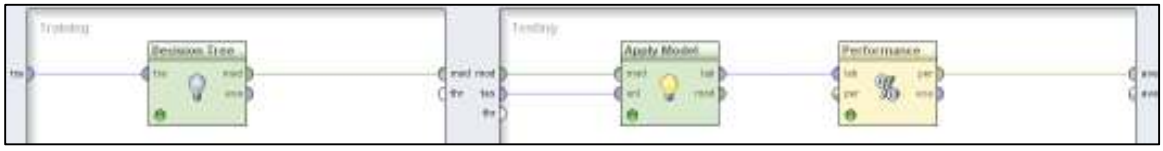

Figura 35: Subprocesos del operador X-Validation

Apéndice 6: Glosario de Acrónimos

Acrónimo	Significado
CAD	Computer Aided Design, diseño asistido por computadoras por sus siglas en inglés. Se refiere a cualquier programa de computación que se usa para facilitar el proceso de diseño.
CRISP-DM	CRoss-Industry Standard Process for Data Mining, proceso de minado de datos estándar para múltiples industrias por sus siglas en inglés, CRISP-DM es una metodología estándar de minería de datos que pueda ser utilizada independientemente de las herramientas que se usen.
DRV	Design Rule Violation, violación de una regla de diseño por sus siglas en inglés. Es cuando no se logra cumplir con una regla de diseño del circuito integrado. Estas reglas de diseño generalmente tienen una relación directa con la manufacturabilidad del producto.
GPU	Graphical Processing Unit, unidad de procesamiento gráfico por sus siglas en inglés. Es el bloque lógico encargado de procesar las imágenes que se despliegan en pantalla.
MDP	Metrics for Design Process, métricas del proceso de diseño por sus siglas en inglés. Es una base de datos utilizada en el entorno de diseño VLSI de Intel.
OLAP	Online Analytical Processing, procesamiento analítico en línea por sus siglas en inglés. Es una técnica de análisis de datos que utiliza cubos para generar diferentes perspectivas y niveles de abstracción de los datos.
QoR	Quality of Results, resultados de calidad por sus siglas en inglés. Representa un grupo de indicadores que se utilizan para medir la calidad de un diseño.

RTL	Register Transfer Level, nivel de transferencia de registros por sus siglas en inglés. Modelo funcional de un circuito integrado que puede usare para validar la funcionalidad del mismo por medio de simulaciones.
TNS	Total Negative Slack, margen negativo total por sus siglas en inglés. Es la suma total de los márgenes negativos de todos los circuitos lógicos que no cumplen el requerimiento de tiempo. El margen negativo es la cantidad de tiempo por la que no se está cumpliendo el requerimiento de tiempo
VLSI	Very Large Scale Integration, integración a muy gran escala por sus siglas en inglés. Se refiere a circuitos integrados que llegan a tener miles de millones de transistores

Tabla 25: Acrónimos