**Sebastian Arango**

**Steven Ma Mei**

**Daniel Galvis**

**Luis Fernando Muñoz**

**Fase 1: Definición del problema**

Síntomas y necesidades:

-La empresa Allers ha estado distribuyendo medicamentos e insumos hospitalarios desde hace 62 años.

-La empresa Allers hace 15 años implementó un software de gestión empresarial llamado SAP Bussines One, desde ese entonces, han estado recolectando información de miles de transacciones mensuales que la empresa maneja.

-La empresa Allers cuenta con información de los clientes, productos y proveedores en el software SAP.

-La empresa Allers busca un sistema de marketing personalizado, donde se pueda saber lo que necesitan sus clientes con precisión para que posteriormente se les haga ofertas de productos con base a estas necesidades.

-La empresa Allers cree que algunos clientes pueden comprar algunos productos que actualmente no compran.

Definición del problema:

La empresa Allers necesita una solución tecnológica que permita el procesamiento de los datos recolectados en sus transacciones, para poder predecir el comportamiento de compra de sus clientes con el objetivo de mejorar las utilidades.

Requerimientos funcionales:

1. Introducción

1.1. Definiciones, Acrónimos y abreviaturas.

Transacciones: Es un acuerdo, comunicación o movimiento llevado a cabo entre un comprador y un vendedor en la que se intercambian un activo contra un pago.

En el contexto del problema las transacciones vendrán representadas por un ID (Número de identificación único) y un conjunto de atributos que se organizan en columnas (identificación del producto vendido, cantidad, etcétera).

SAP: Es el acrónimo utilizado en el nombre de la compañía de software que distribuye SAP Business One. Systems, Applications & Products in Data Processing.

Datos: En el contexto del problema, los datos toman la forma de miles de transacciones.

2. Funciones del sistema:

Con base a lo discutido en la reunión con nuestro cliente, a grandes rasgos se pudo extraer estas dos funcionalidades:

1. Esta primera funcionalidad tiene como propósito predecir y ofrecerle al cliente los productos que él pueda necesitar antes de que él haga cualquier pedido. La predicción se hará con base a transacciones, desde la más reciente hasta la más antigua, que él y otros clientes similares han tenido.
2. Esta segunda funcionalidad es orientada para que, cuando un cliente haga un pedido, se le ofrezcan otros productos. La predicción se hará con base a la probabilidad de que el cliente pueda comprar otros productos dado los productos que demandó en el actual pedido.

3. Requerimientos específicos

|  |  |
| --- | --- |
| **Identificación del requerimiento** | 1 |
| **Nombre del requerimiento** | Cargar datos de transacciones |
| **Características** | El programa es capaz de cargar los datos estructurados sobre las transacciones con los diferentes clientes que el cliente desee subir. |
| **Descripción del requerimiento** | El usuario está en la capacidad de subir un nuevo archivo con datos actualizados sobre las transacciones, a lo que el programa los guardará y empezará a ejercer sus funcionalidades con dichos datos |
| **Prioridad del requerimiento:**  Media | |

|  |  |
| --- | --- |
| **Identificación del requerimiento** | 2 |
| **Nombre del requerimiento** | Recomendar productos antes de comprar |
| **Características** | El programa utilizará la técnica del Data Mining llamada clustering para, a partir de los datos cargados sobre transacciones, hacer una clasificación de compañías similares según su historial de ventas. |
| **Descripción del requerimiento** | Al utilizar clustering, se detectarán compañías similares según sus compras generales. Así, desde ANTES que un cliente vaya a hacer una compra, se le recomendaran productos que compran clientes similares. Lo anterior se hará implementando algún algoritmo de Clustering. |
| **Prioridad del requerimiento:**  Media | |

|  |  |
| --- | --- |
| **Identificación del requerimiento** | 3 |
| **Nombre del requerimiento** | Recomendar productos después de comprar |
| **Características** | El programa utilizará la técnica del Data Mining llamada asociación para, a partir de los datos cargados sobre transacciones, recomendar productos de interés después de haber realizado una compra. |
| **Descripción del requerimiento** | Al utilizar la asociación, se detectarán que productos están relacionados. Así, DESPUÉS del cliente comprar uno o varios productos, se le recomendarán productos relacionados con los ya adquiridos. Lo anterior se hará implementando algún algoritmo de asociación |
| **Prioridad del requerimiento:**  Alta | |

**Fase 2: Recopilación de la información**

*Marco teórico:*

Se describe a continuación los conceptos que se utilizarán como base durante la construcción del aplicativo.

**Desarrollo orientado a objetos**

En el desarrollo orientado a objetos los ingenieros de sistemas modelan las situaciones y/o problemas mediante la utilización de abstracciones llamadas clases y objetos que condensan las relaciones e interacciones que existen entre las entidades involucradas en el problema a analizar.

**Desarrollo con el modelo Mundo-VistaControl**

En el desarrollo con el modelo Mundo-VistaControl las clases de las que se compone el aplicativo se dividen en dos paquetes:

-**Modelo:** En este paquete se modela la lógica del problema.

-**Vista:** En este paquete se modelan las clases que controlan el despliegue de la información y las interacciones con el usuario.

La anterior separación es resultado de aplicar un patrón de arquitectura de software, que separa los datos y la lógica de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Esto se hace con el objetivo de desarrollar software de alta cohesión y bajo acoplamiento, lo que permite en un futuro añadir más módulos y mantener el software sin grandes complicaciones.

**Principios de diseño SOLID[[1]](#footnote-1)**

SOLID es un acrónimo que representa cinco principios básicos de la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) y el [diseño](https://es.wikipedia.org/wiki/Dise%C3%B1o_orientado_a_objetos), donde cada sigla representa:

|  |  |
| --- | --- |
| S | **(*Single responsibility principle*)**  la noción de que un objeto solo debería tener una única responsabilidad. |
| O | **(*Open/closed principle*)**  la noción de que las “entidades de software … deben estar abiertas para su extensión, pero cerradas para su modificación”. |
| L | **(*Liskov substitution principle*)**  la noción de que los “objetos de un programa deberían ser reemplazables por instancias de sus subtipos sin alterar el correcto funcionamiento del programa”. Ver también diseño por contrato. |
| I | **(*Interface segregation principle*)**  la noción de que “muchas interfaces cliente específicas son mejores que una interfaz de propósito general”. |
| D | **(*Dependency inversion principle*)**  la noción de que se debe “depender de abstracciones, no depender de implementaciones”.  La inyección de dependencias es uno de los métodos que siguen este principio. |

Los principios SOLID son guías que pueden ser aplicadas en el desarrollo de software para eliminar [código sucio](https://es.wikipedia.org/w/index.php?title=Code_smell&action=edit&redlink=1) provocando que el programador tenga que [refactorizar](https://es.wikipedia.org/wiki/Refactorizaci%C3%B3n) el [código fuente](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente) hasta que sea legible y extensible.

**Desarrollo guiado por pruebas (TDD)[[2]](#footnote-2)**

Es una práctica de [ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software) que involucra otras dos prácticas: Escribir las pruebas primero (Test First Development) y [Refactorización](https://es.wikipedia.org/wiki/Refactorizaci%C3%B3n) (Refactoring). Para escribir las pruebas generalmente se utilizan las [pruebas unitarias](https://es.wikipedia.org/wiki/Prueba_unitaria) (unit test en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s)). En primer lugar, se escribe una prueba y se verifica que las pruebas fallan. A continuación, se implementa el código que hace que la prueba pase satisfactoriamente, y seguidamente se refactoriza el código escrito. El propósito del desarrollo guiado por pruebas es lograr un código limpio que funcione. La idea es que los requisitos sean traducidos a pruebas, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido.

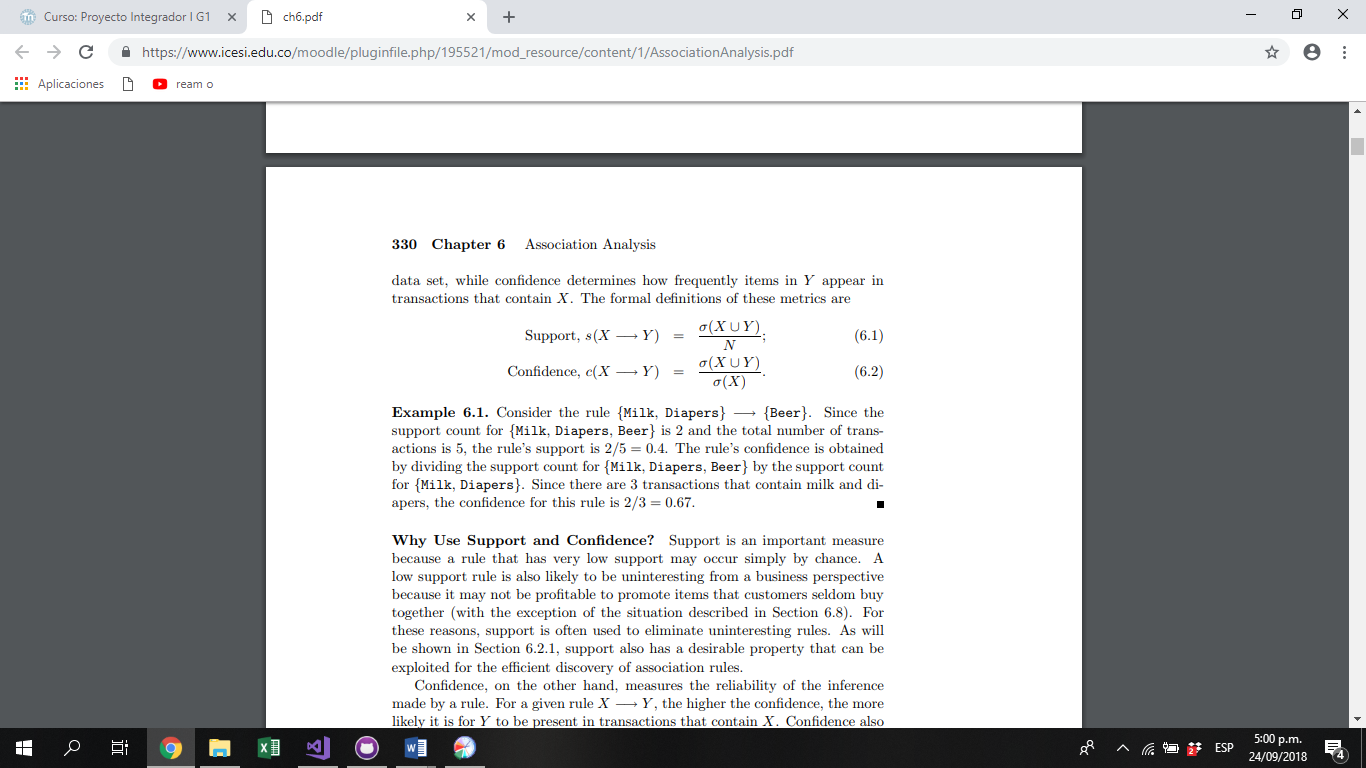
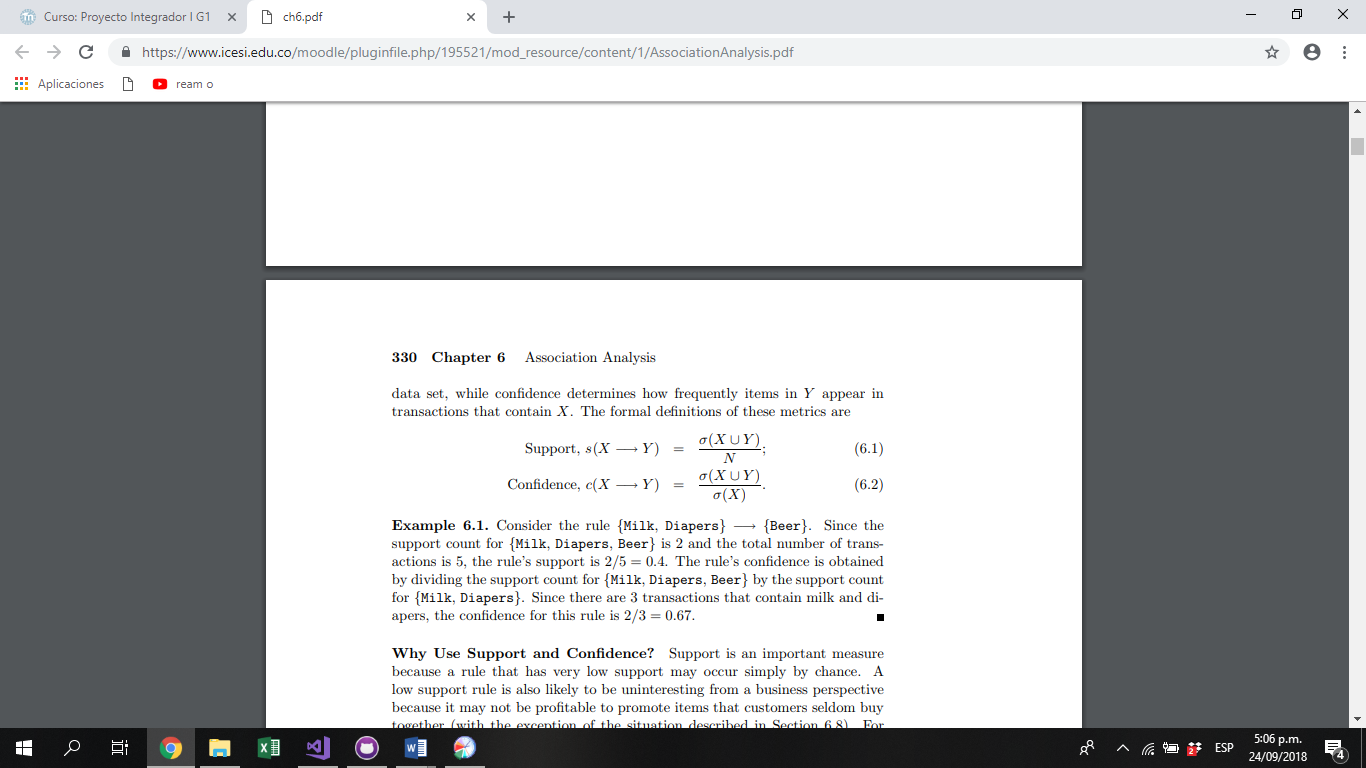
**Estado del arte**

**Mas allá del análisis de asociaciones de intratransacciones: Minando reglas de asociación multidimensionales de intertransacciones multidimensionales**

**(Beyond Intratransaction Association Analysis: Mining Multidimensional Intertransaction Association Rules)[[3]](#footnote-3)**

El papel describe un tipo de análisis de transacciones diferente al hecho normalmente, el cual es analizar reglas de asociación que emergen entre los ítems de una misma transacción, mientras que, en el papel, propone un análisis intertransaccional en vez de uno intratransaccional, donde las reglas de asociación emergen de relaciones que surgen entre las transacciones. Es importante tomar en cuenta este tipo de análisis en el desarrollo de la solución, lo cual implicaría un valor agregado en el producto final.

*Definiciones:*

1. Allers Groups: Empresa Colombiana especializada en la venta de equipos médicos, insumos hospitalarios, instrumental quirúrgico y medicamentos, tanto al por mayor como al detal.
2. Enterprise resource planning (ERP): Es un tipo de aplicación que permite la recolección de datos y todo los procesos que conllevan para generar la interpretación de esos datos[[4]](#footnote-4)
3. Data Mining: Es el proceso por el cual se intenta obtener patrones en un conjunto de datos
4. Asociación (data mining): Es una técnica de data mining en donde se utiliza funciones probabilísticas que permiten saber la ocurrencia de un evento B dado un evento A.
5. Representación binaria: Consiste en representar unas transacciones como cadenas binarias, donde 1 indica que el producto aparece en una transacción y 0 que no.
6. Support: Frecuencia en la que un itemset dado (sets de productos) aparece en las transacciones.  Association Anaylisis: Basic Concepts and Algorithms
7. Frequent Item Sets: Itemsets que aparecen frecuentemente en las transacciones. Se considera que un itemset es frecuente si su support es superior a un minsup definido.
8. Confidence: Frecuencia en la que dos itemsets X e Y aparecen en una transacción que contiene al itemset X. Association Anaylisis: Basic Concepts and Algorithms
9. Rule: Reglas de la forma X-> Y cuya confidencia es superior a un minconf. Además, XY es un Frequent Item set.
10. Principio de Apiori: Teorema que afirma que, si un itemset es frecuente, todos sus subconjuntos deben de ser frecuentes también. A partir de este principio, si se encuentra un conjunto X que no es frecuente, se puede inferir que todo conjunto que contenga a X tampoco va a ser frecuente. Dicho principio evita computar itemsets que de antemano se sabe no van a ser frecuentes.
11. Principio de monotonicidad: Dada una función f, si X es subconjunto de Y, entonces f(X) <= f(Y). El principio de Apiori se basa en la anti-monotonicidad (lo opuesto a monotonicidad).
12. Máximo frequent Item set: Un itemset frecuente para el cual ninguno de sus itemsets inmediatamente superiores es frecuente[[5]](#footnote-5).
13. Frequent itemsets cerrados: Se considera un itemset X como cerrado si es frecuente y ninguno de los conjuntos inmediatamente superiores tiene el mismo support count. Sirve para eliminar las reglas redundantes.[[6]](#footnote-6)
14. Big Data: Enfoque para describir enormes cantidades de datos (tanto estructurados, no estructurados o semi-estructurados) que tomaría demasiado tiempo cargarlos a una base de datos relacional. Es decir, es la información que no puede ser procesada con herramientas tradicionales (Por, por ejemplo, requerir una velocidad de respuesta muy rápida). Esto no quiere decir que se deban despreciar las bases de datos convencionales, sino que van de la mano, cada uno cumpliendo funciones diferentes. Existen varios tipos de Big Data, entre los que se encuentra: Web y Social media, Biométricos, Machine to Machine (M2M), Human Generated y Big Transaction Data. Actualmente, la herramienta con liderazgo en popularidad para analizar Big Data es Hadoop, el cual está inspirado en GoogleFile Systems y utiliza el paradigma MapReduce[[7]](#footnote-7).
15. Base de datos Relacional: Base de datos conformada por un conjunto de tablas, que se manipula de acuerdo al modelo de datos relacional (Donde las relaciones son tuplas, lo que se puede ver como si cada tabla fuera una relación).[[8]](#footnote-8)
16. Paradigma Map-Reduce: Framework para trabajar sobre grandes colecciones de datos en varios grupos de computadoras para aprovechar la computación paralela. Está compuesto por dos funciones: Función Map(), la cual mapea los datos de entrada de la función en paralelo y genera una lista de pares (k2,v2). La otra función es la de Reduce(), la cual se llama en paralelo para cada grupo generado en la función Map y retorna un valor final v3.[[9]](#footnote-9)
17. Business Intelligence: Conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada (interna y externa a la compañía) en información estructurada, para su explotación o para su análisis y conversión en conocimiento, dando así soporte a la toma de decisiones sobre el negocio[[10]](#footnote-10).

*Otros desarrollos:*

1. SAP Bussines one: Es un software para gestión de tipo ERP[[11]](#footnote-11) y diseñado para empresas pequeñas-mediana.
2. Hadoop: Herramienta elaborada por el grupo Apache para manipular datos distribuidos logrando un alto paralelismo en el procesamiento. Está compuesto por Hadoop Distributed Files (Los datos son divididos en bloques y distribuidos a través del Cluster), Hadoop Map-Reduce(Aplica el proceso de Map-Reduce a los datos distribuidos) y Hadoop Common(Conjunto de librerías que soportan subproyectos de Hadoop)3.
3. QlikView: Producto para elaborar aplicaciones de analítica interactivas y Dashboards. Permite a los usuarios construir aplicaciones de analítica sin necesidad de tener grandes habilidades de desarrollo, lo que permite actuar fácilmente a los cambiantes requerimientos empresariales[[12]](#footnote-12).

**Fase 3: Búsqueda de soluciones creativas**

**Ideas para Generar Frequent Itemsets**

1. Recorrido de Itemsets desde los más específicos hasta los más generales, con el objetivo de ir probando primero los itemsets más grandes. En caso de encontrar un frequent itemset, todos sus subconjuntos pasan a ser frecuentes también. (Association Analyisis: Basic Concepts and Algorithms)
2. Hacer una búsqueda en profundidad: Útil para encontrar maximal frequent itemsets. (Association Analyisis: Basic Concepts and Algorithms)
3. Encontrar los Frequent Itemsets con la ayuda de un árbol FP: FP-Growth Algorithm. Hace una representación compacta de las transacciones. (Association Analyisis: Basic Concepts and Algorithms)

**Ideas para el análisis de las transacciones de Allers**

1. Se puede desarrollar una aplicación de escritorio donde se implemente la técnica de análisis de datos de asociación a las transacciones de Allers, con el que se podrá encontrar patrones de compra de los clientes.
2. Se puede construir una aplicación con las técnicas y conceptos de la estadística descriptiva.
3. Se puede desarrollar una aplicación donde se utilice el análisis de datos multidimensional, en vez de ver las transacciones como filas con columnas como atributos, se puede transformar en un arreglo multidimensional donde ciertos atributos se discretizan.
4. Es posible desarrollar un programa que muestre vectores de tendencia con el SVM que ayuden a definir qué productos deberían salir al mercado por medio del análisis de estos, así como la identificación de problemas más frecuentes y ver qué influye en la variación de las transacciones.

**Ideas para la presentación de la información en la interfaz grafica**

1. Se puede presentar la información con la visualización de datos por medio de gráficos de barras, dispersión y tortas, apoyadas por datos numéricos representando dispersión y medidas de tendencia central.
2. Se puede presentar la información por meses, donde en cada mes se va a mostrar por cada cliente o en general las reglas extraídas de ítems comprados en ese mes.
3. Se puede presentar la información por clientes, donde se podrá observar sus compras históricas y predicciones de sus compras en un futuro mediante las reglas extraídas, además, se podrá ver los clientes similares a él en cuanto a los patrones de compra.

**Ideas para analizar a clientes**

1. Se puede desarrollar una aplicación donde se implementen una gran variedad de métricas sobre las transacciones, con las cuales se va a poder medir la similitud entre clientes, y así poder encontrar productos que unos clientes no compran pero que podrían comprar dado su similitud con otros clientes que si los compran.
2. Se puede implementar una aplicación que use algoritmos de clasificación de datos como el Naive Bayes o los Support Vector Machines, con los que se puede clasificar a clientes, pero son algoritmos supervisados, por lo que previamente se necesitaría de una tipología ya impuesta en los clientes.
3. Se puede implementar un software que analice los clientes con el método de agrupamiento K-means, algoritmo no supervisado, con esta técnica se podría agrupar a los clientes basados en sus transacciones, se espera que los clientes en un mismo grupo tuviesen un histórico de compras muy similares.

Técnicas que se utilizaron para la generación de ideas: Relación forzada y lluvia de ideas

Consiste en combinar lo conocido con lo desconocido, esto fuerza la aparición de nuevas situaciones y soluciones.

**Fase 4: Transición de la formulación de ideas a los diseños preliminares**

**Ideas descartadas:**

Para generar Frequent Itemsets:

Como en el problema se extraen relativamente pocos itemsets comparados con todos los ítems dados, entonces el recorrido deja de ser eficaz ya que se espera que haya una gran densidad de transacciones con respecto a la barrera de los frecuentes.

Para el análisis de las transacciones de Allers:

El método de Support Vector Machine utiliza la técnica del aprendizaje supervisado, lo cual se limita para desarrolladores estudiantes debido al poco tiempo que se tiene, requeriría hacer etiquetas a conjuntos de datos y hacer un proceso mucho más elaborado, además, existen otros métodos ligados a la predicción y con menores recursos.

Presentación de la información en la GUI:

La idea de que el programa muestre la información por medio de gráficos de barras, dispersión y tortas, apoyadas por datos numéricos representando dispersión y medidas de tendencia central no es atractiva para el cliente, dejo explicito el hecho de que ya poseía software que contenía tales características, por lo tanto, no se tomara en cuenta esta idea.

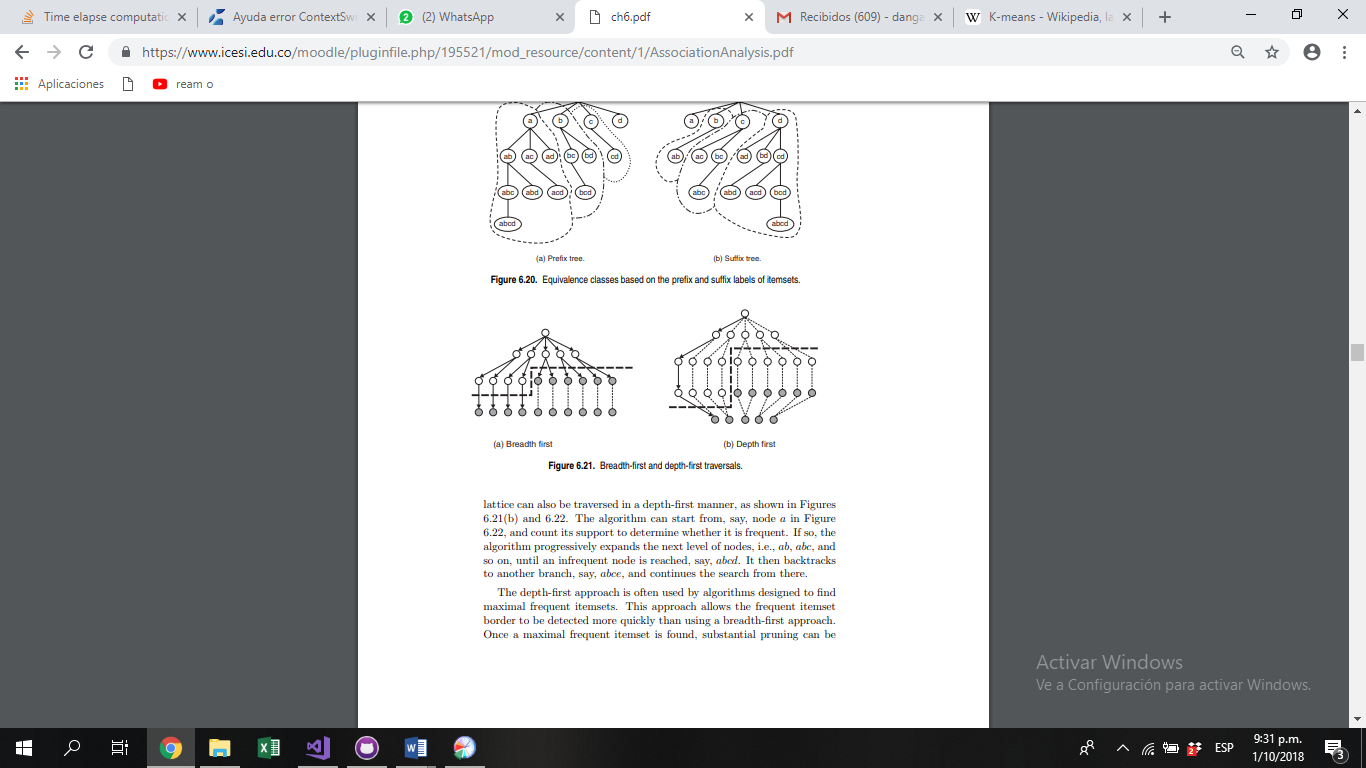
Análisis de los clientes:

Por razone similares a la idea descartada en análisis de transacciones, el SVM resulta ser una alternativa poco probable tanto para transacciones como para clientes, en general, la técnica del aprendizaje supervisado requiere de más detalle para su elaboración.

**Diseños preliminares:**

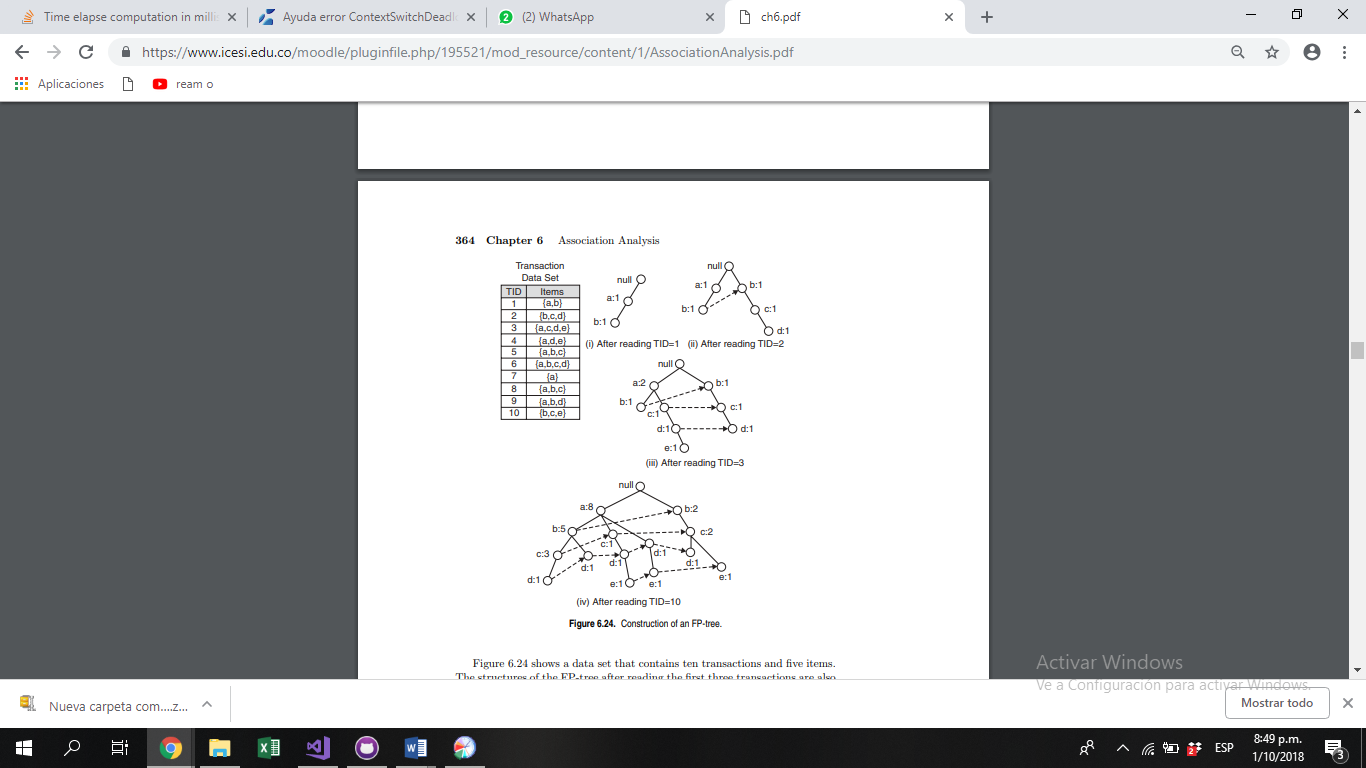
Para generar Frequent ItemSets:

1. El algoritmo va recorriendo los itemsets candidatos en profundidad, en vez de por anchura (Análogamente a los recorridos DFS y BFS de un grafo). Así pues, si bien el Apiori recorre primero los itemsets A, B, C, D ,E, después AB, AC, AD, etc.., recorrerlos por profundidad sería de la forma A, AB, ABC, ABD etc… Así, al encontrar un itemset no frecuente, el algoritmo se devuelve al itemset anterior y pasa a buscar el siguiente (Si para en ABD, se devuelve y pasa a ABCE). Su principal uso es el de encontrar Maximal Frequent Itemsets.



(Association Analyisis: Basic Concepts and Algorithms)

1. Representar las transacciones a través de un FP-Tree, con el objetivo de elaborar un FP Algorithm. Dicha representación es útil para representar las transacciones de manera condensada como un árbol. Los Frequent ítem sets se van encontrando a partir de los “sufijos” de los itemsets (Con qué elemento termina el itemset) siguiendo los caminos que contienen dichos sufijos. Emplea una técnica de “Divide y vencerás”. El siguiente es un ejemplo de FP – tree:



(Association Analyisis: Basic Concepts and Algorithms)

Análisis de las transacciones de Allers:

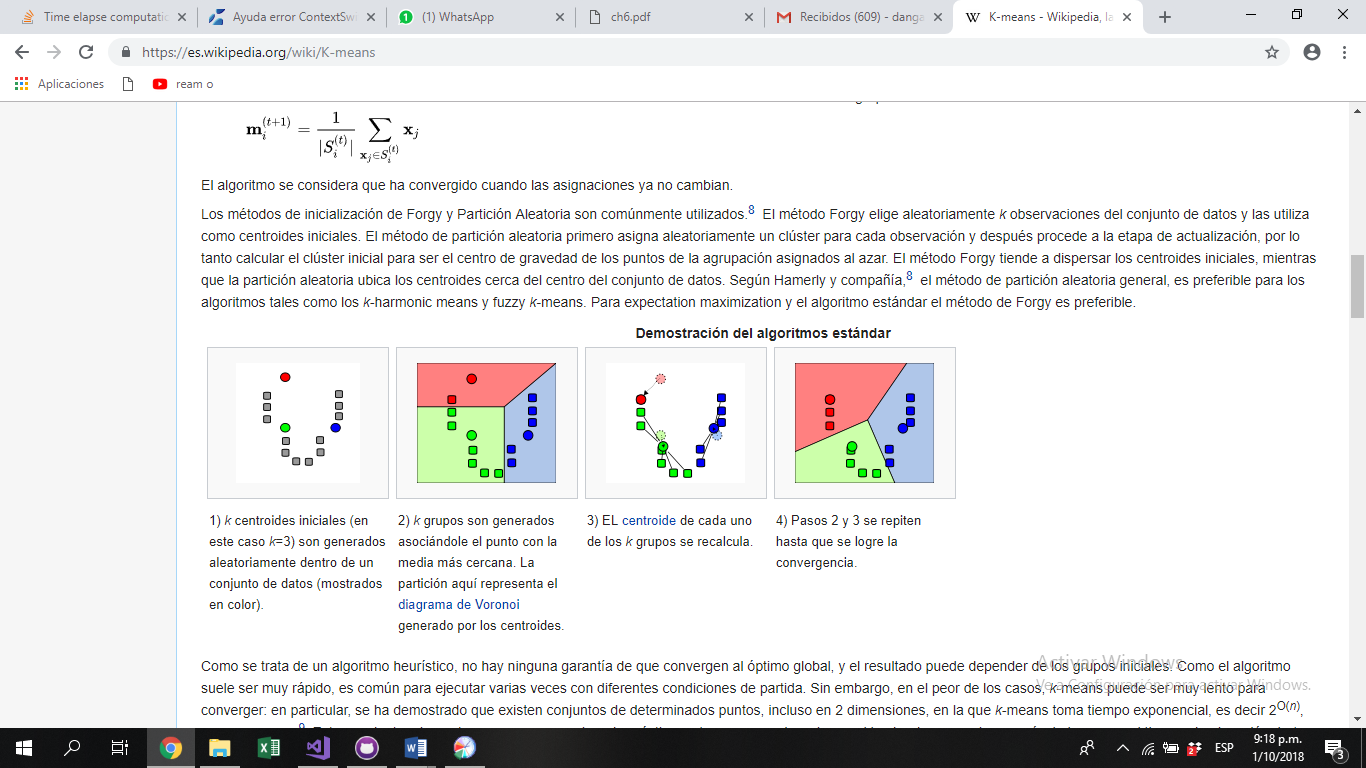
1. La aplicación contiene métodos relacionados con análisis de asociación. Esto es, la aplicación ejecuta procedimientos que aplican el análisis de asociación teniendo en cuenta los métodos predefinidos, las diferentes técnicas y variabilidad en las maneras de recolectar y analizar los datos.
2. Un prototipo del aplicativo que contenga métodos como medidas de tendencia central, anovas, estudio de distribuciones, aplicar probabilidades y concluir con lo que se obtiene usando estadística descriptiva.
3. La aplicación se implementa con matrices multidimensionales para asociar a cada ítem con la transacción y a su vez con los clientes, discretizando algunos atributos para hacer respectivas aproximaciones, en esta aplicación se dejan de utilizar filas y/o columnas para agrupar los objetos del problema.

Presentación de la información en la GUI:

1. La idea de presentar la información por clientes es relevante ya que uno de los objetivos con la aplicación es separar grupos de clientes a los cuales se les podría hacer descuentos u otro tipo de estrategias de marketing mediante el análisis de sus compras históricas, en la GUI, se observaría posibles predicciones de compra para cada cliente y su similitud con otros.
2. Se separa la información de transacciones por mes, esto para facilitar las predicciones y notar más fácilmente el crecimiento o decrecimiento en las ventas de ciertos productos. Esta vez se muestran las reglas extraídas por mes y soportes.

Análisis de los clientes:

1. A partir de los datos de las transacciones, se determinan clientes que son similares entre sí según unas métricas, para así recomendar a un cliente productos que él no está comprando, pero que los clientes similares si adquieren.
2. La aplicación contiene un método de agrupamiento denominado k-means, que ayuda en gran medida a la división de los clientes basados en sus transacciones y al trato de los mismos para realizar análisis más específicos. Si bien se trata de un algoritmo NP – Hard, existen heurísticas empleadas y diferentes variantes que hacen que el problema se solucione con un óptimo local.



(Wikipedia, 2018)

**Fase 5: Evaluación y selección de la mejor solución**

Alcance:

* Las soluciones que se vayan a implementar deben tener una complejidad de tal forma que se puedan llevar a cabo con base al tiempo y conocimientos que se disponen
* Las soluciones deben, por lo menos, llevar a cabo lo mínimo que se pide en los requerimientos funcionales

Criterios transversales a todas las ideas:

**Criterio A:** La idea cumple con la función por la cual se desarrolló, de manera precisa, eficiente y confiable (probada formalmente).

1. La idea no cumple con la función ya sea porque no funciona siempre o porque es ineficiente y poco confiable
2. Cumple con la función, pero lo hace ineficientemente
3. Cumple con la función, es precisa, eficiente y confiable

**Criterio B:** La idea está dentro del alcance

1. Incumple al menos un punto dentro del alcance
2. Cumple con todos los puntos dentro del alcance

Criterios para las ideas para generar frequent ítems:

**Criterio A1:** El principal objetivo de la técnica es optimizar la búsqueda de los ítems sets frecuentes

1. No
2. Sí

**Criterio B2:** El algoritmo genera todos los subconjuntos de los ítems sets frecuentes

1. No
2. Sí

Idea 1: Búsqueda en profundiad

Idea 2: Apoyándose en un árbol FP

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ideas/Criterios | Criterio A | Criterio B | Criterio A1 | Criterio B2 | Sumatoria |
| (1) | 3 | 2 | 1 | 2 | 8 |
| (2) | 3 | 2 | 2 | 2 | 9 |

Si bien con una búsqueda en profundidad se pueden encontrar todos los ítems sets frecuentes de manera “óptima” (basándose en el principio de monotonicidad), sigue utilizando la misma estructura del Apriori y se puede ver como una simple variante de este. Además, su principal objetivo es el de encontrar máximos ítems sets frecuentes. Por su parte, si bien el árbol FP es más complejo de implementar, tiene la ventaja de condensar las transacciones. Además, al ser una estructura diferente a la del Apiori, hay mayores probabilidades que sus eficiencias difieran según las diferentes situaciones (Estadísticamente hablando).

Criterios para las ideas acerca del análisis de transacciones:

**Criterio A1:** La técnica es la adecuada sabiendo que a lo que se quiere llegar es a la inferencia de reglas por medio del análisis de las transacciones

1. No
2. Si

**Criterio B2:** Los resultados de la técnica pueden ser obtenidos mediante otro algoritmo más sencillo propuesto.

1. Sí
2. No

Ideas 1: Utilizar técnicas de asociación

Ideas 2: Utilizar estadística descriptiva

Ideas 3: Utilizar matrices multidimensionales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ideas/Criterios | Criterio A | Criterio B | Criterio A1 | Criterio B2 | Sumatoria |
| (1) | 3 | 2 | 2 | 1 | 8 |
| (2) | 1 | 2 | 1 | 2 | 6 |
| (3) | 1 | 1 | 1 | 1 | 4 |

El recuadro anterior sugiere que para analizar las transacciones de Allers resulta mucho mejor utilizar las técnicas de análisis de asociaciones, por ser especialmente diseñadas para el análisis de transacciones y porque las demás técnicas ya han sido usadas por el cliente o bien son inalcanzables actualmente por los programadores.

Criterios para la interfaz gráfica

**Criterio A1:** La idea presenta la información de manera sencilla, no redundante y completa

1. No
2. Sí

Ideas 1: Mostrar la información categorizada por los clientes

Idea 2: Separar la información por meses y soportes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ideas/Criterios | Criterio A | Criterio B | Criterio A1 | Sumatoria |
| (1) | 3 | 2 | 2 | 7 |
| (2) | 3 | 2 | 2 | 7 |

Ambas ideas de presentar la información por clientes o por meses resultan igual de eficientes y eficaces, pero Allers requiere hacer un marketing personalizado a cada cliente, por lo que presentar la información por clientes se alinea más a sus necesidades.

Criterios para analizar los clientes:

**Criterio A2:** El algoritmo evaluado trata de encontrar patrones y con base a ello hace la clasificación

1. No
2. Si

**Criterio B2:** El algoritmo es supervisado o no.[[13]](#footnote-13)

1. Es supervisado
2. No es supervisado

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ideas/Criterios | Criterio A | Criterio B | Criterio A2 | Criterio B2 | Sumatoria |
| (1) | 3 | 2 | 1 | 2 | 8 |
| (2) | 3 | 2 | 2 | 2 | 9 |

Idea 1: utilizar métricas

Idea 2: Utilizar el algoritmo k-means

El k-means resulta ser el método ideal para encontrar las agrupaciones entre los clientes, se prefiere sobre las métricas al ser más automático de implementar y porque no se tiene que definir criterios de similitud muy específicos.

**Conclusión**

Durante la realización del informe, se encontró con una gran cantidad de material relacionado al análisis de transacciones y clientes, lo cual nos ha sido de mucha utilidad a la hora de generar ideas y en el diseño de la solución, por tanto, cuando se realiza una proyección a futuro del aplicativo, se puede ver con claridad la arquitectura de la solución junto con su interfaz gráfica.

# Bibliografía

(s.f.). *Association Analyisis: Basic Concepts and Algorithms.*

Wikipedia. (01 de 10 de 2018). *K-means*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/K-means

1. Principios SOLID, obtenido de <https://es.wikipedia.org/wiki/SOLID> [↑](#footnote-ref-1)
2. TDD, obtenido de <https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas> [↑](#footnote-ref-2)
3. Obtenido de <http://delivery.acm.org/10.1145/360000/358114/p423-lu.pdf?ip=200.3.193.226&id=358114&acc=ACTIVE%20SERVICE&key=4D9619BEF5D5941F%2EEEED38388178DD50%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1539880364_87696aa691f19a43769cdea6bead38b5> [↑](#footnote-ref-3)
4. ERP, obtenido de https://en.wikipedia.org/wiki/Enterprise\_resource\_planning [↑](#footnote-ref-4)
5. Obtenido de Association Analysis: Basic Concepts and Algorithms pagina 354. [↑](#footnote-ref-5)
6. Obtenido de Association Analysis: Basic Concepts and Algorithms, pagina 355. [↑](#footnote-ref-6)
7. BigData, obtenido de https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/index.html [↑](#footnote-ref-7)
8. Base de datos relacional, obtenido de https://www.ibm.com/support/knowledgecenter/es/SSEPGG\_8.2.0/com.ibm.db2.udb.doc/admin/c0004099.htm [↑](#footnote-ref-8)
9. MapReduce, obtenido de https://es.wikipedia.org/wiki/MapReduce [↑](#footnote-ref-9)
10. Business Intelligence, obtenido de https://www.sinnexus.com/business\_intelligence/ [↑](#footnote-ref-10)
11. SAP Bussines one, obtenido de https://en.wikipedia.org/wiki/SAP\_Business\_One [↑](#footnote-ref-11)
12. QlikView, obtenido de https://www.qlik.com/es-es/products/qlikview [↑](#footnote-ref-12)
13. Al ser supervisado necesita de un gran esfuerzo humano previamente, esfuerzo que aumenta con la cantidad de información. [↑](#footnote-ref-13)