
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“IMPLEMENTACIÓN DE UN MÓDULO DE PREDICCIÓN DE ESCENARIOS PARA ASIGNACIÓN
DE PERSONAL EN EL DESARROLLO DE PROYECTOS”

TESIS

QUE PARA OBTENER EL GRADO DE:
MAESTRA EN SISTEMAS COMPUTACIONALES

PRESENTA:
ING. LORENA HERNÁNDEZ CARMONA

DIRECTOR:
DR. JOSÉ CRISPÍN HERNÁNDEZ HERNÁNDEZ

Agradecimientos

Agradezco a Dios por darme la oportunidad de vivir esta experiencia en mi formación académica y profesional, de igual forma agradezco al Tecnológico de Apizaco por abrirme las puertas y permitirme empezar y culminar mi formación, gracias a Conacyt por dar apoyo económico para la superación de nosotros como profesionistas.

Gracias a mis padres Ma. Socorro Carmona Bonilla y Florencio Hernández Romano, por darme la vida y educarme de la mejor manera posible, por su apoyo y cuidado gracias.

A mi esposo Ricardo Macías Rodríguez por su apoyo incondicional durante mis estudios gracias por ser comprensivo y dejarme realizar como profesionista y a mi hija Camila Macías Hernández por ser mi inspiración y fuerza de cada día, para no rendirme.

A todos los amigos y compañeros que compartieron la experiencia de una etapa más en nuestras vidas, en especial a Laura Reyes Briones, Guadalupe Maya Vázquez y Lauro Reyes Cocoltzi por su amistad incondicional y aquellas aventuras durante nuestro último semestre.

Gracias a mi comité revisor por nunca dejar que me rindiera y brindarme su apoyo incondicional para lograr la obtención de grado, por esas horas de clase y conocimientos compartidos.

Gracias a todos los que me acompañaron en este viaje y estuvieron al pendiente de mis logros.

"Ninguna vida esta completa sin un toque de locura"

Paulo Coelho

Resumen

En la actualidad, las empresas de desarrollo de software se interesan en el seguimiento del avance de sus proyectos, para apoyar a la toma de decisiones que permitan culminar en tiempo y forma el desarrollo de proyectos en proceso. La metodología Kanban se ha utilizado para la gestión de proyectos, permitiendo darles seguimiento basado en su calendario, y siguiendo el flujo del proceso ya establecido. En este trabajo de investigación se presenta el desarrollo de una API que permite monitorear y visualizar la comparación del porcentaje de avance del proyecto: planeado y actual; ayudando a identificar cuando hacer cambios pertinentes para alcanzar el desarrollo del proyecto en la fecha establecida en su formalización.

Por otra parte, la lógica difusa es utilizada como un lenguaje que permite traducir información en lenguaje natural a un lenguaje matemático formal, aún en casos donde el mismo concepto puede tener diferentes grados de imprecisión en diferentes contextos. En este trabajo de investigación se presenta , la implementación de un módulo de selección de patrones graduales frecuentes difusos obtenidos de los históricos de administración de recursos en empresas de desarrollo de software. Se generan las frecuencias de tiempos de cada fase de desarrollo, que permiten aumentar o disminuir los tiempos de desarrollo totales de proyectos software a partir de las posibles relaciones concordantes entre fases de desarrollo de software observadas.

Abstract

Currently, software development companies are interested in monitoring the progress of their projects, to support the decision-making process that will allow the completion of projects in progress in a timely manner. The Kanban methodology has been used for project management, allowing them to be monitored based on their schedule, and following the flow of the process already established. In this research we present the development of an API that allows us to monitor and visualize the comparison of the percentage of progress of the project: planned and current; helping to identify when to make pertinent changes to reach the development of the project in the date established in its formalization.

On the other hand, fuzzy logic is used as a language that allows the translation of natural language information into a formal mathematical language, even in cases, where the same concept may have different degrees of inaccuracy in different contexts. In this research we present the implementation of a module for the selection of fuzzy frequent pattern obtained from resource management histories in software development companies. It generates the time frequencies of each development phase, which allows us to increase or decrease the total development times of software projects based on possible concordant relationships between phases of software development observed.

Índice general

Agradecimientos	II
Resumen	III
Abstract	IV
1 Introducción	1
1.1 Introducción	1
1.2 Descripción del problema	2
1.3 Justificación	3
1.4 Objetivos	4
1.4.1 Objetivo general	4
1.4.2 Objetivos específicos	4
1.5 Estado del arte	5
2 Marco teórico	14
2.1 Lógica Difusa	14
2.1.1 Conjuntos difusos	15
2.1.2 Variables Lingüísticas	16
2.1.3 Funciones de pertenencia	16
2.2 Normalización	16
2.2.1 Ordenamientos difusos	17
2.2.1.1 Ordenamiento fuertemente completo	18
2.2.1.2 Ordenamiento estricto	18
2.2.2 Normas triangulares (T-norms)	19
2.2.2.1 T-norm mínimo	19

2.2.2.2	T-norm producto	19
2.2.2.3	T-norm Lukasiewicz	21
2.2.2.4	T-norm drástico	21
2.2.3	Patrón gradual frecuente	22
2.3	Relaciones difusas	23
2.3.1	Composición de relaciones difusas	24
2.3.2	Propiedades de las relaciones difusas	24
2.4	Planificación de un proyecto	25
2.4.1	Importancia de una buena planeación	25
2.4.2	Valor planeado	26
2.4.3	Estrategia de una buena planeación	26
2.5	Análisis del valor ganado	27
2.6	Metodologías para el control y mejora de procesos	27
2.6.1	Metodología Just in Time	28
2.6.2	Metodología Kanban	30
2.7	Reglas de Asociación	32
2.8	Herramientas de desarrollo	33
2.8.1	Netbeans	33
2.8.2	Maven	34
3	Modelado de la solución	35
3.1	Módulo estatus de un proyecto	35
3.1.1	Análisis y organización de las bases de datos referentes al desempeño de programadores y equipos de programadores	35
3.1.2	Selección del método de procesamiento de las bases de datos	36
3.1.3	Diseño del algoritmo de solución	36
3.1.4	Diseño de estructuras de datos	37
3.1.5	Diseño del medio de presentación	37
3.2	Módulo difuso para la selección de un equipo idóneo	38
3.2.1	Análisis y organización de datos	38
3.2.2	Selección del método para el procesamiento de los datos	38
3.2.3	Diseño del algoritmo de solución	42
3.2.4	Diseño de estructuras de datos	43

3.2.5	Diseño del medio de presentación	43
3.3	Módulo patrones graduales frecuentes	43
3.3.1	Análisis y organización de las bases de datos referentes al desempeño de programadores y equipos de programadores . .	43
3.3.2	Selección del método de normalización de las bases de datos .	45
3.3.3	Diseño del algoritmo de solución	46
3.3.4	Diseño de estructuras de datos	48
3.3.5	Diseño del medio de presentación	49
4	Resultados	51
4.1	Módulo estatus de un proyecto	51
4.1.1	Confección de los instrumentos y métricas de evaluación del módulo	51
4.1.2	Diseño de experimentos para pruebas del módulo	54
4.1.3	Pruebas del algoritmo de valor ganado	54
4.1.4	Elaboración de reporte de interpretación de resultados	54
4.2	Módulo difuso para la selección de un equipo idóneo	56
4.2.1	Confección de los instrumentos y métricas de evaluación del módulo	56
4.2.2	Diseño de experimentos para pruebas del módulo	57
4.2.3	Pruebas del algoritmo	57
4.3	Módulo patrones graduales frecuentes	64
4.3.1	Confección de los instrumentos y métricas de evaluación del módulo	64
4.3.2	Diseño de experimentos para pruebas del módulo	66
4.3.3	Pruebas del algoritmo	66
4.3.4	Reporte de interpretación de resultados	68
5	Conclusiones	71
A	Artículos	73
B	Carta de liberación	87

C Carta de satisfacción

89

Bibliografía

91

Índice de figuras

1.1	Presentación comercial del software de gestión de proyectos Asana . . .	7
1.2	Ambiente de trabajo en Trello	8
1.3	Tablero de trabajo en Basecamp	8
1.4	Presentación comercial del software de gestión de proyectos Teambox	10
1.5	Área de trabajo de JIRA Software	11
1.6	Presentación comercial del software de gestión de proyectos Brightpod	12
1.7	Presentación comercial del software de gestión de proyectos KANBANTool	13
2.1	Representación de un conjunto difuso	15
2.2	Gráfica de T norm mínimo.	20
2.3	Gráfica de T norm producto.	20
2.4	Gráfica de T norm Lukasiewicz.	21
2.5	Gráfica de T norm drástico	22
2.6	Ejemplo de valor ganado	28
2.7	Proceso de la metodología Just In Time	29
2.8	Ejemplo de la metodología Kanban	32
2.9	Sección de trabajo de Netbeans	33
3.1	Resultados.	42
3.2	Algoritmo de solución.	47
4.1	Resultados.	55
4.2	Evaluación del proyecto semanal.	55
4.3	Gráfica de comparación.	56
4.4	Máximos.	59

4.5	Asignación por cumplimiento de tareas	59
4.6	Asignación por puntualidad	59
4.7	Resultado para asignación por habilidad	60
4.8	Resultado para asignación por cumplimiento de tareas	60
4.9	Resultado para asignación por puntualidad	61
4.10	Resultado para asignación por habilidad de programación	61
4.11	Resultado para asignación por cumplimiento de tareas	62
4.12	Resultado para asignación por puntualidad	62
4.13	Extracto de la base de datos utilizada para pruebas experimentales.	65
4.14	Interpretación de la salida.	66
4.15	Interpretación de la salida (primera modificación).	67
4.16	Interpretación de la salida (segunda modificación).	67
4.17	Interpretación de la salida (segunda modificación mostrando diferente umbral).	68
4.18	Lista de atributos con resultados para visualización.	68
4.19	Lista de porcentaje en patrones graduales frecuentes.	70

Índice de tablas

1.1	Ventajas y desventajas de usar Trello Software.	7
1.2	Comparación de herramientas existentes.	13
2.1	Algunas aplicaciones de la lógica difusa.	15
3.1	Base de datos sintética para pruebas experimentales.	35
3.2	Habilidad para programar.	39
3.3	Habilidad requerida para programar en un proyecto.	39
3.4	Selección de la primer fila.	40
3.5	Selección de la primer columna.	40
3.6	Mínimos obtenidos.	41
3.7	Composición max-min.	41
3.8	Relación obtenida.	41
3.9	Base de datos sintética.	43
3.10	Base de datos sintética.	44
3.11	Base de datos sintética parecida a la real.	44
3.12	Base de datos sintética utilizada para ejemplo práctico.	45
3.13	Selección columna a normalizar.	45
3.14	Selección mínimo y máximo.	46
3.15	Columna normalizada.	46
3.16	Base de datos sintética normalizada.	46
4.1	Primer tabla para cálculo de valor ganado.	53
4.2	Valoración por semana.	53
4.3	Cumplimiento de tareas.	58
4.4	Cumplimiento requerido para el proyecto.	58

4.5	Columnas o atributos de la base de datos.	69
-----	---	----

Índice de ecuaciones

2.1	Normalización basada en unidad o Función escalar.	17
2.2	Ordenamiento fuertemente completo	18
2.3	Ordenamiento estricto	18
2.4	T-norm mínimo	19
2.5	T-norm producto	19
2.6	T-norm Lukasiewicz	21
2.7	T-norm drástico	22
2.8	Coficiente gradual	23
2.9	Composición max-min.	24
2.10	Composición máx-product.	24

Capítulo 1

Introducción

1.1. Introducción

La industria de desarrollo de software es de gran importancia en la actualidad por la alta demanda de tecnología que se observa en el país, interviene en todos los procesos que abren paso a la nueva economía. Se le considera una industria blanca que no contamina y que genera fuentes de trabajo bien remuneradas. En esta industria es importante tener en cuenta los tiempos de desarrollo de cada etapa de implementación, para medir el rendimiento de cada integrante de un equipo de trabajo y así determinar un nivel de conocimiento, el cual dé a la empresa, una idea de cómo podrían conformarse con probabilidad de llegar al objetivo de desarrollo en tiempo y forma con el equipo de trabajo adecuado, sin necesidad de tener pérdidas en recursos, ya sean económicos o en tiempo.

La principal función de la industria de desarrollo de software es realizar proyectos a la medida de cada empresa, es decir, en base a las necesidades que presenta. Para muchas empresas el desarrollo de software es un medio de ayuda para contribuir en la gestión de recursos, ya que permite administrar de la mejor manera las herramientas con las que se cuenta para lograr un objetivo.

En el mercado existen herramientas de software que ayudan con la administración de proyectos, e incluyen el uso del modelo Kanban y Just in Time. Sin embargo no son capaces de determinar si una tarea en ejecución está fuera del tiempo planeado de acuerdo a la fecha establecida en su formalización. Esta actividad es realizada

por el líder de proyecto quien monitorea constantemente cada actividad en tiempo real y así identifica si está fuera del tiempo establecido, este tipo de metodologías se planea con respecto a etapas de ejecución siendo posible cambiar la tarea de una a otra etapa según se requiera.

Es importante llevar de la mano la tecnología pues es una herramienta que facilita y apoya el trabajo de cualquier persona. En este capítulo se describen algunas de las tantas herramientas que existen para la gestión de proyectos con la finalidad de dar a conocer como funcionan y en que aspectos pueden servir a un usuario con necesidades que puedan adaptarse a dicho software y así apoyarse de estas herramientas para agilizar sus tareas y terminar en menor tiempo posible sin que exista algún tipo de retraso.

1.2. Descripción del problema

Una de las problemáticas con las que se encuentra la industria de desarrollo de software es la demanda que se tiene, la cual requiere de un equipo trabajando en una implementación y al mismo tiempo en otros proyectos, dando como resultado un retraso en fechas de entrega, o bien en fechas de ejecución, es por eso que la misma empresa necesita administrar sus propios recursos para llegar a su objetivo planeado en tiempo y forma.

Para dar solución a este problema se propone el desarrollo de una API que permita la valoración de un proyecto en ejecución determinando el avance respecto a la planeación que se realizó antes de empezar la implementación, o bien un aumento en tiempo, dando oportunidad al líder de proyecto tomar decisiones para llegar al fin establecido en su planeación, empleando el método de valor ganado para determinar el porcentaje de avance actual con respecto del que se estipuló al inicio y el porcentaje que tendría que estar cubierto.

Una segunda problemática que se puede encontrar en empresas dedicadas al desarrollo de software, es el constante atraso que se tiene en la entrega de un sistema, causando tiempos perdidos, a veces disminuyendo la credibilidad y creando desconfianza en los clientes.

La solución propuesta consiste en buscar coincidencias en un histórico existente,

para determinar cuáles son los errores cometidos frecuentemente, y así poder dar una posible solución para enfrentar un problema similar o igual. En el presente trabajo se desarrolla una propuesta de solución e implementación de un módulo de minería de datos basada en extracción de patrones graduales frecuentes para determinar las tareas que deben tener prioridad en la implementación con el fin de evitar tomar más tiempo del necesario en la ejecución en una etapa de desarrollo dentro de un proyecto.

Existe una problemática mas la cual consiste en que la empresa busca mejorar sus procesos de desarrollo en cada equipo que se designe para un proyecto nuevo de manera que este tenga el mejor desempeño en base a lo que se requiere lograr, es decir que se elija a los mejores prospectos para trabajar en base a las habilidades de programación, disponibilidad, capacidad de trabajo en equipo heterogéneo que tenga cada uno de los integrantes de la empresa.

Para este caso se propone un modulo en el cual se analicen las diferentes características de los integrantes de la empresa a base de relaciones difusas para plantear el mejor esenario de elección de personas para su asignación a un proyecto con respecto a los requisitos que tenga.

1.3. Justificación

Es evidente que la tecnología genera resultados positivos y nuevas fronteras en el desarrollo de software, siendo capaz de crear sistemas y aplicaciones inteligentes, intuitivas y funcionales para cubrir las necesidades que requiere la sociedad actual, proporcionando una mejora en muchos aspectos como reducción de tiempos, consultas inmediatas, administración adecuada de recursos y la estimación y proyección de resultados. Se busca una mejora a la herramienta con la que cuenta la empresa para que genere la satisfacción de ésta y sus clientes con un módulo que aplique la minería de datos con patrones graduales a un grupo de datos referentes al desempeño de programadores y un módulo más proyectando el avance de un proyecto en marcha dando a conocer un posible atraso sobre la programación del mismo. En el primer módulo la aplicación desarrollada pretende ayudar en la programación de un proyecto a desarrollar dando un porcentaje a cada fase con respecto a otra dándole a

saber al usuario que tanto tiempo se debe invertir en una fase para que la siguiente tenga éxito, por ejemplo si se tiene una fase de diseño y una mas de programación es posible que se determine a base de correlaciones que se debe invertir mas tiempo en diseño para que la programación no contenga errores y sea necesario re programar la tarea y regresar a la fase de diseño para corregir dichos errores. Para un segundo módulo con el desarrollo de una aplicación se ayudará a determinar como es que se esta comportando el desarrollo de un proyecto con respecto a una planeación ya establecida con tiempos específicos, para determinar si esté va en tiempo o esta desfasado y poder tomar decisiones que permitan contrarrestar este desfase y lograr la meta comprometida en tiempo y forma. Complementando con un tercer módulo se pretende ayudar a la empresa de desarrollo de software en la elección de un equipo de trabajo para llegar en tiempo y forma al fin de un proyecto. La propuesta de selección se realizará en base a los requerimientos del proyecto para el cual se busca un equipo idóneo. Así la empresa asegura un buen desarrollo y disminuye la probabilidad de fallas en su planeación y fecha de entrega.

1.4. Objetivos

1.4.1. Objetivo general

Implementar tres módulos, el primero dedicado a la extracción de patrones graduales frecuentes para dar una posible correlación o dependencia de una fase con otra tomando en cuenta un histórico de tiempos en cada una de las fases de desarrollo el segundo; dedicado a valorar un proyecto basado en su tiempo de desarrollo real contra una planeación previa al inicio del desarrollo y así poder determinar el estado del proyecto. Y un tercero dedicado a encontrar los mejores escenarios para la selección del equipo idóneo para un proyecto a base de relaciones difusas tomando en cuenta varios factores de trabajo.

1.4.2. Objetivos específicos

Primer módulo:

- Analizar y organizar la base de datos

- Seleccionar el método de normalización de las bases de datos
- Diseñar el algoritmo de normalización de las bases de datos
- Diseñar el algoritmo de extracción de patrones graduales frecuentes
- Implementar el algoritmo de normalización y extracción de patrones graduales frecuentes

Segundo módulo:

- Analizar y organizar la base de datos
- Seleccionar metodología aplicable
- Diseñar el algoritmo de valoración para un proyecto
- Implementar el algoritmo de valoración para un proyecto

Tercer módulo:

- Analizar y organizar la base de datos
- Selección de una metodología
- Diseñar el algoritmo
- Implementar algoritmo

1.5. Estado del arte

La gestión de proyectos, entonces, es el uso del conocimiento, habilidades y técnicas para ejecutar proyectos de manera eficaz y eficiente. Se trata de una competencia estratégica para organizaciones, que permite vincular los resultados de un proyecto con las metas comerciales para posicionarse mejor en el mercado [21].

En esta sección se describe el estado del arte de algunas herramientas que permiten la gestión de proyectos, que se encuentran presentes en el mercado actualmente.

- Asana. Es una herramienta de gestión de proyectos que permite gestionar las tareas mediante la asignación de responsables y fechas, así como generar una gráfica que refleja el flujo de trabajo a lo largo del desarrollo del proyecto. Se trata de una aplicación para gestionar, administrar y compartir proyectos. Permite organizar tareas y objetivos, así como visualizar el avance particular o general de los proyectos en los que los diferentes miembros del equipo están involucrados [14].

Asana fue creada en Febrero de 2009 por Dustin Moskovitz y Justin Rosenstein. La versión lanzada recientemente fue presentada el 30 de Septiembre, este cambio está encaminado a facilitar las tareas de seguimiento y cumplimiento de los objetivos.

El producto tiene muchas funcionalidades, como espacios de trabajo, proyectos personales, tareas, etiquetas, notas, comentarios, y un buzón que organiza y actualiza la información en tiempo real. El producto está diseñado para facilitar a las personas y a los equipos la planificación en la gestión de sus proyectos y tareas. Cada equipo tiene un espacio de trabajo. Los espacios de trabajo contienen proyectos y los proyectos contienen tareas [18].

Los fundadores de Asana piensan que “Las tareas son unidades atómicas de trabajo.” En cada tarea, los usuarios pueden añadir notas, comentarios, ficheros y etiquetas. Los usuarios pueden seguir proyectos y tareas, y cuando cambia el estado de un proyecto o una tarea, los seguidores tienen la información de dichos cambios en sus buzones. Es gratis para equipos de hasta 15 miembros. En la figura 1.1 se muestra el logo de presentación comercial del software que se puede adquirir.

- Trello. Trello es una herramienta de colaboración que organiza los proyectos en tableros. Gracias a Trello, se puede saber cuáles son las tareas que se llevan a cabo, quién trabaja en una tarea determinada y cuál es el estado de un proceso. Consiste en la creación de tableros para cada proyecto. Dentro de cada tablero hay listas y cada lista tiene tarjetas con información detallada que puede navegar hacia otras listas dentro del mismo tablero. Puede ser utilizado desde



Figura 1.1. Presentación comercial del software de gestión de proyectos Asana

un teléfono o computadora de escritorio y se pueden agregar a las personas que necesiten en cada proyecto [8].

En la tabla 1.1 se dan a conocer algunas de las ventajas que se tienen al usar este software al igual que sus desventajas.

Ventajas	Desventajas
Versátil Potentes funciones de trabajo en grupo Actividades actualizadas en tiempo real Comparte públicamente los tableros Etiquetado y datos completos para tarjetas	Especifica para ciertas metodologías

Tabla 1.1. Ventajas y desventajas de usar Trello Software.

En la figura 1.2 se presenta un ejemplo del ambiente de trabajo que tiene esta herramienta.

- Basecamp. Es un gestor de proyectos, usado por empresas de todo el mundo para la organización y gestión de tareas de proyectos colectivos. Ofrece diversos planes entre los que se incluye uso ilimitado y gratuito, pero que sólo permite administrar un proyecto y tiene bloqueada la opción para subir archivos [7].

Dispone de gestión de tareas simplificado para dar rápido acceso a los integrantes del equipo al trabajo que deben desempeñar. En la figura 1.3 se muestra el tablero de trabajo de la herramienta descrita antes. Dentro de sus funcionalidades se pueden encontrar las siguientes:

- Gestión de tareas: permite vincular las tareas con los trabajadores,

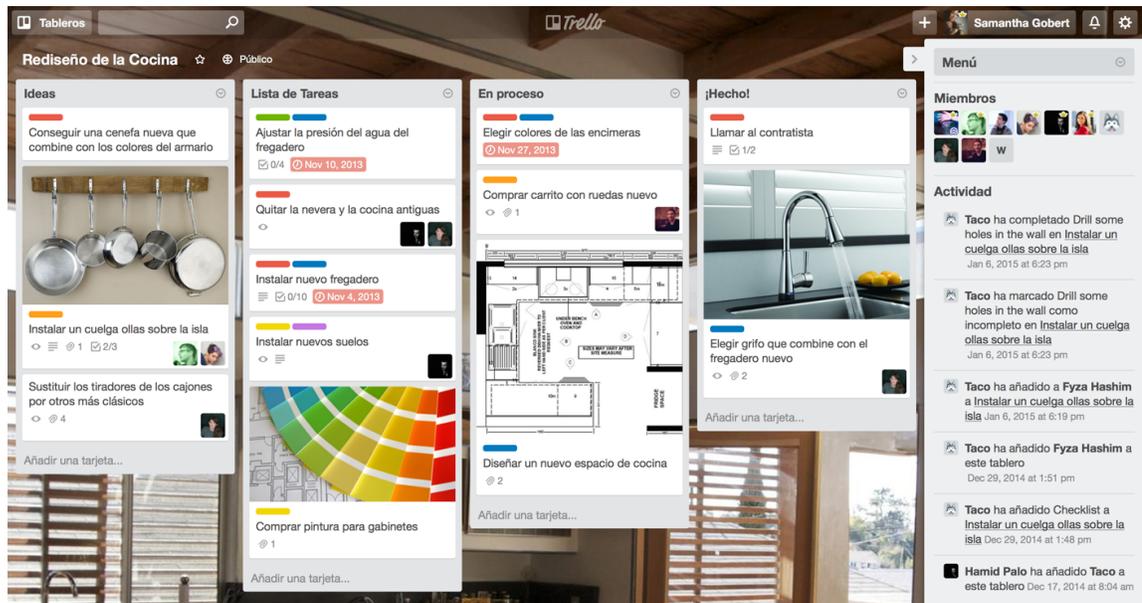


Figura 1.2. Ambiente de trabajo en Trello

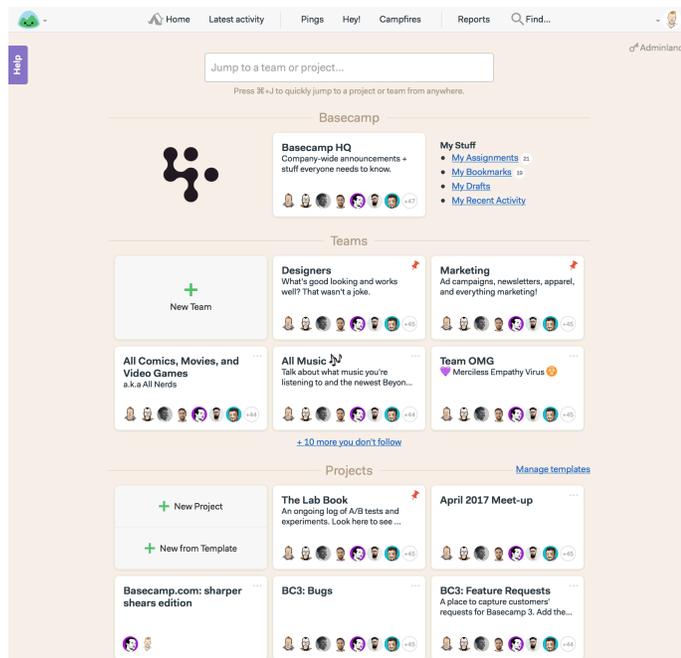


Figura 1.3. Tablero de trabajo en Basecamp

fechas dentro del calendario y adjunta documentación por si es necesario compartir conocimiento.

- **Discusiones colaborativas:** permite la creación de un foro entre trabajadores para poder compartir información mediante las entradas de texto. En las discusiones se permite adjuntar imágenes, enlaces con referencia: lugares web e incluso incrustar vídeos.
- **Lista de tareas:** consiste en un listado que permite ver a simple vista las tareas a realizar en dos grupos: el grupo de todas las tareas y las que son básicas. También se pueden ordenar por asignaciones y por fechas.
- **Documentos de texto:** la aplicación web ofrece a los usuarios la posibilidad de utilizar un blog de notas en donde guardar contenidos de texto.
- **Eventos:** Basecamp permite a los usuarios crear eventos dentro de un calendario. Se indica un título, fecha y los participantes vinculados a éste. De manera que el calendario de cada trabajador quedará actualizado, programando el evento dentro de la fecha seleccionada y dar aviso mediante un correo electrónico.
- **Agenda:** la herramienta permite visualizar un calendario con los eventos o tareas que se tienen vinculadas, de manera que con un simple vistazo se puede saber el trabajo a realizar o los eventos a los que han sido citados sin tener que revisar cada una de las tareas.
- **Notificaciones:** éstas se generan en tiempo real de modo que cuando un trabajador registra, por ejemplo, una tarea a otro usuario, aparecen notificaciones en tiempo real.
- **Creación de proyectos mediante plantillas:** es una funcionalidad muy interesante si se van a crear muchos proyectos parecidos ya que ahorra tiempo al jefe de proyecto a la hora de personalizar cada uno de ellos.

El inconveniente con este software es que sólo se ofrece en idioma inglés.

- **Teambox.** Es, al igual que BaseCamp, una de las herramientas más fáciles e intuitivas de usar. Es muy útil para organizar proyectos colaborativos. Ayuda

a gestionar muy fácilmente la importancia y prioridad de las tareas, y permite que los usuarios envíen actualizaciones sobre el progreso del proyecto [4].

Teambox permite compartir tareas y documentos de una forma muy sencilla. Con una interface similar a Facebook, hace el trabajo más social y permite implicar a miembros del equipo dentro y fuera de la empresa. Las funcionalidades son de tareas, documentos y conversaciones, y además integra herramientas como Google Docs y Calendar.

Es casi exclusivo de la web por lo que la integración en los diferentes navegadores es excepcional, ya que esta adaptada incluso para trabajar perfectamente desde móvil, además de poseer aplicaciones nativas para Ipad e Iphone. En la figura 1.4 se observa el logo de la presentación comercial de esta herramienta.



Figura 1.4. Presentación comercial del software de gestión de proyectos Teambox

- JIRA Software. Es una aplicación web para el seguimiento de errores, de incidentes y para la gestión operativa de proyectos. JIRA Software también se utiliza en áreas no técnicas para la administración de tareas. JIRA puede ser utilizado para la gestión de procesos y para la mejora de procesos gracias a sus funciones para la organización de flujos de trabajo [16].

Mediante esta aplicación los jefes de proyecto establecen:

- El calendario de las diferentes fases en que se compone el proyecto.
- La estimación de la duración de las tareas.
- La asignación de las diferentes tareas y casos de uso, a los desarrolladores.
- Control de los plazos del proyecto.
- Control de los costos del proyecto.
- Control y gestión de las auditorias realizadas sobre el proyecto.

- Identificación y gestión de los riesgos surgidos durante el proyecto.
- Identificación y control de las no conformidades que puedan surgir durante el proyecto.
- Gestión de las acciones correctivas sobre los riesgos y no conformidades encontrados.
- Gestión de la asignación de responsabilidades a terceras partes.

En la figura 1.5 se muestra un ejemplo del área de trabajo que maneja esta herramienta.

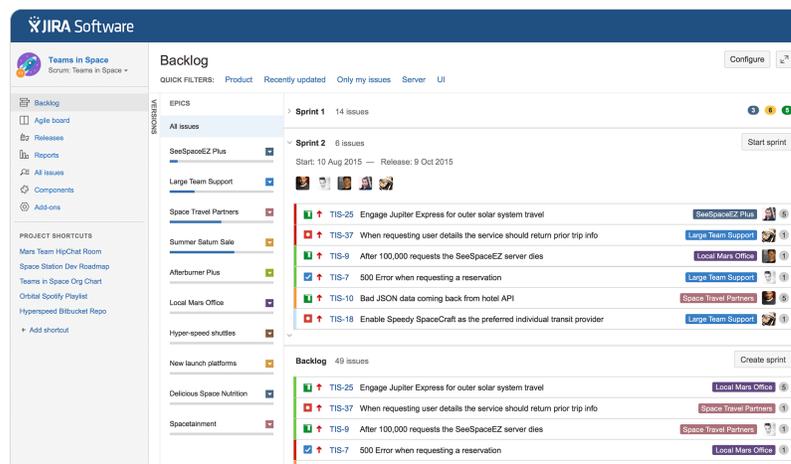


Figura 1.5. Área de trabajo de JIRA Software

- Brightpod. Es una plataforma de software de gestión de proyectos basado en la nube. Es adecuado para todo tipo de empresas, y se utiliza principalmente por las agencias de marketing o los equipos de marketing de las empresas.

En Brightpod, los proyectos son llamados “pods”, y los usuarios pueden agregar flujos de trabajo integrados por las listas de tarea y, metas. A las tareas también se les puede asignar un nivel de prioridad. Los indicadores visuales incluyen barras de progreso.

Tiene un diseño y un servicio muy intuitivo que ayudará a administrar los proyectos de manera más eficiente. Ofrece gestión de proyectos y tareas, listas de tareas, flujos de trabajo, calendario, mensajes, informes, personalizaciones

de la interfaz de usuario, y la integración con Google Drive y Dropbox [1]. En la figura 1.6 se muestra la presentación comercial de este software gestor de proyectos.



Figura 1.6. Presentación comercial del software de gestión de proyectos Brightpod

- KANBAN TooL. Es una solución para la gestión visual de procesos que ayuda a los equipos a trabajar más eficientemente, visualizar el flujo de trabajo, analizar y mejorar los procesos de trabajo de acuerdo con los métodos Kanban [12].

Se trata de una herramienta para poder controlar el seguimiento de un desarrollo basado en el método Kanban de forma online, de manera que se puede tener tableros de trabajo en los cuales el jefe de proyecto puede seguir las diferentes evoluciones. Aparte de crear diferentes tableros para seguir el método, da la facilidad de crear gráficas indicando los diferentes estados de la aplicación. Es una herramienta perfecta, ya que es gratuita hasta dos usuarios [13].

En la figura 1.7 se muestra la presentación comercial de dicha herramienta para su adquisición.

En la tabla 1.2 se muestra una comparación de algunas características con las que cuentan los diferentes tipos de software existentes en el mercado para la gestión de proyectos, siendo sólo algunos de la gran variedad que se puede encontrar. En

KANBAN TOOL

Figura 1.7. Presentación comercial del software de gestión de proyectos KANBANTool

Software	Asana	Trello	Basecamp	Teambox	JIRA Software	Brightpod	KANBAN Tool
Workspaces	Si	Si	Si	Si	Si	Si	Si
Notificaciones	Si	No	No	Si	No	No	No
Integración con otras plataformas	Si	No	Si	No	Si	No	No
Aplicación móvil	Si	Si	Si	No	Si	No	No
Manejo de pizarra de equipo de trabajo	No	Si	No	No	No	No	Si
Disponible para descargar	Si	Si	No	Si	Si	Si	No
Incluye diagrama de Gantt ó gráficas	No	No	Si	Si	Si	No	Si
Software gratuito	Limitado	Limitado	Limitado	Limitado	Limitado	Limitado	Limitado

Tabla 1.2. Comparación de herramientas existentes.

la tabla se presenta lo mínimo que debe ofrecer una herramienta para la gestión de proyectos. La aplicación en la que se trabajará cuenta con estas características mínimas dando como complemento el módulo desarrollado y descrito en este trabajo.

Capítulo 2

Marco teórico

2.1. Lógica Difusa

Una de las principales aplicaciones de la lógica difusa son sistemas de control. A partir de una entrada genere una salida para actuar sobre determinado mecanismo.

Es un tipo de lógica que reconoce más que simples valores verdaderos y falsos. Con lógica difusa, las proposiciones pueden ser representadas con grados de veracidad o falsedad. Por ejemplo, la sentencia *"hoy es un día soleado"*, puede ser 100 % verdad si no hay nubes, 80 % verdad si hay pocas nubes, 50 % verdad si existe neblina y 0 % si llueve todo el día.

La Lógica Difusa ha sido probada ampliamente en sistemas expertos y otras aplicaciones de inteligencia artificial. Es también utilizada en algunos correctores de voz para sugerir una lista de probables palabras a reemplazar por una mal dicha. La Lógica Difusa actualmente está relacionada y fundamentada en la teoría de los Conjuntos Difusos. Según esta teoría, el grado de pertenencia de un elemento a un conjunto va a estar determinado por una función de pertenencia, que puede tomar todos los valores reales comprendidos en el intervalo $[0,1]$ [17].

Algunas áreas de aplicación de la lógica difusa se describen en la tabla 2.1:

Área	Aplicaciones
Informática	Computadores, desarrollo de software, procesamiento de datos.
Transportes	Sistemas de transmisión, y frenado de automóviles, regularización de consumo.
Industria	Robots, reconocimiento de patrones, visión artificial, procesamiento de señales, sensores.
Electrodomésticos	Lavadoras, aire acondicionado, hornos.
Economía	Análisis de mercados y de riesgos, estimaciones, sistemas de decisión.
Simulación	Modelado de sistemas no lineales.
Medicina	Diagnóstico clínico
Microelectrónica	Fabricación de chips
Consumo	Cámaras de vídeo y TV

Tabla 2.1. Algunas aplicaciones de la lógica difusa.

2.1.1. Conjuntos difusos

La función de pertenencia de un elemento a un Conjunto Difuso se representa según la figura 2.1.

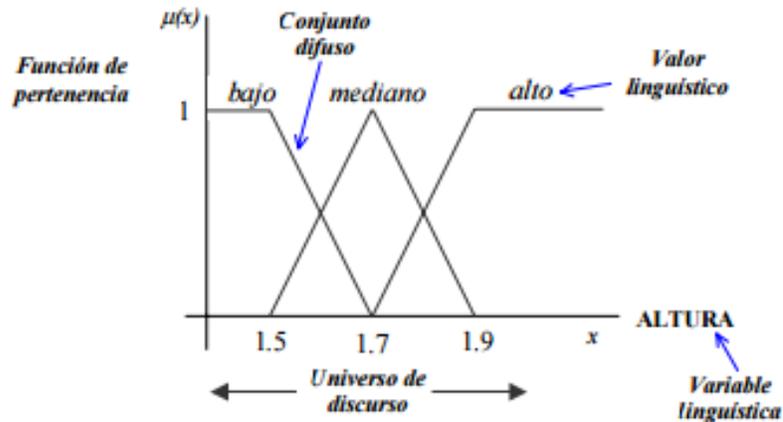


Figura 2.1. Representación de un conjunto difuso

Los conjuntos difusos no son mas que una serie de grados de verdad que suelen utilizarse en el lenguaje natural para describir una situación que cae en un grado de verdad o falsedad. Un ejemplo claro y utilizado en la vida diaria es el clima, "hoy está medio nublado", a esto se le llama variable lingüística la cual determina el grado de verdad o falsedad [22]. En los conjuntos difusos se agrupa una colección de elementos, en la que es posible determinar para un objeto o situación cualquiera, en un universo dado, si éste pertenece o no al conjunto.

2.1.2. Variables Lingüísticas

Una variable lingüística es un grupo de términos, que permite expresar y trabajar con observaciones y medidas de incertidumbre; de un conjunto difuso en un universo de discurso, es una alternativa para modelar el pensamiento humano.

2.1.3. Funciones de pertenencia

Son fórmulas matemáticas que permiten caracterizar conjuntos difusos, así como su forma geométrica mediante la determinación del grado de pertenencia en el intervalo $[0,1]$ de cada uno de los elementos comprendidos en tales conjuntos.

2.2. Normalización

La normalización de bases de datos consiste en aplicar una serie de reglas, específicamente no se permite que un campo contenga más de un dato en su dominio de columna ni valores nulos.

La normalización se utiliza para evitar la redundancia de los datos, disminuir problemas de actualización de los datos en las tablas actuales y proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

- Cada tabla debe tener un nombre único
- No puede haber dos filas iguales
- No se permiten los duplicados
- Todos los datos en una columna deben ser del mismo tipo

La normalización consiste en transformar los datos reales en una escala de cero a uno, en este trabajo de investigación se utiliza el método de normalización basado en unidad, el cual consiste en tomar cada columna para encontrar de esta el número máximo y mínimo, para poder realizar la operación correspondiente con

cada elemento que se encuentra en la columna, prosiguiendo a realizar este cálculo por cada columna que contenga la base de datos [5].

Normalización basada en unidad o Función escalar

Es un método utilizado para estandarizar la gama de variables o características de datos independientes en el procesamiento de datos, que también se conoce como la normalización de datos y, en general se lleva a cabo durante la etapa de procesamiento previo de datos. El método más sencillo es cambiar la escala de la gama de características para escalar el rango de $[0, 1]$ o $[-1, 1]$. Selección de la gama de destino depende de la naturaleza de los datos. La ecuación general es dada como:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.1)$$

donde x es un valor original, x' es el valor normalizado. El resultado de combinar los ordenamientos difusos y la medida de correlación de rango es un robusto coeficiente de correlación de rangos idealmente adecuado para medir la correlación de rangos para datos numéricos perturbados por el ruido. Este innovador coeficiente de correlación se conoce como medida de correlación de rango basada en el orden fuzzy [23].

2.2.1. Ordenamientos difusos

Los ordenamientos difusos están diseñados para evitar la ambigüedad de los datos que se requieren procesar. Es por ello que se utiliza el término difuso pues es posible determinar mas de un resultado verdadero o falso, dando posibilidad de expresar enunciados como alto, medio, bajo. Mostrando un ejemplo para dichos enunciados se puede decir que todos los hombres son altos en México, se sabe que no todos son altos pues algunos tienen estatura baja, estatura media y estatura alta, a esto se refiere el término de utilizar diferentes expresiones distintas a un verdadero o falso. El orden difuso es una relación transitiva difusa a continuación se describen algunos de los tipos de ordenamientos difusos.

2.2.1.1. Ordenamiento fuertemente completo

Pertenece a la familia de ordenamientos difusos basados en teoremas para su aplicación para este caso existen algunas reglas o declaraciones.

Este teorema demuestra de la relación difusa binaria es ordenada fuertemente completa si T_L-E_r en R con $r > 0$.

Se calcula la diferencia entre el atributo en la posición j y el atributo en la posición i , multiplicando el resultado por $1-(1/r)$, enseguida se calcula el valor máximo entre 0 y el resultado de la multiplicación, finalmente se calcula el valor mínimo entre 1 y el resultado anterior. Su definición normal es:

$$Lr_{i,j} = \min(1, \max(0, (1 - \frac{1}{r})) * (Ax_j - Ax_i)) \quad (2.2)$$

donde Lr representa el ordenamiento fuertemente completo, r es una constante que debe ser mayor a 0 para dar resultados certeros y depende de la aplicación en la que se esté implementando, i y j representan la posición de una matriz dada, Ax_j y Ax_i se asume como el atributo x en la posición i o j según sea el caso.

2.2.1.2. Ordenamiento estricto

Una relación difusa binaria R es llamada ordenada estricta con respecto a la T -norma T y a la T -equivalente de E .

Se calcula la diferencia entre 1 y el valor resultante del ordenamiento fuertemente completo en su posición i, j . Su definición normal es:

$$R_{i,j} = 1 - Lr_{i,j} \quad (2.3)$$

donde R representa el ordenamiento estricto, i y j representan la posición de una matriz dada, Lr se asume como el ordenamiento fuertemente completo.

2.2.2. Normas triangulares (T-norms)

Las T-norms utilizadas en lógica difusa son una familia de lógicas no clásicas, de manera informal delimitada por tener una semántica que toma el intervalo unitario real $[0, 1]$, para el sistema de valores de verdad y las funciones llamadas t-norms para interpretaciones admisibles de conjunción. Se utilizan principalmente en aplicaciones de lógica difusa y teoría de conjuntos difusos como base teórica para el razonamiento aproximado[22]. Su definición formal para cada una de estas es la siguiente:

2.2.2.1. T-norm mínimo

También conocida como t-norm de Gödel la cual cuenta con la semántica estándar para la lógica difusa dentro de las conjunciones además de que ocurre en la mayoría de las t-norms lógicas difusas. Su fórmula está dada de la siguiente manera como se muestra en la ecuación 2.4:

$$TM = \min(Ax1_{i,j}, Ax2_{i,j}) \quad (2.4)$$

donde $Ax1$ y $Ax2$ representan un atributo de la matriz a la que se aplicará la norma, i y j representan la posición en la matriz.

En la figura 2.2 se puede visualizar una gráfica de t-norm mínimo

2.2.2.2. T-norm producto

la t-norma del producto es la semántica estándar para la conjunción fuerte en la lógica difusa del producto. Es una t-norma de Archimedean estricta. La cual es definida por la ecuación 2.5.

$$TP = Ax1_{i,j} * Ax2_{i,j} \quad (2.5)$$

donde $Ax1$ y $Ax2$ representan un atributo de la matriz a la que se aplicará la norma, i y j representan la posición en la matriz.

En la figura 2.3 se observa un ejemplo gráfico de t-norm producto.

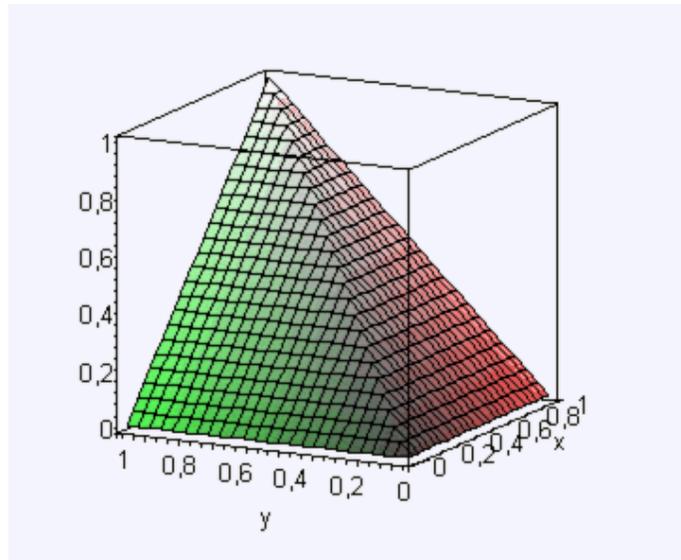


Figura 2.2. Gráfica de T norm mínimo.

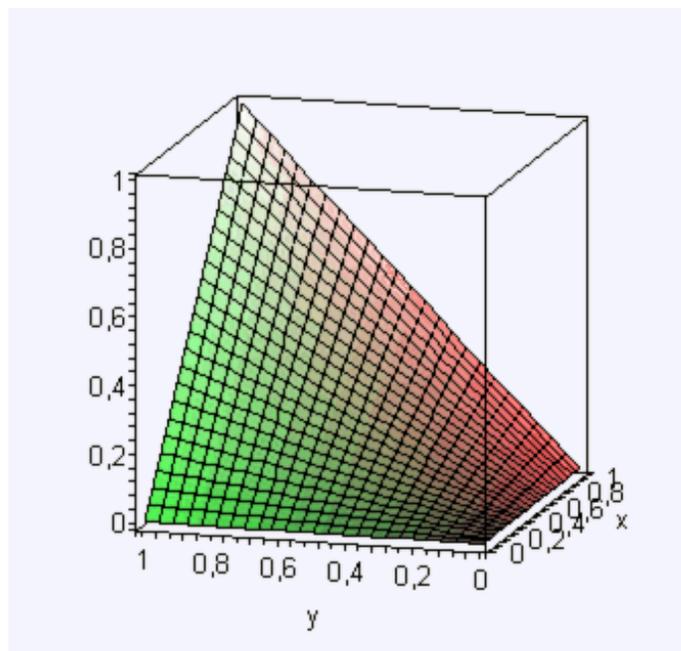


Figura 2.3. Gráfica de T norm producto.

2.2.2.3. T-norm Lukasiewicz

El nombre viene de la semántica estándar para la conjunción fuerte en la lógica difusa Lukasiewicz. Es una t-norma de Archimedean nilpotent, pointwise más pequeño que la t-norma del producto. Su definición formal se muestra en la ecuación 2.6.

$$TL = \max((Ax1_{i,j} + Ax2_{i,j} - 1), 0) \quad (2.6)$$

donde $Ax1$ y $Ax2$ representan un atributo de la matriz a la que se aplicará la norma, i y j representan la posición en la matriz.

En la figura 2.4 se muestra un ejemplo de gráfica para t-norm Lukasiewicz.

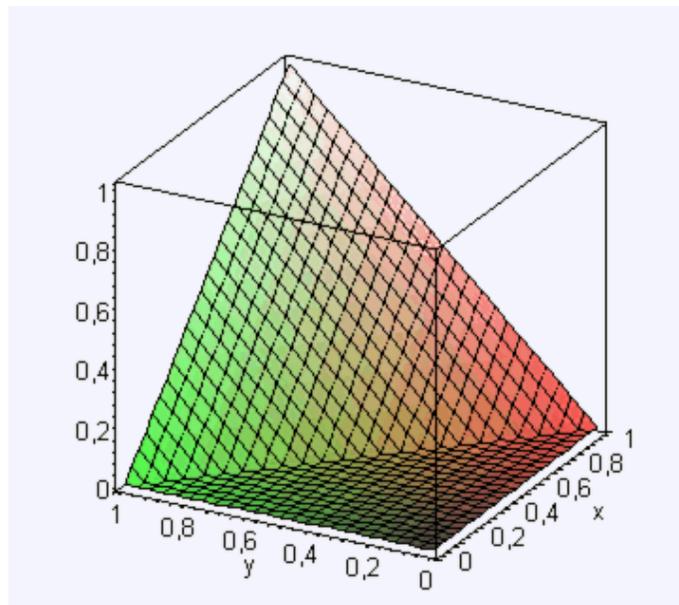


Figura 2.4. Gráfica de T norm Lukasiewicz.

2.2.2.4. T-norm drástico

El nombre refleja el hecho que la t-norma drástica es la t-norma más pequeña. Es una t-norma de Archimedean correcta y continua. La ecuación 2.7 es su definición

formal para su aplicación.

$$TD = \begin{cases} Ax2 & \text{si } Ax1 = 1 \\ Ax1 & \text{si } Ax2 = 1 \\ 0 & \text{en otro caso} \end{cases} \quad (2.7)$$

donde $Ax1$ y $Ax2$ representan valores dentro de una matriz dada.

En la figura 2.5 se puede visualizar un ejemplo de la aplicación de T-norm drástico.

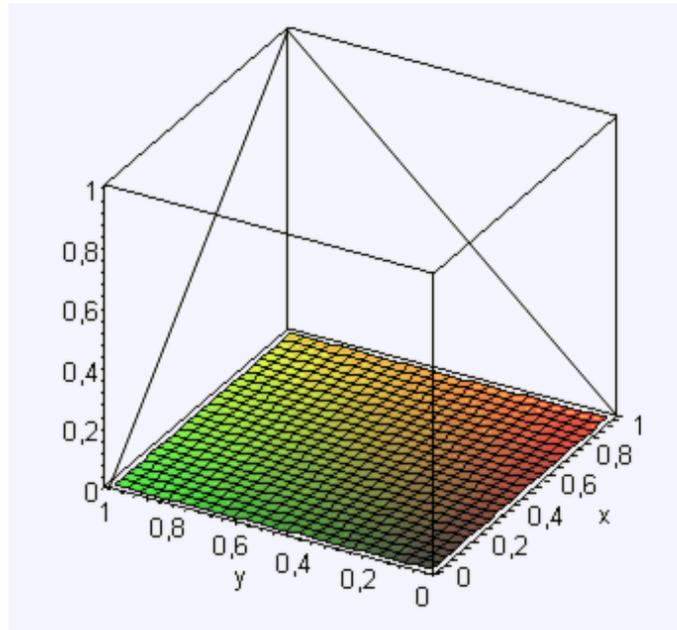


Figura 2.5. Gráfica de T norm drástico

2.2.3. Patrón gradual frecuente

Se usan para medir una variación en la relación existente entre dos tipos de objetos, permiten expresar una tendencia en los datos en lugar de ver solamente cuándo se satisface la relación de similitud [11].

Para realizar el cálculo del coeficiente gradual se utiliza la siguiente ecuación:

$$C = \frac{\sum MR T - norm}{n(n - 1)} \quad (2.8)$$

donde C representa la concordancia entre los atributos analizados, $MR T-norm$ es la matriz resultante de aplicar una norma triangular y n es el número de elementos de la matriz.

2.3. Relaciones difusas

Las relaciones difusas son importantes para el modelado difuso, el diagnóstico difuso y el control difuso. También tienen aplicaciones en campos como la psicología, la medicina, la economía y la sociología.

Una relación difusa se caracteriza por los mismos dos elementos que un conjunto difuso. Primero es una lista que contiene los pares del grado del elemento y de la calidad de miembro, los elementos de la relación se definen como pares ordenados. Estos elementos se agrupan de nuevo con sus grados de miembros, que son valores que van de 0 a 1.

El segundo elemento que caracteriza las relaciones difusas es el espacio universal. Para las relaciones, el espacio universal consiste en un par de pares ordenados. El primer par define el espacio universal que se utilizará para el primer conjunto considerado en la relación, y el segundo par define el espacio universal para el segundo conjunto [19].

Las relaciones difusas se definen como un conjunto difuso sobre el espacio de producto cartesiano de dos conjuntos de elementos ordinarios definidos por X e Y , al producto cartesiano se le denota como: $X \times Y$ el cual resulta de una relación $R(X, Y)$.

Esta relación representa el grado de presencia o ausencia de asociación, interacción o interconexión entre elementos de dos o más conjuntos difusos [9].

2.3.1. Composición de relaciones difusas

Sea una relación difusa en UxV y una relación difusa en VxW . Vamos a presentar dos formas de componer tales relaciones [6].

1. Composición max-min:

Si la intersección se realiza con la operación mín y la proyección con la operación max, se tiene:

$$\prod \gamma(v_j) = \max_{x_i}(\min(\prod x(u_i), \prod R(u_i, w_j))) \quad (2.9)$$

2. Composición max-product:

Si la intersección se realiza con el producto punto y la proyección con la operación max, se tiene:

$$\prod \gamma(v_j) = \max_{x_i}(\prod x(u_i) \times \prod R(u_i, v_j)) \quad (2.10)$$

2.3.2. Propiedades de las relaciones difusas

Enseguida se describen algunas de las propiedad de las relaciones difusas[24]:

- Propiedad reflexiva

Una relación R sobre un conjunto A es reflexiva si para todo $x \in A$ entonces $(x,x) \in R$. En otras palabras una relación es reflexiva si todo elemento del conjunto sobre el que está definida, está relacionado consigo mismo.

$\forall X \in A$ se cumple que $(x,x) \in R$.

- Propiedad antirreflexiva

Una relación R sobre un conjunto A es antirreflexiva si para todo $x \in A$ se cumple que $(x,x) \notin R$, es decir que $\forall x \in A$ se cumple que x no está relacionado consigo mismo.

- Propiedad simétrica

Una relación R sobre un conjunto A es simétrica si para todo $x \in A$, $y \in A$, si $(x,y) \in R$ entonces $(y,x) \in R$. Dicho de otra forma: $\forall x,y \in A$ se cumple que si $(x,y) \in R$ entonces $(y,x) \in R$.

- Propiedad antisimétrica

Una relación R sobre un conjunto A es antisimétrica si para todo $x \in A$, $y \in A$, si $x R y$ e $y R x$ entonces $x=y$. De nuevo: $\forall x,y \in A$ se cumple que si $(x,y), (y,x) \in R$ entonces $x=y$.

- Propiedad asimétrica

Una relación R sobre un conjunto A es asimétrica si para todo $x \in A$, $y \in A$, si $(x,y) \in R$ entonces $(y,x) \notin R$. Dicho de otra forma: $\forall x,y \in A$ se cumple que si $(x,y) \in R$ entonces $(y,x) \notin R$.

- Propiedad transitiva

Una relación R sobre un conjunto A es transitiva si para todo $x \in A$, $y \in A$, $z \in A$ si $(x,y) \in R$ y $(y,z) \in R$ entonces $(x,z) \in R$. $\forall x,y,z \in A$ se cumple que si $(x,y), (y,z) \in R$ entonces $(x,z) \in R$.

2.4. Planificación de un proyecto

2.4.1. Importancia de una buena planeación

En primer lugar se debe tener una idea de en que tiempo se terminará de realizar un desarrollo, quizá para poder dar al usuario una posible fecha para que el valore si le es factible pagar por dicho desarrollo. Para ello se debe tener claro que tareas y en que tiempo se llevarán a cabo de no ser así podría ser un problema para la entrega del desarrollo y generaría problemas con el cliente. Otro punto que debe quedar planeado es a que personas se les designarán cada una de las tareas que tiene un proyecto, ya que se debe tener en cuenta la disponibilidad de cada integrante pues no se puede designar a una persona que tiene responsabilidades en otros proyectos y no pueda cubrir los requerimientos de uno nuevo. En el mercado existen diferentes empresas

que se dedican al desarrollo de software es por ello de que se debe asegurar de que el tiempo planeado se cumpla pues el cliente estará satisfecho y sera una ventaja para combatir la competencia. Tambien se debe tener en cuenta cualquier contratiempo pues no se sabe en que momento podra desfasarse el desarrollo y es necesario tomarlo en cuenta desde la planeación. Por eso es conveniente tener una buena planeación pues de esta depende la entrega en tiempo de un producto software.

2.4.2. Valor planeado

Para cada tarea debe ser asignado un porcentaje del tiempo total que requerirá el desarrollo del proyecto, es decir horas que se estima realizar una tarea. Es importante asignar un tiempo equitativo para todas las tareas siempre teniendo en cuenta los posibles retrasos. Es posible que las tareas sean eliminadas pero siendo remplazadas con otras pues los requerimientos siempre cambien, surgen debido a la necesidad que tiene el cliente.

2.4.3. Estrategia de una buena planeación

- Se debe definir una lista de tareas.
- Estimar tamaño de cada tarea.
- Estimar esfuerzo por tarea.
- Determinar las horas por semana que se le dedicarán.
- Calcular una fecha de finalización para cada tarea y para el todo la lista de tareas.
- Contar con los requisitos para empezar el desarrollo como los datos y su disponibilidad histórica.

El seguimiento será más sencillo si se sigue la lista de tareas en el orden planeado. Para realizar un seguimiento del estado del proyecto en un entorno dinámico, se necesita asignar un valor que mida la contribución de cada tarea en el proyecto. Se puede:

- sumar el valor de las tareas completadas
- comparar este valor con el valor del trabajo total
- calcular el porcentaje de finalización del trabajo

Todo esto se puede reducir al término de valor ganado.

2.5. Análisis del valor ganado

El método del Análisis del Valor Ganado (AVG) es una técnica extremadamente sencilla, a pesar de la sensación diametralmente opuesta que puede provocar la reciente explosión en la literatura de títulos dedicados al tema, así como el poco uso práctico que se le da en el país. El propósito de esta investigación es intentar demostrar cuán sencilla es su aplicación y ofrecer unas claves para un uso correcto y, sobretodo, adecuado.

El análisis del valor ganado se realiza para medir el estado del proyecto por medio de la respuesta a las tres siguientes preguntas: ¿Qué tanto trabajo se planificó? ¿Qué tanto trabajo actualmente se ha completado? ¿Qué tanto ha costado completar el trabajo actual? El Valor Planeado (PV) es una función del tiempo y representa el valor económico que será invertido en un proyecto. El Valor Ganado (EV) es una función del tiempo y representa el avance de un proyecto. En la figura 2.6 se muestra un ejemplo de lo ya mencionado con anterioridad [15].

2.6. Metodologías para el control y mejora de procesos

Es común en la actualidad escuchar el término "servicio completo", y éste implica que la producción y prestación de bienes y servicios alcancen un grado de conformidad y satisfacción en todos los procesos que componen la cadena de valor, desde los procesos que afectan la calidad del producto o la prestación del servicio, como los procesos que inciden en el soporte, la posventa y los servicios complementarios. En la actualidad existen infinidad de instrumentos y herramientas de apoyo para

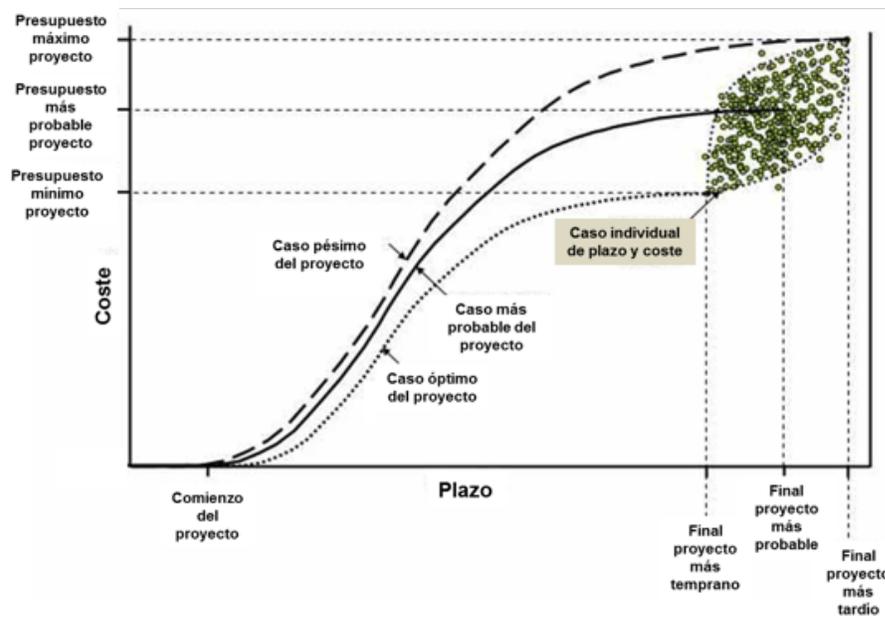


Figura 2.6. Ejemplo de valor ganado

la gestión de procesos de mejora en sus distintos ámbitos: análisis de problemas, generación y organización de ideas o representación de procesos o flujos de trabajo. Cada herramienta tiene una especialización y forma de utilización diferentes, así como diversas ventajas y puntos débiles. Enseguida se presentan dos metodologías importantes en la industria generadora de productos o servicios a la medida.

2.6.1. Metodología Just in Time

Es una filosofía industrial que se concentra en eliminar los costos que se originan de todas las actividades internas y externas y que no añaden valor al producto o servicio en una organización. Los objetivos de un programa JIT incluyen el de producir a la medida de la demanda, mejorar constantemente y eliminar desperdicios de todo tipo [10]. La metodología JIT tiene cuatro objetivos esenciales que son:

- Atacar los problemas fundamentales que se originan por el mal control, manejo y distribución de mercancías; así como las barreras para la flexibilidad y cambio de un ambiente de calidad.

- Eliminar los costos que se originan por el manejo, mantenimiento y todo que no agregue valor al producto o servicio que se originan dentro de las operaciones de la empresa.
- Buscar la simplicidad de los procesos y productos.
- Diseñar sistemas para identificar problemas así como las técnicas y procedimientos que conlleven a implantar sistemas que están encaminados al control total de la calidad.

La implementación de JIT comprende cinco fases:

1. Organización
2. Educación
3. Planeación
4. Ejecución
5. Revisión



Figura 2.7. Proceso de la metodología Just In Time

Ventajas

- Reduce tiempo y costo en inventarios.
- Minimiza pérdidas por suministros inservibles.
- Existe una relación directa con los proveedores.
- Se generan compras a largo plazo permitiendo una planeación óptima ofreciendo mejores precios.
- Es flexible y se pueden corregir errores rápidamente

Desventajas

- Desabasto de recursos para el desarrollo del producto por mala planificación.
- Aumento en los precios con proveedores.

2.6.2. Metodología Kanban

Los sistemas Kanban consisten en un conjunto de formas de comunicarse e intercambiar información entre los diferentes operarios de una línea de producción, de una empresa, o entre proveedor y cliente. Su propósito es simplificar la comunicación, agilizando y evitando errores producidos por falta de información [2]. La herramienta Kanban se basa en unas reglas y consejos que deben cumplirse para poder implementarla correctamente:

- La primera de las reglas es que no se debe mandar producto defectuoso a los procesos subsecuentes, la producción de productos defectuosos implica costos tales como la inversión en materiales, equipo y mano de obra que no va a poder ser vendida.
- La segunda regla es que los procesos subsecuentes requerirán sólo lo necesario. Esto significa que el proceso subsecuente pedirá el material que necesita al proceso anterior, en la cantidad necesaria y en el momento adecuado.

- La tercera dice que hay que producir sólo la cantidad de producto requerida por el proceso siguiente, por lo que se debe restringir la producción a lo requerido y fabricar según llegue el pedido.
- La cuarta regla afirma que hay que optimizar la producción, de manera que se pueda producir solamente la cantidad necesaria requerida por los procesos subsecuentes, se hace necesario para todos los procesos mantener al equipo y a los trabajadores de tal manera que puedan producir materiales en el momento necesario y en la cantidad necesaria.
- En la quinta regla lo que se pretende es evitar las especulaciones: no se vale especular sobre si el proceso subsecuente va a necesitar más material la siguiente vez. Tampoco, el proceso subsecuente puede preguntarle al proceso anterior si podría empezar el siguiente lote un poco más temprano. Es muy importante que esté bien balanceada la producción.
- La sexta y última regla sirve para estabilizar y racionalizar los procesos. El trabajo defectuoso existe si el trabajo no está estandarizado y racionalizado, si esto no es tomado en cuenta seguirán existiendo partes defectuosas.

En la figura 2.8 se ilustra un ejemplo de la metodología kanban y como es su aplicación a los procesos para su mejora continua.

Ventajas

- Reducción del trabajo en progreso.
- Control del flujo de trabajo.
- Fácil de aplicar.
- Promueve el trabajo en equipo.
- Reducción del tiempo perdido.
- Facilidad para añadir mejoras.

Desventajas

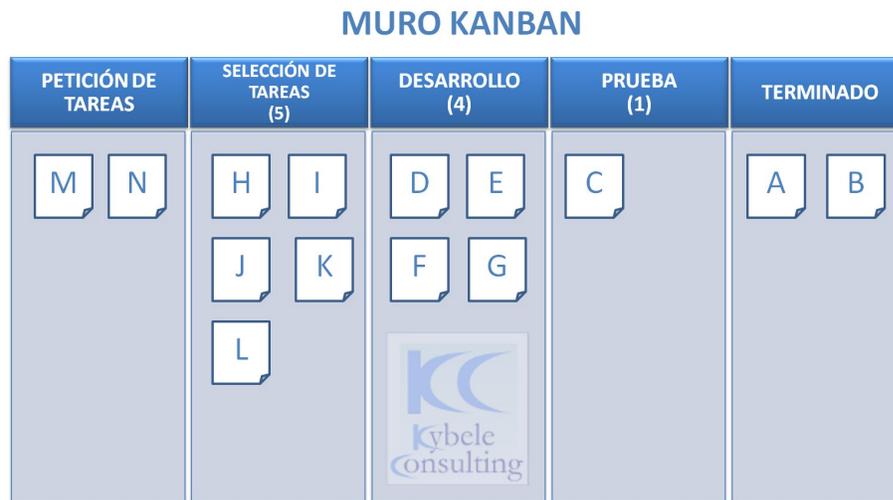


Figura 2.8. Ejemplo de la metodología Kanban

- Dificultad de realizar las entregas a tiempo en grandes proyectos.
- Falta de reglas.
- Dificultad a la hora de preveer posibles problemas.

2.7. Reglas de Asociación

El objetivo de la aplicación de reglas de asociación es tratar de buscar posibles relaciones existentes entre la presencia o no de un ítem en determinados conjunto de transacciones, por ejemplo se utilizan para identificar asociaciones entre la presencia o ausencia de artículos en transacciones comerciales, un ejemplo típico de regla de asociación es: el 90 % de la gente que compra pan, compra leche, en lenguaje natural se puede expresar como: “la mayoría de la gente que compra pan compra leche”, pudiéndose expresar mediante la notación $\text{pan} \Rightarrow \text{leche}$, en esta notación la parte izquierda se denomina antecedente y la parte derecha consecuente.

Adaptándolo a la investigación presentada en este documento se pretende someter a reglas asociativas un porcentaje de avance en el desarrollo de un proyecto.

2.8. Herramientas de desarrollo

2.8.1. Netbeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software [3].

El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). En la figura 2.9 se puede observar el ambiente de trabajo de la herramienta de programación descrita antes.

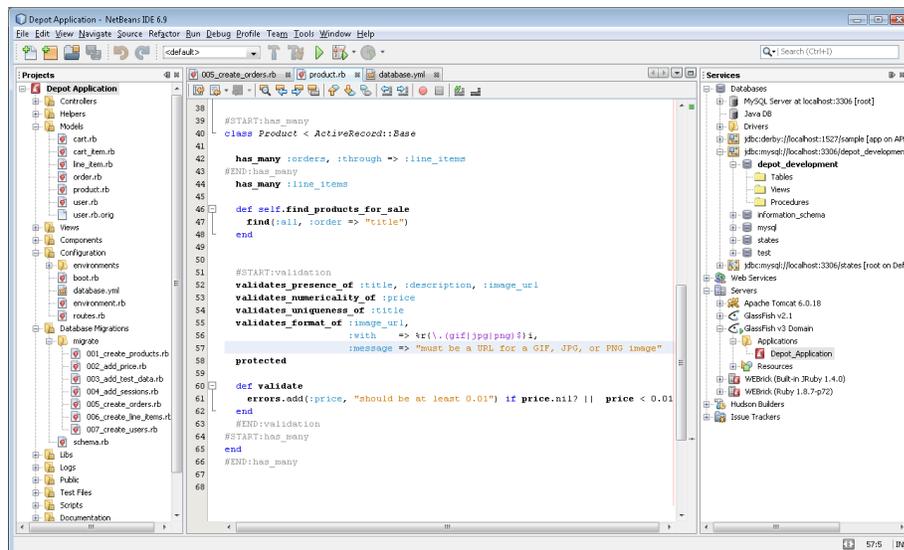


Figura 2.9. Sección de trabajo de Netbeans

2.8.2. Maven

Maven se utiliza en la gestión y construcción de software. Posee la capacidad de realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. Es decir, hace posible la creación de software con dependencias incluidas dentro de la estructura del JAR. Es necesario definir todas las dependencias del proyecto (librerías externas utilizadas) en un fichero propio de todo proyecto Maven, el POM (Project Object Model), es un archivo en formato XML que contiene todo lo necesario para que a la hora de generar el fichero ejecutable de una aplicación éste contenga todo lo que necesita para su ejecución en su interior.

Sin embargo, la característica más importante de Maven es su capacidad de trabajar en red. Cuando se define las dependencias de Maven, este sistema se encargará de ubicar las librerías que se desean utilizar en Maven Central, el cual es un repositorio que contiene cientos de librerías constantemente actualizadas por sus creadores. Maven permite incluso buscar versiones más recientes o más antiguas de un código dado y agregarlas al proyecto. Todo se hará de forma automática sin que el usuario tenga que hacer nada más que definir las dependencias [20].

Capítulo 3

Modelado de la solución

3.1. Módulo estatus de un proyecto

3.1.1. Análisis y organización de las bases de datos referentes al desempeño de programadores y equipos de programadores

Para el desarrollo del modelado de la solución para el primer problema descrito se sugirió una implementación en lenguaje Java, para la cual se utilizaron datos sintéticos, tomando en cuenta la estructura propuesta por la empresa, con el fin de realizar pruebas en el desarrollo del código, estos datos sintéticos se ilustran en la tabla 3.1.

Inicio_Planeado	Inicio_Real	Fin_Planeado	Fin_Real
21/12/2015 00:00	22/12/2015 09:50	29/02/2016 00:00	29/12/2015 09:50
28/12/2015 00:00	23/12/2015 13:11	31/01/2016 00:00	31/12/2015 13:11
24/12/2015 00:00	26/12/2015 09:57	26/12/2016 00:00	27/01/2016 00:00
21/12/2015 00:00	27/12/2015 20:40	29/01/2016 00:00	30/01/2016 00:00
11/01/2016 00:00	29/12/2015 10:00	16/03/2016 00:00	17/03/2016 00:00
04/11/2015 00:00	29/12/2015 11:14	09/02/2016 00:00	10/02/2016 00:00
18/01/2016 00:00	29/12/2015 13:05	18/03/2016 00:00	19/03/2016 00:00
29/12/2015 00:00	29/12/2015 14:29	01/04/2016 00:00	02/04/2016 00:00
29/12/2015 14:34	29/12/2015 14:34	29/12/2015 14:34	30/12/2015 14:34
29/12/2015 14:35	29/12/2015 14:35	29/12/2015 14:35	30/01/2016 14:35

Tabla 3.1. Base de datos sintética para pruebas experimentales.

La base de datos sintética utilizada se creó a partir de la estructura pactada con la empresa para poder integrar el módulo sin dificultad alguna.

3.1.2. Selección del método de procesamiento de las bases de datos

Para realizar la evaluación de los datos de un proyecto en proceso se tendrán en cuenta el método de valor ganado para evaluar el estado del proyecto y poder aplicar un tipo de métrica para su dirección. También es una forma eficaz de comunicar a los interesados del proyecto el estado del presupuesto y diseño en tiempo.

El análisis del valor ganado se realiza para medir el estado del proyecto por medio de la respuesta a las tres siguientes preguntas: ¿Qué tanto trabajo se planifico? ¿Qué tanto trabajo actualmente se ha completado? ¿Qué tanto ha costado completar el trabajo actual?

El Valor Planeado (PV) es una función del tiempo y representa el valor económico que será invertido en un proyecto.

El Valor Ganado (EV) es una función del tiempo y representa el avance de un proyecto.

3.1.3. Diseño del algoritmo de solución

Algoritmo de solución

1. Inicio.
2. Leer base de datos a procesar.
3. Extraer fechas.
4. Determinar el tiempo en días entre fechas planeadas y fechas reales.
5. Realizar cálculo de valor ganado.
6. Valorar por semana el avance del proyecto.
7. Mostrar porcentaje de avance comparado con lo planeado.

El desarrollo del algoritmo consiste en dar un valor de entre 0 y 100 con el fin de determinar el avance de un proyecto en desarrollo para considerar de ser necesario un refuerzo para lograr terminar en tiempo y forma el desarrollo. Para lograr este resultado es necesario obtener y procesar datos del proyecto, entre ellos es necesario obtener el número de tareas que lo conforman y sus fechas de planeación, así como sus fechas reales de las tareas ya realizadas, a su vez se necesita saber si una tarea fue completada para poder ser tomada en cuenta para la valoración del proyecto. Obtenidos los datos se realizará el cálculo de dos tablas que ayudarán en el proceso de esta valoración siendo la segunda la más importante pues en esta se observará el porcentaje de tiempo de planeación y tiempo actual, y así observar cual es el comportamiento del equipo de trabajo de cada proyecto.

3.1.4. Diseño de estructuras de datos

Para el cálculo del método de valor ganado, es necesario generar diferentes estructuras de datos para el almacenamiento de los datos que se obtendrán, la primer estructura que se utiliza es una matriz para almacenar los datos extraídos de un archivo con extensión .csv, para la extracción de fechas se requiere de arreglo de listas del tamaño de los atributos que serán procesados; para el cálculo de los valores paleados y valores ganados es necesario generar una matriz de la longitud del arreglo de listas de fechas que se procesarán, y para determinar avance de cada semana del proyecto se generará una segunda matriz para poder ver en qué parte se está cumpliendo o donde se está generando el retraso en cuanto a tiempos.

3.1.5. Diseño del medio de presentación

Para la presentación de resultados se pretende mostrar una tabla que muestre:

- Semana
- Tiempo Estimado
- Tiempo Actual
- Acumulado de tiempo actual

- Acumulado Valor Planeado
- Acumulado Valor Ganado

Con el fin de dar al usuario resultados de utilidad estos campos serán incluidos en una tabla para poder comparar las dos ultimas columnas y visualizar como se esta comportando el proyecto sobre el desarrollo, para poder tomar decisiones para lograr el objetivo en tiempo y forma.

La salida sera representada en porcentajes de 0 a 100% con el fin de darle un peso a cada tarea que es necesaria para el desarrollo de proyecto de software.

3.2. Módulo difuso para la selección de un equipo idóneo

3.2.1. Análisis y organización de datos

Para el modelado de solución para el problema descrito sobre los diferentes escenarios que se proponen para la mejor selección de un equipo óptimo para desarrollar un proyecto de la manera mas rápida y sin retrasos; se utilizan datos sintéticos con el fin de realizar pruebas para el desarrollo del código, utilizando las tablas 3.2 y 3.3.

3.2.2. Selección del método para el procesamiento de los datos

Para este módulo se aplicarán relaciones difusas con la composición max-min; se muestra un ejemplo utilizando la fórmula 2.9. Lo primero que se debe realizar es tomar la fila de la primer base de datos y la primera columna de la segunda base de datos.

Ya teniendo la selección se compara uno a uno los valores buscando los valores mínimos. Para este caso se toma para la primer comparación 0.2 contra 0.2, después 1 contra 0.4 y así sucesivamente hasta terminar es desarrollo se puede observar en la tabla 3.6. Y de estos valores se escoge el máximo y se asigna a la relación obtenida

id_empleado \ Nivel	Android	Java	AngularJS	MySQL	PHP	JSF
57	0.2	1	0	1	0.2	1
58	0.2	0.6	0.6	0.6	0.2	0.6
60	0.2	0	0.8	0.2	0.4	0
61	0.2	0.4	0.2	0.4	0.6	0.2
63	0.2	0.2	0	0.2	0.2	0.5
72	0	0.6	0.4	0.6	0.8	0.6
74	0	0.2	0.6	0.4	1	0.2
80	0	0.6	0.2	0	0.5	0.6
82	0	0.6	0.4	0.4	0.3	0.6
83	0.2	0.6	0	0.6	0.6	0.6
111	0.4	0.6	0.2	0.2	0.9	0
114	0.2	0.2	0.2	0.2	0.1	1
117	0	0.2	0	0.2	0.6	1
119	0.6	0.6	0	0.6	1	1
122	1	0.4	0.6	0.4	0.6	0.4
124	0	0.2	0.8	1	0.8	1

Tabla 3.2. Habilidad para programar.

id_proyecto \ Nivel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Android	0.2	1	0.1	0.2	0.5	0.2	0.1	0.2	0.2	1	0.1	0.2	0.5	0.2	0.1	0.2
Java	0.4	0.5	0.9	0.5	0.3	0.1	0.9	0.5	0.5	0.5	0.9	0.5	0.3	0.1	0.9	0.5
AngularJS	0.6	0.2	1	0.6	0.4	1	1	0.6	0.6	0.2	1	0.6	0.4	1	1	0.6
MySQL	0.1	0.3	0.5	0.8	1	0.5	0.5	0.8	0.8	0.3	0.5	0.8	1	0.5	0.5	0.8
PHP	0.5	0.5	0.6	0.9	0.8	0.2	0.6	0.9	0.2	0.5	0.6	0.9	0.8	0.2	0.6	0.9
JSF	0.7	0.9	0.8	1	0.5	0.6	0.8	1	0.4	0.9	0.8	1	0.5	0.6	0.8	1

Tabla 3.3. Habilidad requerida para programar en un proyecto.

id_empleado \ Nivel	Android	Java	AngularJS	MySQL	PHP	JSF
57	0.2	1	0	1	0.2	1
58	0.2	0.6	0.6	0.6	0.2	0.6
60	0.2	0	0.8	0.2	0.4	0
61	0.2	0.4	0.2	0.4	0.6	0.2
63	0.2	0.2	0	0.2	0.2	0.5
72	0	0.6	0.4	0.6	0.8	0.6
74	0	0.2	0.6	0.4	1	0.2
80	0	0.6	0.2	0	0.5	0.6
82	0	0.6	0.4	0.4	0.3	0.6
83	0.2	0.6	0	0.6	0.6	0.6
111	0.4	0.6	0.2	0.2	0.9	0
114	0.2	0.2	0.2	0.2	0.1	1
117	0	0.2	0	0.2	0.6	1
119	0.6	0.6	0	0.6	1	1
122	1	0.4	0.6	0.4	0.6	0.4
124	0	0.2	0.8	1	0.8	1

Tabla 3.4. Selección de la primer fila.

id_proyecto \ Nivel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Android	0.2	1	0.1	0.2	0.5	0.2	0.1	0.2	0.2	1	0.1	0.2	0.5	0.2	0.1	0.2
Java	0.4	0.5	0.9	0.5	0.3	0.1	0.9	0.5	0.5	0.5	0.9	0.5	0.3	0.1	0.9	0.5
AngularJS	0.6	0.2	1	0.6	0.4	1	1	0.6	0.6	0.2	1	0.6	0.4	1	1	0.6
MySQL	0.1	0.3	0.5	0.8	1	0.5	0.5	0.8	0.8	0.3	0.5	0.8	1	0.5	0.5	0.8
PHP	0.5	0.5	0.6	0.9	0.8	0.2	0.6	0.9	0.2	0.5	0.6	0.9	0.8	0.2	0.6	0.9
JSF	0.7	0.9	0.8	1	0.5	0.6	0.8	1	0.4	0.9	0.8	1	0.5	0.6	0.8	1

Tabla 3.5. Selección de la primer columna.

de estos dos conjuntos de datos; en la tabla 3.8 podemos observar los mínimos encontrados y en la tabla 3.7 el proceso de la composición max-min terminado.

Mínimos
0.2
0.4
0
0.1
0.2
0.7

Tabla 3.6. Mínimos obtenidos.

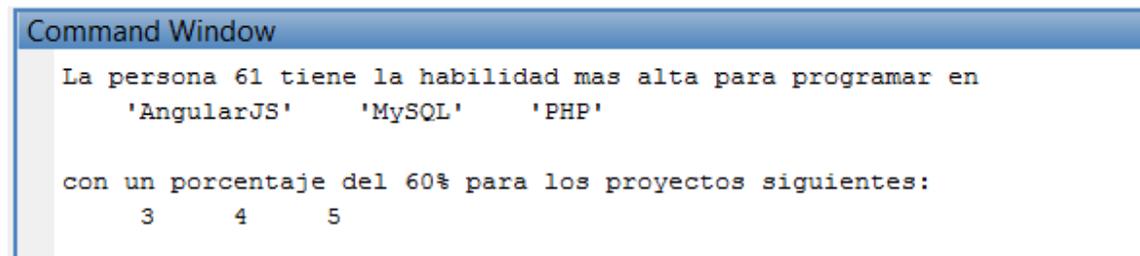
Máximo
0.7

Tabla 3.7. Composición max-min.

id_empleador \ id_proyecto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
57	0.7	0.9	0.9	1	1	6	9	1	8	9	9	1	1	0.6	0.9	1
58	0.6	0.6	0.6	0.8	1	0.6	0.6	0.8	0.8	0.6	0.6	0.8	1	0.6	0.6	0.8
60	0.6	0.4	0.8	0.6	0.4	0.8	0.8	0.6	0.6	0.4	0.8	0.6	0.4	0.8	0.8	0.6
61	0.5	0.5	0.6	0.6	0.6	0.4	0.6	0.6	0.4	0.5	0.6	0.6	0.6	0.4	0.6	0.6
63	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5
72	0.6	0.6	0.6	0.8	0.8	0.6	0.6	0.8	0.6	0.6	0.6	0.8	0.8	0.6	0.6	0.8
74	0.6	0.5	0.6	0.9	0.8	0.6	0.6	0.9	0.6	0.5	0.6	0.9	0.8	0.6	0.6	0.9
80	0.6	0.6	0.6	0.6	0.5	0.6	0.6	0.6	0.5	0.6	0.6	0.6	0.5	0.6	0.6	0.6
82	0.6	0.6	0.6	0.6	0.5	0.6	0.6	0.6	0.5	0.6	0.6	0.6	0.5	0.6	0.6	0.6
83	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
111	0.5	0.5	0.6	0.9	0.8	0.2	0.6	0.9	0.5	0.5	0.6	0.9	0.8	0.2	0.6	0.9
114	0.7	0.9	0.8	1	0.5	0.6	0.8	1	0.4	0.9	0.8	1	0.5	0.6	0.8	1
117	0.7	0.9	0.8	1	0.6	0.6	0.8	1	0.4	0.9	0.8	1	0.6	0.6	0.8	1
119	0.7	0.9	0.8	1	0.8	0.6	0.8	1	0.6	0.9	0.8	1	0.8	0.6	0.8	1
122	0.6	1	0.6	0.6	0.6	0.6	0.6	0.6	1	0.6	0.6	0.6	0.6	0.6	0.6	0.6
124	0.7	0.9	0.8	1	1	0.8	0.8	1	0.8	0.9	0.8	1	1	0.8	0.8	1

Tabla 3.8. Relación obtenida.

El resultado que se puede observar en la implementación del algoritmo de solución es una tabla resistente de los valores maximos y un enunciado que da la evaluación del personal esto lo podemos observar en la figura 3.1.



```
Command Window
La persona 61 tiene la habilidad mas alta para programar en
  'AngularJS'   'MySQL'   'PHP'

con un porcentaje del 60% para los proyectos siguientes:
  3     4     5
```

Figura 3.1. Resultados.

3.2.3. Diseño del algoritmo de solución

Algoritmo de solución

1. Inicio.
2. Cargar bases de datos a relacionar.
3. Comparar fila de la primer base de datos contra la columna de la segunda.
4. Extraer los mínimos de la comparación.
5. Extraer el máximo de los mínimos encontrados.
6. Almacenar máximos en matriz de relación.
7. En base a los datos por fila escoger el máximo(s).
8. Mostrar la asignación en base al máximo.

El fin de aplicar este algoritmo es encontrar a los mejores integrantes para el desarrollo de un proyecto tomando en cuenta algunas cualidades de cada uno en el desempeño de este trabajo, como por ejemplo sus habilidades de programación. Para este módulo se manejarán valores de 0 a 1 para valorar cada una de sus cualidades y poder hacer un evaluación; de manera que sea de utilidad para el líder de proyectos asignar a su personal a un proyecto y evitando errores de planeación y posibles retrasos en el desarrollo de las tareas asignadas a cada integrante.

3.2.4. Diseño de estructuras de datos

Para determinar el equipo idóneo en este módulo es necesario contar con estructuras de datos para el almacenamiento y procesamiento de los datos, las estructuras que se utilizarán serán matrices las cuales almacenarán las dos bases de datos a relacionarse, el resultado de aplicar el componente máx-mín y una más para guardar todos los proyectos en los que puede participar un integrante.

3.2.5. Diseño del medio de presentación

Para la presentación de resultados de aplicar el algoritmo se pretende mostrar el empleado evaluado para asignarlo a un proyecto, las habilidades con las que cuenta y podría colaborar, con su respectivo porcentaje de conocimiento y en los posibles proyectos que puede ser asignado.

3.3. Módulo patrones graduales frecuentes

3.3.1. Análisis y organización de las bases de datos referentes al desempeño de programadores y equipos de programadores

Para el modelado de un módulo que de solución al segundo problema descrito se propuso la programación en los lenguajes Scilab y Java donde se utilizaron datos sintéticos para realizar pruebas en el desarrollo del código, estos datos sintéticos se ilustran en la tabla 3.9 y 3.10.

Datos sintéticos utilizados como entrada para la primera prueba (3.9).

Temperatura	Velocidad viento	Humedad
20.2	3.16	0.76
19.5	2.2	0.66
12.7	5.12	0.89
12.2	5.5	0.25

Tabla 3.9. Base de datos sintética.

También se utilizó una tabla con transacciones describiendo 10 atributos numéricos, con datos sintéticos (3.10).

t \ A	A1	A	A3	A4	A5	A6	A7	A8	A9
t1	1	76	4	96	0	42	0	0.58	0.61
t2	5	0	48	25	72	85	36	0.99	0
t3	8	0	13	0	0	0	93	0.86	1
t4	79	0	75	0	31	0	83	0.16	0
t5	1	21	36	70	42	97	0	0.22	0.22
t6	63	0	0	86	77	0	19	0.55	0.18
t7	57	97	18	54	23	0	63	0	0.59
t8	80	82	86	91	0	81	96	0.65	0
t9	75	91	70	94	38	26	0	0.73	0

Tabla 3.10. Base de datos sintética.

Para la implementación en lenguaje Java se utilizan datos sintéticos muy parecidos a los reales manejados por la empresa en un archivo Excel el cual contiene datos de las tareas que se asignan a los programadores con cierto nivel de habilidades que se requiere para llevar a cabo la tarea valorado entre un nivel de 0 a 5, incluyendo también sus tiempos para cada fase de un proyecto.(Se muestra en la tabla 3.11)

Id perfil	Id actividad	Nivel WS SOAP	Backlog	Análisis	Diseño	Desarrollo	Pruebas	Tiempo total
57	1	0	20	15	37	15	87	174
74	2	0	95	43	78	225	42	483
82	3	0	41	98	118	142	14	413
58	4	0	28	26	35	25	114	228
82	5	0	18	105	32	529	25	709
62	6	0	16	48	55	86	27	232
61	7	1	9	38	14	91	79	231
62	8	0	16	42	59	52	18	187

Tabla 3.11. Base de datos sintética parecida a la real.

3.3.2. Selección del método de normalización de las bases de datos

Para el caso de esta investigación se aplicará la normalización basada en unidad de la cual se a hecho una descripción en el capítulo anterior, del cual se muestra un ejemplo enseguida aplicando la ecuación 2.1.

A continuación se detalla la normalización basada en unidad con un ejemplo práctico:

Temperatura	Velocidad viento	Humedad
20.2	3.16	0.76
19.5	2.2	0.66
12.7	5.12	0.89
12.2	5.5	0.25

Tabla 3.12. Base de datos sintética utilizada para ejemplo práctico.

En primer lugar se selecciona la columna a normalizar como se muestra en la tabla 3.13

Temperatura	Velocidad viento	Humedad
20.2	3.16	0.76
19.5	2.2	0.66
12.7	5.12	0.89
12.2	5.5	0.25

Tabla 3.13. Selección columna a normalizar.

Se localiza el valor mínimo y máximo de la columna seleccionada se muestra en la tabla 3.14

Tomando cada uno de los elementos de la columna y aplicando la 2.1 citada con anterioridad se obtiene el siguiente resultado (véase tabla 3.15):

$$x' = \frac{20,2 - 12,2}{20,2 - 12,2} = 1$$

$$x' = \frac{19,5 - 12,2}{20,2 - 12,2} = 0,91$$

Temperatura	Velocidad viento	Humedad
20.2	3.16	0.76
19.5	2.2	0.66
12.7	5.12	0.89
12.2	5.5	0.25

Tabla 3.14. Selección mínimo y máximo.

$$x' = \frac{12,7 - 12,2}{20,2 - 12,2} = 0,06$$

$$x' = \frac{12,2 - 12,2}{20,2 - 12,2} = 0$$

Temperatura	Velocidad viento	Humedad
1	3.16	0.76
0.91	2.2	0.66
0.06	5.12	0.89
0	5.5	0.25

Tabla 3.15. Columna normalizada.

Siguiendo con el mismo procedimiento para cada columna se obtiene la siguiente tabla normalizada (véase tabla 3.16):

Temperatura	Velocidad viento	Humedad
1	0.29	0.79
0.91	0	0.64
0.06	0.88	1
0	1	0

Tabla 3.16. Base de datos sintética normalizada.

3.3.3. Diseño del algoritmo de solución

La normalización consiste en transformar los datos reales en una escala de cero a uno, para este trabajo de investigación se utiliza el método de normalización

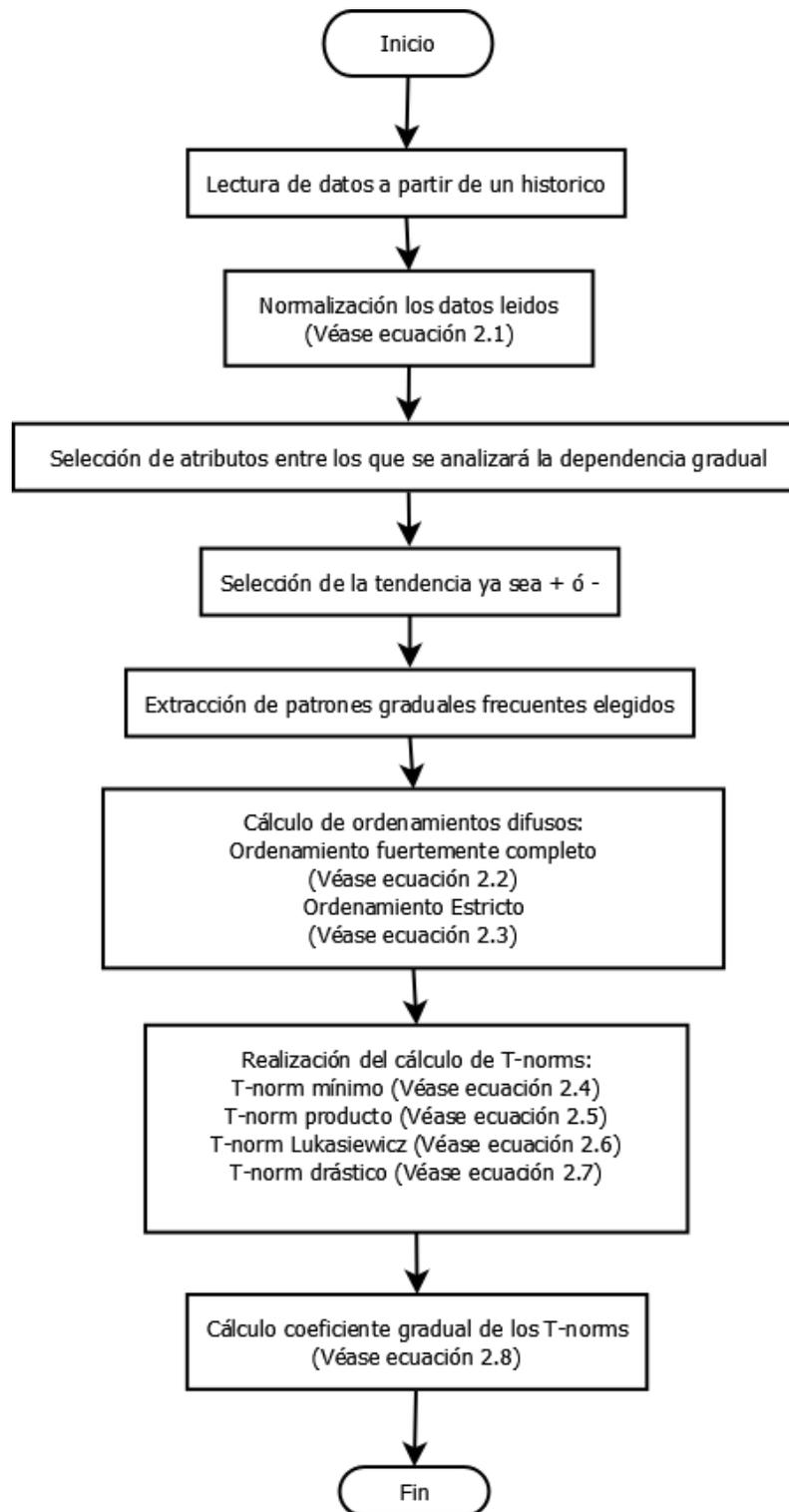


Figura 3.2. Algoritmo de solución.

basado en unidad, el cual consiste en tomar cada columna para encontrar de esta el número máximo y mínimo, para poder realizar la operación correspondiente con cada elemento que se encuentra en la columna, prosiguiendo a realizar este cálculo por cada columna que contenga la base de datos.

Se extraen las columnas con las cuales se realizará el proceso para obtener resultados que sean de utilidad para la dirección de la empresa.

Este algoritmo muestra cómo se realiza el cálculo de ordenamientos difusos teniendo en cuenta la extracción de los atributos entre los cuales se analizará la correlación, aplicando las formulas dadas en el diagrama de la figura 3.2 para cada ordenamiento.

- Ordenamiento fuertemente completo: se calcula la diferencia entre el atributo en la posición j y el atributo en la posición i , multiplicando el resultado por $1-(1/r)$, enseguida se calcula el valor máximo entre 0 y el resultado de la multiplicación, finalmente se calcula el valor mínimo entre 1 y el resultado anterior.
- Ordenamiento estricto: se calcula la diferencia entre 1 y el valor resultante del ordenamiento fuertemente completo en su posición i, j . Se calcula cada una de las T-norms y al final de cada una se realiza la suma de la matriz resultante dividiendo entre el número total de elementos sumados y finalmente se multiplica por 100 para obtener un porcentaje de correlación entre dos o más atributos.

3.3.4. Diseño de estructuras de datos

En el cálculo de patrones graduales frecuentes en la primera implementación desarrollada en Scilab, es necesario generar diferentes estructuras de datos para el almacenamiento de los datos que se obtendrán, la primer estructura que se utiliza es una matriz para almacenar los datos extraídos de un archivo con extensión .csv; en el caso de la normalización de la base de datos se requiere de una matriz del tamaño de los atributos que serán procesados; para el cálculo de los ordenamientos difusos se requiere de una matriz del tamaño de la longitud de la columna que se procesará,

y para el cálculo de los concordantes se requiere de una matriz del mismo tamaño que la del ordenamiento estricto.

Para el caso de la implementación realizada en Java se requiere de ciertas estructuras de datos, una lista de arreglos para el almacenamiento de la base de datos, una lista para los encabezados de los atributos de la base de datos, una lista para la extracción de los nombres de los atributos de los encabezados que serán procesados, una matriz para almacenar los datos normalizados, un vector para la extracción de las columnas de atributos con las que se realizará el cálculo de ordenamientos difusos, para los cuales se necesita de dos matrices de la longitud de cada atributo, una matriz para almacenar la concordancia entre los atributos, una lista para almacenar las diferentes combinaciones de los atributos, sus respectivas tendencias y su porcentaje de concordancia.

La estructura de la base de datos que se necesita durante la ejecución del programa para ambas versiones, debe contener más de una columna y más de una fila para poder realizar la correspondiente correlación entre estas, las columnas se denominan atributos o lista de atributos.

3.3.5. Diseño del medio de presentación

Para la presentación de resultados se pretende realizar la interpretación de la correlación entre dos o mas atributos con las respectivas combinaciones entre estos, según la tendencia que les ha sido asignada la cual puede ser de incremento o decremento para cada atributo. Por ejemplo, si se tienen cinco atributos:

- Tiempo Backlog
- Tiempo Análisis
- Tiempo Pruebas
- Tiempo Diseño
- Tiempo Desarrollo

Se pueden obtener algunas combinaciones como las siguientes, donde aumenta o disminuye representan la tendencia que se le aplica a cada atributo:

- Tiempo Backlog aumenta, Tiempo análisis aumenta
- Tiempo Backlog aumenta, Tiempo análisis disminuye
- Tiempo Backlog disminuye, Tiempo análisis aumenta
- Tiempo Backlog disminuye, Tiempo análisis disminuye

La salida presentará el porcentaje de concordancia obtenido entre cada una de las combinaciones que se realizan con los atributos obtenidos de la base de datos.

Capítulo 4

Resultados

4.1. Módulo estatus de un proyecto

4.1.1. Confección de los instrumentos y métricas de evaluación del módulo

Para la evaluación del módulo se desarrollaron instrumentos y métricas que consistieron en el uso de la programación de tareas de un proyecto con su respectivo equipo de trabajo y sus fechas planeadas para cada una de estas.

Con respecto al algoritmo de solución descrito el capítulo 3, dando seguimiento al desarrollo se describe como es que se aplicó paso a paso.

Primer paso.

Los datos ocupados para la evaluación del desempeño del módulo, consistió de las siguientes características:

- La base de datos dada debe tener un formato .csv para realizar la lectura desde el programa software desarrollado para el módulo de validación de proyectos.
- Se trata de un conjunto de datos correspondientes a una tabla que contiene los siguientes datos respecto a los proyectos en desarrollo:
 1. Tarea_ID
 2. Inicio_Planeado

3. Inicio_Real
4. Fin_Planeado
5. Fin_Real
6. ID_Responsable
7. Estatus

Segundo paso.

- Del conjunto de columnas con las que cuenta la tabla, se seleccionaron las que representan almacenamiento fechas, para comenzar el tiempo que necesita una tarea en su planeación y finalización, las columnas Tarea_ID y ID_Responsable sólo serán tomadas en cuenta para identificar la tarea y la persona a cargo de ella, la columna Estatus ayudará para saber hasta que punto calcular el valor ganado.

Tercer paso.

- Se realizó el cálculo en días entre la extracción de fechas realizada en el paso anterior.

Cuarto paso.

- Realizar el cálculo del valor ganado como se puede observar en la tabla 4.1, tomando en cuenta el tiempo estimado se le asigna un porcentaje a cada tarea para lograr ver al final el avance que se tiene dentro de todo el proyecto.

Después de obtener la tabla 4.1 se realiza una comparación por semana donde se ve reflejada la diferencia de tiempos ya sea a favor o en contra de lo planeado esto se puede observar en la tabla 4.2.

La métrica que se utilizó para evaluar el módulo fue un valor ganado el cual es calculado y es dentro de un rango de 0 a 100, esta métrica sirve para destacar el avance de un proyecto siendo de ayuda para el usuario final en la toma de decisiones.

Tarea ID	ID Responsable	Tiempo Estimado	Acumulado Tiempo Estimado	Semana Planeada	Valor Planeado	Acumulado Valor Planeado	Tiempo Actual	Semana Actual	Valor Ganado	Acumulado Valor Ganado
1	67	70	70	12	13.06	13.06	7	2	1.31	1.31
2	68	34	104	18	6.34	19.40	8	2	1.49	2.80
3	70	33	137	23	6.16	25.56	31	6	5.78	8.58
4	71	39	176	30	7.28	32.84	33	6	6.16	14.74
5	73	65	241	41	12.13	44.97	78	13	14.55	29.29
6	74	97	338	57	18.10	63.07	42	7	7.84	37.13
7	75	60	398	67	11.19	74.26	80	14	14.93	52.06
8	76	94	492	82	17.54	91.80	94	16		
9	77	0	492	82	0.00	91.80	1	1		
10	78	0	492	82	0.00	91.80	1	1		
11	80	44	536	90	8.21	100.00	30	5		
12	81	0	536	90	0.00	100.00	12	2		
13	82	0	536	90	0.00	100.00	10	2		
14	83	0	536	90	0.00	100.00	11	2		
15	85	0	536	90	0.00	100.00	2	1		
16	86	0	536	90	0.00	100.00	2	1		
17	87	0	536	90	0.00	100.00	1	1		

Tabla 4.1. Primer tabla para cálculo de valor ganado.

Semana	Tiempo estimado	Tiempo actual	Acumulado del tiempo actual	Acumulado valor planeado	Acumulado valor ganado
12	70	7	7	13.06	1.31
18	34	8	15	19.40	2.80
2	33	31	46	25.56	8.58
30	39	33	79	32.84	14.74
41	65	78	157	44.97	29.29
57	97	42	199	63.07	37.13
67	60	80	279	74.26	52.06
82	94	95	374	91.8	
90	44	67	441	100.00	

Tabla 4.2. Valoración por semana.

4.1.2. Diseño de experimentos para pruebas del módulo

La base de datos obtenida es preparada de manera que se puedan extraer los atributos a analizar, para realizar el cálculo de valores planeados y valores ganados. El programa está desarrollado para la construcción de dos tablas, para lograr identificar los avances de cada uno de los proyectos por tareas dando como resultado un porcentaje de avance comparado con el porcentaje de lo planeado, un porcentaje igual en ambos casos se interpreta que el proyecto va de acuerdo a la planeación en tiempo, si es menor quiere decir que se llevo menos tiempo del planeado, y si es mayor se interpreta como un desfase en la planeación.

Para lograr un análisis completo de una base de datos, que almacena los tiempos de duración en las tareas de desarrollo, se realizan los diferentes cálculos para obtener las tablas que se requieren.

4.1.3. Pruebas del algoritmo de valor ganado

El resultado que arrojo la implementación de dicho algoritmo son dos tablas que tienen representados los datos que se requieren para dar un resultado entendible y de ayuda en la toma de decisiones a la empresa con respecto a los proyectos que se encuentran en desarrollo, los datos que se desplegaban como se muestra en la figura 4.1, representan un dato concreto para que la empresa pueda interpretar para su beneficio.

4.1.4. Elaboración de reporte de interpretación de resultados

El interés de este trabajo consiste en identificar errores en la planeación de desarrollo en tareas, se considera para este trabajo seleccionar de un archivo de datos Excel, los atributos o columnas que representen los tiempos que deben durar y en caso de que una tarea éste terminada, el tiempo que duro la tarea en su desarrollo de acuerdo al tiempo asignado para su realización.

Con estas columnas se realizan las diferencias de fechas para así obtener los días que están programados para cada tarea, esto para lograr asignarle un porcentaje a cada tarea para su evaluación.

Tarea_ID	ID_Responsable	Tiempo Estimado	Acumulado	Semana Planeada	Valor Planeado	Acumulado	Valor Planeado			
1	67	70	70	12	13.06	13.06	7	2.	1.31	1.31
2	68	34	104	18	6.34	19.40	8	2.	1.49	2.80
3	70	33	137	23	6.16	25.56	31	6.	5.78	8.58
4	71	39	176	30	7.28	32.84	33	6.	6.16	14.74
5	73	65	241	41	12.13	44.97	78	13	14.55	29.29
6	74	97	338	57	18.10	63.07	42	7.	7.84	37.13
7	75	60	398	67	11.19	74.26	80	14	14.93	52.06
8	76	94	492	82	17.54	91.80	94	16		
9	77	0	492	82	0.00	91.80	1	1.		
10	78	0	492	82	0.00	91.80	1	1.		
11	80	44	536	90	8.21	100.01	30	5.		
12	81	0	536	90	0.00	100.01	12	2.		
13	82	0	536	90	0.00	100.01	10	2.		
14	83	0	536	90	0.00	100.01	11	2.		
15	85	0	536	90	0.00	100.01	2	1.		
16	86	0	536	90	0.00	100.01	2	1.		
17	87	0	536	90	0.00	100.01	1	1.		

Semana	Tiempo Estimado	Tiempo Actual	Acumulado de tiempo actuales	Acumulado Valor Planeado	Acumulado Valor Ganado
12	70	7	13.06	1.31	
18	34	8	19.40	2.80	
23	33	31	25.56	8.58	
30	39	33	32.84	14.74	
41	65	78	44.97	29.29	
57	97	42	199	63.07	37.13
67	60	80	279	74.26	52.06
82	94	95	374	91.8	
90	44	67	441	100.01	

Figura 4.1. Resultados.

La tabla 3.1 contiene información similar a la que será alimentada en la API desarrollada de la cual serán extraídos los datos necesarios.

Por ejemplo, la figura 4.2 muestra la tabla final obtenida en base a una evaluación semanal; la cual muestra un proceso de desarrollo menor al tiempo planeado, siendo un buen desarrollo y generando un histórico bueno para el equipo de trabajo involucrado en éste.

Semana	Tiempo Estimado	Tiempo Actual	Acumulado de tiempo actuales	Acumulado Valor Planeado	Acumulado Valor Ganado
12	70	7	13.06	1.31	
18	34	8	19.40	2.80	
23	33	31	25.56	8.58	
30	39	33	32.84	14.74	
41	65	78	44.97	29.29	
57	97	42	199	63.07	37.13
67	60	80	279	74.26	52.06
82	94	95	374	91.8	
90	44	67	441	100.01	

Figura 4.2. Evaluación del proyecto semanal.

Ahora es posible interpretar que para el ejemplo presentado en la figura 4.3 la evaluación es buena ya que el proyecto presenta ahorro en tiempo.

En la gráfica se puede observar que el tiempo actual está por debajo del planeado, dando un buen resultado para el equipo de trabajo en el módulo de factibilidad de equipo de trabajo.

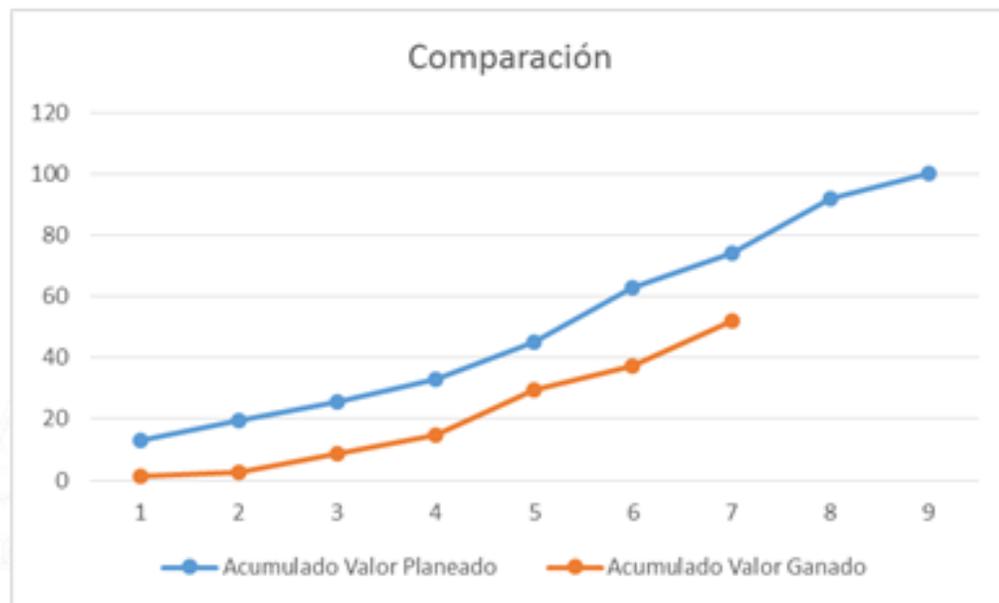


Figura 4.3. Gráfica de comparación.

4.2. Módulo difuso para la selección de un equipo idóneo

4.2.1. Confección de los instrumentos y métricas de evaluación del módulo

Para la evaluación del módulo se desarrollaron instrumentos y métricas que consisten en evaluar la cualidades de cada integrante que puede ser candidato para el desarrollo de un proyecto. Con respecto al algoritmo de solución presentado en el capítulo 3, se debe cumplir con ciertas características para poder procesar la información. Para poder crear relaciones difusas debemos contar con dos matrices las cuales deberán contener valores numéricos entre 0 y 1 para su procesamiento; para generar una relación difusa estas dos matrices deberán compartir una serie de atributos para poder realizarla. Los datos que se muestran son sintéticos en su totalidad solo tomando en cuenta algunas de las disciplinas con las que dispone la empresa para la evaluación de su personal. La métrica que se utilizó para evaluar el módulo fue composición máx-mín el cual se calcula y como resultado muestra un

porcentaje en el que se domina la cualidad, lo cual es de ayuda para poder determinar si una persona es apta para participar en el desarrollo de un proyecto dado con ciertos requerimientos.

4.2.2. Diseño de experimentos para pruebas del módulo

El programa desarrollo esta diseñado para realizar relaciones difusas a partir de dos bases de datos ya preparadas antes de ser leídas directamente en el programa, estas relaciones son calculadas a base del componente máx-mín, con el fin de visualizar la evaluación de un prospecto para asinación en un proyecto de desarrollo dando como salida su habilidad, porcentaje y posible proyecto a ser asignado.

4.2.3. Pruebas del algoritmo

Se tomaron tres factores para evaluar a un empleado los cuales fueron habilidad para programar, cumplimiento en sus tareas en fechas estipuladas y puntualidad de llegada al trabajo. Pues son factores importantes para que un proyecto llegue a un fin exitoso para realizar las pruebas se tomaron 3 pares de bases de datos diferentes obteniendo un resultado para cada factor. En las tablas 4.3 y 4.4 se puede observar un par de estas.

Por cada relación difusa entre par de bases de datos se genera una matriz de máximos que podemos observar en la figura 4.4.

Y al final se puede ver las posibles asignaciones por persona de acuerdo al factor evaluado las figuras 4.5 y 4.6 muestran estos resultados.

Para dar una sugerencia más exacta se realizara una comparativa con la composición min-max con la finalidad de dar el resultado mas exacto para lograr el objetivo que es la selección de los mejores integrantes para el desarrollo de un proyecto teniendo en cuenta lo necesario para poder participar en este.

Para la composición max-min tenemos lo siguientes resultados aplicando los tres factores mencionados con anterioridad que se pueden observar en las figuras 4.7, 4.8 y 4.9.

Para la composición min-max se obtuvieron los siguientes resultados que se pueden visualizar en las figuras 4.10, 4.11 y 4.12.

Cumplimiento id_empleado	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Tarea 6
57	0.7	0.9	0.9	1	1	0.6
58	0.6	0.6	0.6	0.8	1	0.6
60	0.6	0.4	0.8	0.6	0.4	0.8
61	0.5	0.5	0.5	0.6	0.6	0.4
63	0.5	0.5	0.5	0.5	0.5	0.5
72	0.6	0.6	0.6	0.8	0.8	0.6
74	0.6	0.5	0.6	0.9	0.8	0.6
80	0.6	0.6	0.6	0.6	0.5	0.6
82	0.6	0.6	0.6	0.6	0.5	0.6
83	0.6	0.6	0.6	0.6	0.6	0.6
111	0.5	0.5	0.6	0.9	0.8	0.2
114	0.7	0.9	0.8	1	0.5	0.6
117	0.7	0.9	0.8	1	0.6	0.6
119	0.7	0.9	0.8	1	0.8	0.6
122	0.6	1	0.6	0.6	0.6	0.6
124	0.7	0.9	0.8	1	1	0.8

Tabla 4.3. Cumplimiento de tareas.

Cumplimiento id_proyecto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Tarea 1	0.5	0.5	0.6	0.9	0.8	0.2	0.6	0.9	0.5	0.5	0.6	0.9	0.8	0.2	0.6	0.9
Tarea 2	0.7	0.9	0.8	1	0.5	0.6	0.8	1	0.4	0.9	0.8	1	0.5	0.6	0.8	1
Tarea 3	0.7	0.9	0.8	1	0.6	0.6	0.8	1	0.4	0.9	0.8	1	0.6	0.6	0.8	1
Tarea 4	0.7	0.9	0.8	1	0.8	0.6	0.8	1	0.6	0.9	0.8	1	0.8	0.6	0.8	1
Tarea 5	0.6	1	0.6	0.6	0.6	0.6	0.6	0.6	0.6	1	0.6	0.6	0.6	0.6	0.6	0.6
Tarea 6	0.7	0.9	0.8	1	1	0.8	0.8	1	0.8	0.9	0.8	1	1	0.8	0.8	1

Tabla 4.4. Cumplimiento requerido para el proyecto.

Command Window

Columns 1 through 12

0.7000	0.9000	0.9000	1.0000	1.0000	0.6000	0.9000	1.0000	0.8000	0.9000	0.9000	1.0000
0.6000	0.6000	0.6000	0.8000	1.0000	0.6000	0.6000	0.8000	0.8000	0.6000	0.6000	0.8000
0.6000	0.4000	0.8000	0.6000	0.4000	0.8000	0.8000	0.6000	0.6000	0.4000	0.8000	0.6000
0.5000	0.5000	0.6000	0.6000	0.6000	0.4000	0.6000	0.6000	0.4000	0.5000	0.6000	0.6000
0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.4000	0.5000	0.5000	0.5000
0.6000	0.6000	0.6000	0.8000	0.8000	0.6000	0.6000	0.8000	0.6000	0.6000	0.6000	0.8000
0.6000	0.5000	0.6000	0.9000	0.8000	0.6000	0.6000	0.9000	0.6000	0.5000	0.6000	0.9000
0.6000	0.6000	0.6000	0.6000	0.5000	0.6000	0.6000	0.6000	0.5000	0.6000	0.6000	0.6000
0.6000	0.6000	0.6000	0.6000	0.5000	0.6000	0.6000	0.6000	0.5000	0.6000	0.6000	0.6000
0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000
0.5000	0.5000	0.6000	0.9000	0.8000	0.2000	0.6000	0.9000	0.5000	0.5000	0.6000	0.9000
0.7000	0.9000	0.8000	1.0000	0.5000	0.6000	0.8000	1.0000	0.4000	0.9000	0.8000	1.0000
0.7000	0.9000	0.8000	1.0000	0.6000	0.6000	0.8000	1.0000	0.4000	0.9000	0.8000	1.0000
0.7000	0.9000	0.8000	1.0000	0.8000	0.6000	0.8000	1.0000	0.6000	0.9000	0.8000	1.0000
0.6000	1.0000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	1.0000	0.6000	0.6000
0.7000	0.9000	0.8000	1.0000	1.0000	0.8000	0.8000	1.0000	0.8000	0.9000	0.8000	1.0000

Columns 13 through 16

1.0000	0.6000	0.9000	1.0000
1.0000	0.6000	0.6000	0.8000
0.4000	0.8000	0.8000	0.6000
0.6000	0.4000	0.6000	0.6000
0.5000	0.5000	0.5000	0.5000
0.8000	0.6000	0.6000	0.8000
0.8000	0.6000	0.6000	0.9000
0.5000	0.6000	0.6000	0.6000
0.5000	0.6000	0.6000	0.6000
0.6000	0.6000	0.6000	0.6000
0.8000	0.2000	0.6000	0.9000
0.5000	0.6000	0.8000	1.0000

Figura 4.4. Máximos.

Command Window

La persona 74 tiene el cumplimiento mas alto para la

'Tarea 2' 'Tarea 4'

con un porcentaje del 90% para los proyectos siguientes:

2 4

Figura 4.5. Asignación por cumplimiento de tareas

Command Window

La persona 74 tiene una puntualidad mas alta los dias

'Martes' 'Miercoles' 'Jueves' 'Viernes' 'Sabado'

con un porcentaje del 60% para los proyectos siguientes:

2 3 4 5 6

Figura 4.6. Asignación por puntualidad

```

Command Window
La persona 58 tiene la habilidad mas alta para programar en
'Java' 'AngularJS' 'MySQL' 'JSF'
con un porcentaje del 60% para los proyectos siguientes:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
fx |

Command Window
La persona 61 tiene la habilidad mas alta para programar en
'PHP'
con un porcentaje del 60% para los proyectos siguientes:
3 4 5 7 8 11 12 13 15 16
fx |

Command Window
La persona 124 tiene la habilidad mas alta para programar en
'MySQL' 'JSF'
con un porcentaje del 100% para los proyectos siguientes:
4 5 8 12 13 16
fx |
    
```

Figura 4.7. Resultado para asignación por habilidad

```

Command Window
La persona 57 tiene el cumplimiento mas alto para la
'Tarea 4' 'Tarea 5'
con un porcentaje del 100% para los proyectos siguientes:
2 4
fx |

Command Window
La persona 60 tiene el cumplimiento mas alto para la
'Tarea 3' 'Tarea 6'
con un porcentaje del 80% para los proyectos siguientes:
2 3 4 5 6
fx |

Command Window
La persona 63 tiene el cumplimiento mas alto para la
'Tarea 1' 'Tarea 2' 'Tarea 3' 'Tarea 4' 'Tarea 5' 'Tarea 6'
con un porcentaje del 50% para los proyectos siguientes:
1 2 3 4 5 6
fx |
    
```

Figura 4.8. Resultado para asignación por cumplimiento de tareas

```

Command Window
La persona 58 tiene una puntualidad mas alta los dias
'Lunes' 'Martes' 'Miercoles' 'Jueves' 'Viernes' 'Sabado'
con un porcentaje del 50% para los proyectos siguientes:
 2   3   4   5   6
fx |

Command Window
La persona 63 tiene una puntualidad mas alta los dias
'Lunes' 'Martes' 'Miercoles' 'Jueves' 'Viernes' 'Sabado'
con un porcentaje del 60% para los proyectos siguientes:
 2   3   4   5   6
fx |

Command Window
La persona 114 tiene una puntualidad mas alta los dias
'Jueves' 'Viernes'
con un porcentaje del 80% para los proyectos siguientes:
 5
fx |
    
```

Figura 4.9. Resultado para asignación por puntualidad

```

Command Window
La persona 58 tiene la habilidad mas alta para programar en
'Java' 'AngularJS' 'MySQL' 'JSF'
con un porcentaje del 60% para los proyectos siguientes:
 2   5   10  13
fx |

Command Window
La persona 61 tiene la habilidad mas alta para programar en
'PHP'
con un porcentaje del 60% para los proyectos siguientes:
 5   13
fx |

Command Window
La persona 124 tiene la habilidad mas alta para programar en
'MySQL' 'JSF'
con un porcentaje del 100% para los proyectos siguientes:
 2   10
fx |
    
```

Figura 4.10. Resultado para asignación por habilidad de programación

```

Command Window
La persona 57 tiene el cumplimiento mas alto para la
'Tarea 4' 'Tarea 5'
con un porcentaje del 100% para los proyectos siguientes:
  4
fx |

Command Window
La persona 60 tiene el cumplimiento mas alto para la
'Tarea 3' 'Tarea 6'
con un porcentaje del 80% para los proyectos siguientes:
  1  2  3  4  6
fx |

Command Window
La persona 63 tiene el cumplimiento mas alto para la
'Tarea 1' 'Tarea 2' 'Tarea 3' 'Tarea 4' 'Tarea 5' 'Tarea 6'
con un porcentaje del 50% para los proyectos siguientes:
  3  4
fx |
    
```

Figura 4.11. Resultado para asignación por cumplimiento de tareas

```

Command Window
La persona 58 tiene una puntualidad mas alta los dias
'Lunes' 'Martes' 'Miercoles' 'Jueves' 'Viernes' 'Sabado'
con un porcentaje del 50% para los proyectos siguientes:
  1  2  3  4  5  6
fx |

Command Window
La persona 63 tiene una puntualidad mas alta los dias
'Lunes' 'Martes' 'Miercoles' 'Jueves' 'Viernes' 'Sabado'
con un porcentaje del 60% para los proyectos siguientes:
  1  2  3  4  5  6
fx |

Command Window
La persona 114 tiene una puntualidad mas alta los dias
'Jueves' 'Viernes'
con un porcentaje del 80% para los proyectos siguientes:
  1  2  3  4  5  6
fx |
    
```

Figura 4.12. Resultado para asignación por puntualidad

Se logra observar en las figuras que se tiene un mismo porcentaje de exactitud pero en la composición min-max da menos posibles proyectos a los cuales puede ser asignado un empleado.

4.3. Módulo patrones graduales frecuentes

4.3.1. Confección de los instrumentos y métricas de evaluación del módulo

Para la evaluación del módulo se desarrollaron instrumentos y métricas que consistieron en el uso de un histórico que contiene las habilidades de cada programador, así como los tiempos que requirió cada programador para el desarrollo de las diferentes fases en las que participaron en el desarrollo de diferentes proyectos.

El histórico que fue ocupado para la evaluación del desempeño del módulo, consistió de las siguientes características:

- Tratándose de un conjunto de datos de 61 atributos y 51 registros, donde los atributos se refieren a las habilidades de cada programador además de las fases que conforman un proyecto.
- Del conjunto de columnas con las que cuenta el histórico, se seleccionaron las columnas que indican tiempo las cuales deberán ser indicadas por los caracteres “T_” en el encabezado para que puedan ser identificadas las columnas de interés pasando por alto las demás, esto es debido a que para la empresa le es de gran dificultad filtrar columnas de la sección de datos que obtienen.
- La base de datos dada debe tener un formato .csv para realizar la lectura desde el programa software desarrollado para el módulo de patrones graduales frecuentes.

La razón por la cual sólo se ocuparon ciertos atributos de este histórico, es porque se requiere analizar solo los tiempos que utiliza un programador para llevar a cabo una tarea, así como los patrones graduales frecuentes que se enfocaron a tiempos en desarrollo de un programador, el histórico además de contar con datos de tiempo, contiene información de las habilidades que tiene cada desarrollador, midiendo su nivel en una escala de 0 al 5, donde el cero es que no tiene conocimientos de la habilidad y el 5 refleja un conocimiento avanzado. En la figura 4.13 se muestra un extracto de la base de datos que se manejó para las pruebas de ejecución.

id-perfil	id-actividad	nivel-.NET	nivel-Adobe-illustrator	nivel-Adobe-InDesign	nivel-Analysis	nivel-Android	Backlog	Analisis	Diseño	Desarro	Pruebas	Tiempo-Total
57	1	0	0	0	1	0	20	15	37	15	87	174
74	2	0	0	0	0	0	95	43	78	225	42	483
82	3	0	0	0	0	0	41	98	118	142	14	413
58	4	0	0	0	1	0	28	26	35	25	114	228
82	5	0	0	0	0	0	18	105	32	529	25	709
62	6	0	0	0	1	0	16	48	55	86	27	232
61	7	0	0	0	0	0	9	38	14	91	79	231
62	8	0	0	0	1	0	16	42	59	52	18	187
82	9	0	0	0	0	0	40	63	133	53	17	306
111	10	0	0	0	0	0	21	103	35	116	122	397
114	11	0	0	0	0	0	81	99	45	472	192	889
60	12	0	0	0	1	0	23	18	28	19	88	176
114	13	0	0	0	0	0	12	57	21	87	14	191
65	14	0	0	0	0	0	2	16	26	53	19	116
83	15	0	0	0	0	0	39	77	90	60	20	286
114	16	0	0	0	0	0	2	68	55	259	323	707
74	17	0	0	0	0	0	80	34	30	438	40	622
72	18	0	0	0	0	0	55	23	43	517	51	689
57	19	0	0	0	2	0	25	25	33	20	103	206
84	20	0	0	0	0	0	37	62	97	465	20	681
80	21	0	0	0	0	0	78	22	120	204	40	464
83	22	0	0	0	0	0	35	69	91	444	42	681
91	23	0	0	0	0	0	88	18	10	32	16	164

Figura 4.13. Extracto de la base de datos utilizada para pruebas experimentales.

La aplicación de métricas consiste en la aplicación continua de mediciones en el proceso de evaluación de software y sus productos, para suministrar información relevante a tiempo.

Las métricas que pueden aplicarse se encuentran clasificadas según los criterios siguientes:

- De complejidad: métrica que define la medición de la complejidad: volumen, tamaño, anidaciones, y configuración.
- De calidad: métrica que define la calidad del software: exactitud, estructuración o modularidad, pruebas, y mantenimiento.
- De competencia: métrica que intenta valorar o medir las actividades de productividad de los programadores con respecto a su certeza, rapidez, eficiencia y competencia.
- De desempeño: métrica que mide la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware.
- Estilizadas: métrica de experimentación y de preferencia: estilo de código, convenciones, limitaciones, etc.

La métrica que se utilizó para evaluar el módulo fue un umbral que deberá ser dado por el usuario siendo el 10 el valor mínimo, el umbral contendrá la frecuencia que se tiene en el histórico a procesar, esta métrica sirve para destacar las opciones con mayor frecuencia siendo de ayuda para el usuario final.

Se busca que los resultados que da como salida el programa tengan sentido de manera que sean de apoyo y utilidad en la toma de decisiones de asignación y planeación de tiempos de desarrollo.

4.3.2. Diseño de experimentos para pruebas del módulo

La base de datos obtenida es preparada de manera que se puedan extraer los atributos a analizar, para realizar el cálculo de concordantes graduales entre ellos. El programa está desarrollado para realizar correlaciones desde dos hasta n número de atributos, para lograr identificar los casos que se encuentren por encima de cierto umbral de porcentaje el cual se ha establecido con un valor de diez pudiendo ser modificado por el usuario.

Para lograr un análisis completo de una base de datos histórica, que almacena los tiempos de duración en las fases de desarrollo, se realizan las diferentes combinaciones entre los atributos y tendencias con que cuenta dicha base de datos, modificando en repetidas ocasiones el umbral dando como resultado una filtración de las correlaciones más frecuentes con mayor nivel de confianza cuando se incrementa el valor del umbral.

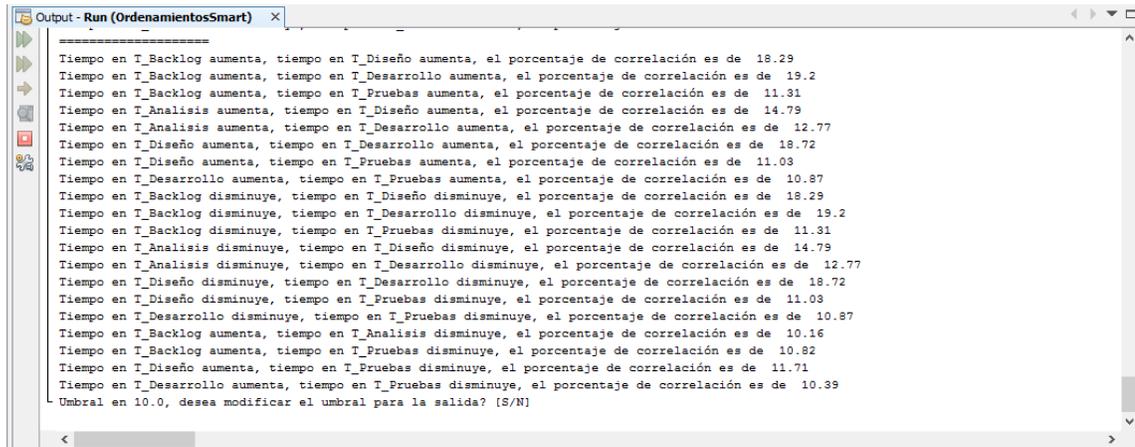
4.3.3. Pruebas del algoritmo

El primer resultado del programa que se presentó a la empresa, generó debate entre la empresa y los colaboradores del proyecto, puesto que los datos que se desplegaban como se muestra en la figura 4.14, no representaban un dato concreto que la empresa pudiera interpretar para su beneficio, por lo que fue necesario modificar la interpretación de la salida.

```
=====
Cuando la fase Backlog tiene un incremento en horas, la fase Diseño presenta incremento en horas, el porcentaje de correlación es de 19.64
Cuando la fase Analisis tiene un incremento en horas, la fase Diseño presenta incremento en horas, el porcentaje de correlación es de 16.28
Cuando la fase Backlog tiene un decremento en horas, la fase Diseño presenta decremento, el porcentaje de correlación es de 19.64
Cuando la fase Analisis tiene un decremento en horas, la fase Diseño presenta decremento, el porcentaje de correlación es de 16.28
-----
```

Figura 4.14. Interpretación de la salida.

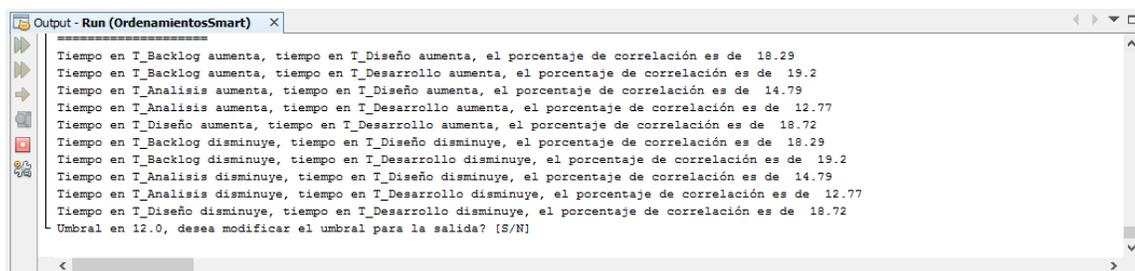
Dado que en un principio, el umbral estaba definido con un valor fijo, también se requirió que esté pudiera ser modificado por el usuario como se muestra en la figura 4.15.



```
Output - Run (OrdenamientosSmart) x
-----
Tiempo en T_Backlog aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 19.2
Tiempo en T_Backlog aumenta, tiempo en T_Pruebas aumenta, el porcentaje de correlación es de 11.31
Tiempo en T_Análisis aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 14.79
Tiempo en T_Análisis aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 12.77
Tiempo en T_Diseño aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 18.72
Tiempo en T_Diseño aumenta, tiempo en T_Pruebas aumenta, el porcentaje de correlación es de 11.03
Tiempo en T_Desarrollo aumenta, tiempo en T_Pruebas aumenta, el porcentaje de correlación es de 10.87
Tiempo en T_Desarrollo disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 19.2
Tiempo en T_Backlog disminuye, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 11.31
Tiempo en T_Análisis disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 14.79
Tiempo en T_Análisis disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 12.77
Tiempo en T_Diseño disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 18.72
Tiempo en T_Diseño disminuye, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 11.03
Tiempo en T_Desarrollo disminuye, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 10.87
Tiempo en T_Backlog aumenta, tiempo en T_Análisis disminuye, el porcentaje de correlación es de 10.16
Tiempo en T_Backlog aumenta, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 10.82
Tiempo en T_Diseño aumenta, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 11.71
Tiempo en T_Desarrollo aumenta, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 10.39
Umbral en 10.0, desea modificar el umbral para la salida? [S/N]
```

Figura 4.15. Interpretación de la salida (primera modificación).

Una vez que el programa da la flexibilidad de elegir el umbral que se aplica a los datos de salida, es posible filtrar los datos con un mayor nivel de confianza. También se modificó el enunciado de la salida para una mejor interpretación, figura 4.16 y 4.17.



```
Output - Run (OrdenamientosSmart) x
-----
Tiempo en T_Backlog aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 19.2
Tiempo en T_Análisis aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 14.79
Tiempo en T_Análisis aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 12.77
Tiempo en T_Diseño aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 18.72
Tiempo en T_Backlog disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 19.2
Tiempo en T_Análisis disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 14.79
Tiempo en T_Análisis disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 12.77
Tiempo en T_Diseño disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 18.72
Umbral en 12.0, desea modificar el umbral para la salida? [S/N]
```

Figura 4.16. Interpretación de la salida (segunda modificación).

Finalmente, la empresa requiere que los datos que se muestran en pantalla, además de ser desplegados, puedan ser almacenados de manera que para su propósitos, puedan ser graficados al momento de hacer la implementación, por lo que la salida en forma de enunciado no es un dato válido para graficar, de esta manera

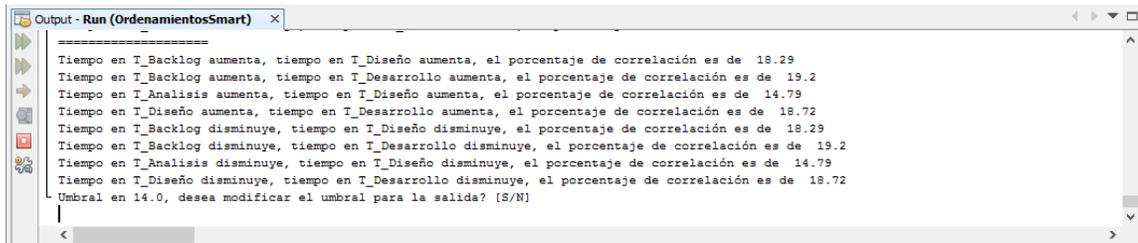


Figura 4.17. Interpretación de la salida (segunda modificación mostrando diferente umbral).

se crea una lista de resultados con los valores disponibles para graficar (véase figura 4.18).

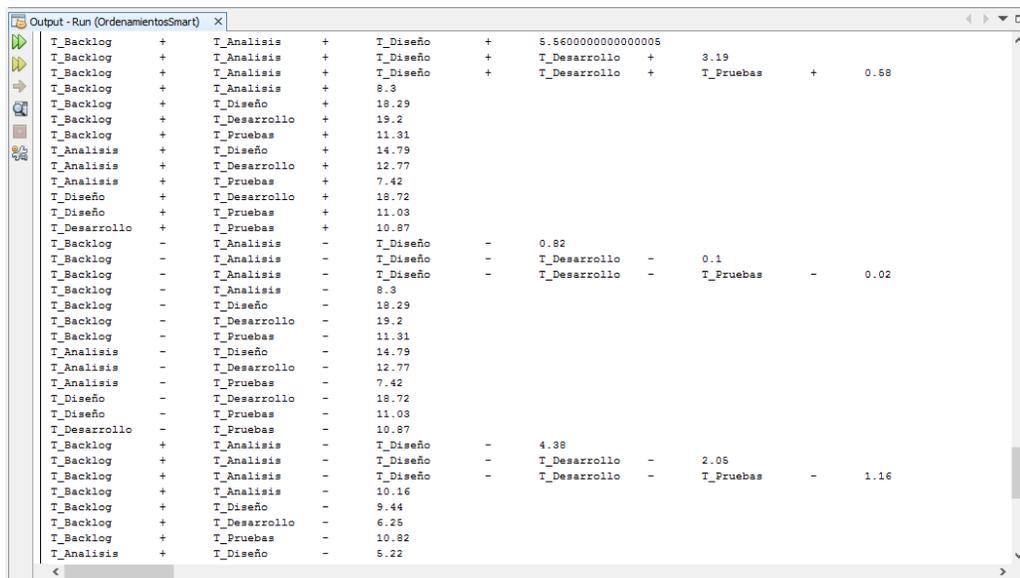


Figura 4.18. Lista de atributos con resultados para visualización.

4.3.4. Reporte de interpretación de resultados

El uso de las tecnologías conlleva en primer lugar a la recolección de datos, lo que a través del tiempo genera grandes volúmenes de información, la cual al ser procesada con patrones graduales frecuentes se genera un porcentaje de concordancia, en segundo lugar se busca encontrar dentro del conjunto de datos, los patrones que

aparecen de forma frecuente haciendo de estos grandes volúmenes de información un filtrado quedando sólo la información más relevante.

El interés de este trabajo consiste en identificar las tendencias en términos de correlaciones, se considera para este trabajo seleccionar de un archivo de datos Excel, los atributos o columnas que representen los tiempos que han durado las fases en su desarrollo para proyectos anteriores.

T_Backlog	T_Análisis	T_Diseño	T_Desarrollo	T_Pruebas
20	15	37	15	87
95	43	78	225	42
41	98	118	142	14
28	26	35	25	114
18	105	32	529	25
16	48	55	86	27
9	38	14	91	79

Tabla 4.5. Columnas o atributos de la base de datos.

Por ejemplo, dos atributos Análisis y Diseño, se denomina patrón frecuente a las siguientes ocurrencias dentro de la base de datos:

{un aumento de tiempo en el Análisis, es acompañado de un aumento en el tiempo de Diseño}

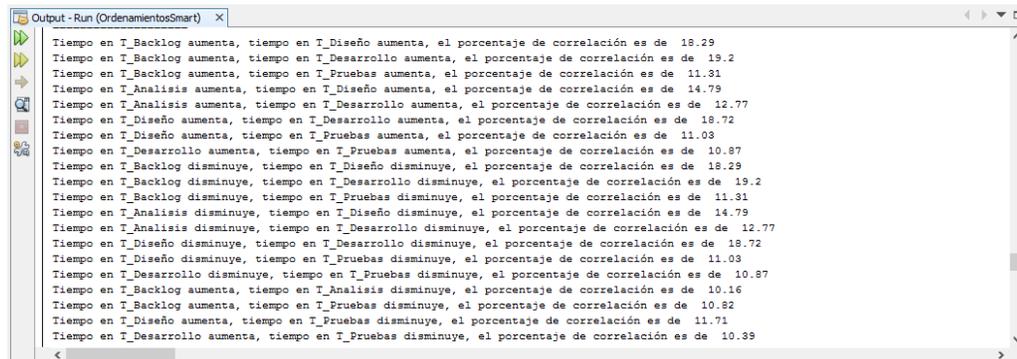
{un aumento de tiempo en el Análisis, es acompañado de una disminución en el tiempo de Diseño}

{una disminución en el tiempo de Análisis, es acompañada de un aumento en el tiempo de Diseño}

{una disminución en el tiempo de Análisis, es acompañada de una disminución en el tiempo de Diseño}

La tabla 4.5 contiene información similar a la almacenada en la base de datos de la cual serán extraídos los atributos cuyo nombre inicie con “T_”.

Por ejemplo, analizando una muestra de la base de datos histórica, se conoce que cuando el tiempo en Backlog aumenta, el tiempo en Diseño aumenta existe un nivel de soporte del 18.29%. La figura 43 muestra la lista de patrones graduales frecuentes de los datos de la tabla 4.5.



```
Output - Run (OrdenamientosSmart) x
Tiempo en T_Backlog aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 19.2
Tiempo en T_Backlog aumenta, tiempo en T_Pruebas aumenta, el porcentaje de correlación es de 11.31
Tiempo en T_Analisis aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 14.79
Tiempo en T_Analisis aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 12.77
Tiempo en T_Diseño aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 18.72
Tiempo en T_Diseño aumenta, tiempo en T_Pruebas aumenta, el porcentaje de correlación es de 11.03
Tiempo en T_Desarrollo aumenta, tiempo en T_Pruebas aumenta, el porcentaje de correlación es de 10.87
Tiempo en T_Backlog disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 19.2
Tiempo en T_Backlog disminuye, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 11.31
Tiempo en T_Analisis disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 14.79
Tiempo en T_Analisis disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 12.77
Tiempo en T_Diseño disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 18.72
Tiempo en T_Diseño disminuye, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 11.03
Tiempo en T_Desarrollo disminuye, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 10.87
Tiempo en T_Backlog aumenta, tiempo en T_Analisis disminuye, el porcentaje de correlación es de 10.16
Tiempo en T_Backlog aumenta, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 10.82
Tiempo en T_Diseño aumenta, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 11.71
Tiempo en T_Desarrollo aumenta, tiempo en T_Pruebas disminuye, el porcentaje de correlación es de 10.39
```

Figura 4.19. Lista de porcentaje en patrones graduales frecuentes.

Ahora es posible interpretar que un aumento en tiempo en Backlog, el tiempo en Diseño aumenta, con un porcentaje de soporte del 18.29%.

Capítulo 5

Conclusiones

En el primer capítulo se dio a conocer el problema, justificación, los objetivos del desarrollo de la tesis, y el estado del arte es decir algunas herramientas existentes que realizan tareas similares al desarrollo de esta investigación. En el segundo capítulo se podrá leer un poco de las herramientas de análisis y desarrollos que se ocuparon durante la investigación para llegar a la generación del resultado deseado. Y algunas de las metodologías como Kanban y Just In Time usadas por la empresa con la que se colaboró. En el tercer capítulo se realizó el modelado de solución para las problemáticas descritas en el capítulo 1 con los requerimientos que hizo saber la empresa la cual proporcionó visualmente un ejemplo de como se trabaja su información con la finalidad de poder adaptar el desarrollo a la necesidad de la organización. En el quinto capítulo se da a conocer los resultados experimentales que se obtuvieron de la investigación con datos sintéticos, debido a que la información de la organización es confidencial.

En este trabajo se ha mostrado tan solo una aplicación del uso de la extracción de patrones graduales frecuentes como una herramienta de conocimiento e interpretación para la toma de decisiones en el desarrollo de proyectos. Existen diferentes algoritmos que han sido propuestos para descubrir tales patrones.

Para el estudio realizado, se concluye que la relación que existe entre una fase de desarrollo de un proyecto, se encuentra relacionada con las fases posteriores, de tal manera que el incremento o decremento del tiempo que se invierte en el desarrollo de una fase, influye en el desarrollo de la fase posterior. Este resultado puede

considerarse como valioso en futuras planificaciones para desarrollo de proyectos, pues el tiempo que se invierta a una fase modificará el tiempo de desempeño de las fases siguientes.

De igual manera se presentó la propuesta de solución e implementación de una API, la cual cumplió con los requisitos planteados al inicio de su desarrollo, así como con las necesidades de los clientes, las cuales se tomaron a partir del levantamiento de requerimientos y de las encuestas realizadas. El sistema facilita la toma de decisiones para los líderes de proyecto dando opciones para lograr el objetivo propuesto durante la planeación del proyecto a desarrollar, dando cumplimiento a sus contratos en el desarrollo de software a la medida de sus clientes.

También se presentó un cuadro comparativo, cuadro 1, en el cual se puede observar que existe gran variedad de software para la administración de proyectos. Es cierto que estas herramientas son muy completas pero en ocasiones las empresas no pueden acoplarse a la manera de trabajo con estas, ya sea por su diseño gráfico poco intuitivo o por falta de herramientas que ayuden a controlar un proyecto en ejecución. La herramienta desarrollada cuenta con el cálculo del método de valor planeado y valor ganado.

Los módulos fueron diseñados para complementar una herramienta de gestión de proyectos, la cual ha sido diseñada a medida de las necesidades de una empresa, este complemento fue desarrollado para mejorar el flujo de ejecución de tareas dando como ayuda al líder de proyecto tomar decisiones en el momento que se requiera para evitar incurrir en fechas de entrega posteriores a las pactadas con sus clientes.

Para el módulo difuso para la selección de un equipo idóneo se puede concluir que se logró el desarrollo para proponer la asignación de acuerdo a tres factores dando al usuario final la oportunidad de crear el mejor equipo para un proyecto de acuerdo a los requerimientos que este tiene. En el módulo se realizaron pruebas con dos composiciones de las cuales se recomienda usar la composición max-min debido a que da más opciones a cada integrante de asignarlo a un proyecto, ya que el otro método empleado reduce los campos en los que se puede desempeñar un empleado.

Apéndice A

Artículos

IMPLEMENTACIÓN DE UNA API PARA EL SEGUIMIENTO DE PROYECTOS BASADOS EN LAS METODOLOGÍAS JUST IN TIME Y KANBAN

Ing. Lorena Hernández Carmona¹, Lic. Laura Reyes Briones²,
Dr. José Crispín Hernández Hernández³ y Dr. Edmundo Bonilla Huerta⁴

Resumen—En la actualidad, las empresas de desarrollo de software se interesan en el seguimiento del avance de sus proyectos, para apoyar a la toma de decisiones imprescindibles que permitan culminar en tiempo y forma el desarrollo de proyectos en proceso. La metodología Kanban se ha utilizado para la gestión de proyectos, permitiendo darles seguimiento basado en su calendario, y siguiendo el flujo del proceso ya establecido. En este artículo se presenta el desarrollo de una API que permite monitorear y visualizar la comparación del porcentaje de avance: planeado y actual; ayudando a identificar cuando hacer cambios pertinentes para alcanzar el desarrollo del proyecto en la fecha establecida en su formalización.

Palabras clave—Gestión de proyectos, Kanban, Just in Time, valor ganado, valor planeado.

Introducción

La industria de desarrollo de software es de gran importancia en la actualidad por la alta demanda de tecnología que se observa en el país, interviene en todos los procesos que abre paso a la nueva economía, se le considera una industria blanca que no contamina y que genera fuentes de trabajo bien remuneradas.

La principal función de la industria de desarrollo de software es realizar proyectos a la medida de cada empresa, es decir, en base a las necesidades que presenta. Para muchas empresas el desarrollo de software es un medio de ayuda para contribuir en la gestión de recursos, ya que permite administrar de la mejor manera las herramientas con las que se cuenta para lograr un objetivo.

En el mercado existen herramientas de software que ayudan con la administración de proyectos, e incluyen el uso del modelo Kanban y Just in Time. Sin embargo no son capaces de determinar si una tarea en ejecución está fuera del tiempo planeado de acuerdo a la fecha establecida en su formalización. Esta actividad es realizada por el líder de proyecto quien monitorea constantemente cada actividad en tiempo real y así identifica si está fuera del tiempo establecido, este tipo de metodologías se planea con respecto a etapas de ejecución siendo posible cambiar la tarea de una a otra etapa según se requiera.

Una de las problemáticas con las que se encuentra la industria de desarrollo de software es la demanda que se tiene, la cual requiere de un equipo trabajando en su implementación y al mismo tiempo en otros proyectos, dando como resultado un retraso en fechas de entrega, o bien en fechas de ejecución, es por eso que la misma empresa necesita administrar sus propios recursos para llegar a su objetivo planeado en tiempo y forma.

Para dar solución a este problema se propone el desarrollo de una API que permita la valoración de un proyecto en ejecución determinando el avance respecto a la planeación que se realizó antes de empezar la implementación, o bien un aumento en tiempo, dando oportunidad al líder de proyecto tomar decisiones para llegar al fin establecido en su planeación, empleando el método de valor ganado para determinar el porcentaje de avance actual con respecto del que se estipuló al inicio y el porcentaje que tendría que estar cubierto.

Software para gestión de proyectos existente

La gestión de proyecto, entonces, es el uso del conocimiento, habilidades y técnicas para ejecutar proyectos de manera eficaz y eficiente. Se trata de una competencia estratégica para organizaciones, que permite vincular los resultados de un proyecto con las metas comerciales para posicionarse mejor en el mercado (Roberts, Wallace, 2014).

¹ La Ing. Lorena Hernández Carmona es Estudiante de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, Tlaxcala, México. lorena_h_c@hotmail.com

² La Lic. Laura Reyes Briones es Estudiante de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, Tlaxcala, México. lrb8008@hotmail.com

³ El Dr. José Crispín Hernández Hernández es docente del Instituto Tecnológico de Apizaco, Tlaxcala, México. josechh@yahoo.com

⁴ El Dr. Edmundo Bonilla Huerta es docente del Instituto Tecnológico de Apizaco, Tlaxcala, México. edbonn@walla.com

En esta sección se describe el estado del arte de algunas herramientas que permiten la gestión de proyectos, que se encuentran presentes en el mercado actualmente.

Asana. Se trata de una aplicación para gestionar, administrar y compartir proyectos. Permite organizar tareas y objetivos, así como visualizar el avance particular o general de los proyectos en los que los diferentes miembros del equipo están involucrados (Asana, 2017).

Trello. Consiste en la creación de tableros para cada proyecto. Dentro de cada tablero hay listas y cada lista tiene tarjetas con información detallada que puede navegar hacia otras listas dentro del mismo tablero. Puede ser utilizado desde un teléfono o computadora de escritorio y se pueden agregar a las personas que necesiten en cada proyecto (Trello, 2017).

BaseCamp. Es un gestor de proyectos, usado por empresas de todo el mundo para la organización y gestión de tareas de proyectos colectivos. Ofrece diversos planes entre los que se incluye uso ilimitado y gratuito, pero que sólo permite administrar un proyecto y tiene bloqueada la opción para subir archivos (BaseCamp, 2017).

Teambox. Es, al igual que BaseCamp, una de las herramientas más fáciles e intuitivas de usar. Es muy útil para organizar proyectos colaborativos. Ayuda a gestionar muy fácilmente la importancia y prioridad de las tareas, y permite que los usuarios envíen actualizaciones sobre el progreso del proyecto (Teambox, 2017).

JIRA Software. Es una aplicación web para el seguimiento de errores, de incidentes y para la gestión operativa de proyectos. JIRA Software también se utiliza en áreas no técnicas para la administración de tareas. JIRA puede ser utilizado para la gestión de procesos y para la mejora de procesos gracias a sus funciones para la organización de flujos de trabajo (JIRA Software, 2017).

Brightpod. Tiene un diseño y un servicio muy intuitivo que le ayudará a administrar los proyectos de manera más eficiente. Ofrece gestión de proyectos y tareas, listas de tareas, flujos de trabajo, calendario, mensajes, informes, personalizaciones de la interfaz de usuario, y la integración con Google Drive y Dropbox (Brightpod, 2017).

KANBAN Tool. Es una solución para la gestión visual de procesos que ayuda a los equipos a trabajar más eficientemente, visualizar el flujo de trabajo, analizar y mejorar los procesos de trabajo de acuerdo con los métodos Kanban (KANBAN Tool, 2017).

Características/Software	Asana	Trello	Basecamp	Teambox	JIRA Software	Brightpod	Kanban Tool
Workspaces	Si	Si	Si	Si	Si	Si	Si
Notificaciones	Si	No	No	Si	No	No	No
Integración con otras plataformas	Si	No	Si	No	Si	No	No
Aplicación móvil	Si	Si	Si	No	Si	No	No
Manejo de pizarra de equipo de trabajo	No	Si	No	No	No	No	Si
Disponible para descargar	Si	Si	No	Si	Si	Si	No
Diagrama de Gantt o gráficas	No	No	Si	Si	Si	No	Si
Gratuita	Limitada	Limitada	Limitada	Limitada	Limitada	Limitada	Limitada

Cuadro 1. Comparación de herramientas existentes.

En el cuadro 1 se muestra una comparación de algunas características con las que cuentan los software existentes en el mercado para la gestión de proyectos, siendo sólo de algunos de entre gran variedad que se puede encontrar. Este cuadro se presenta con la finalidad de conocer que es lo mínimo que debe ofrecer una herramienta para la gestión de proyectos, la aplicación en la que se trabajará cuenta con estas características mínimas dando como complemento el módulo desarrollado y descrito en este trabajo.

Análisis y propuesta de implementación de API para el seguimiento de un proyecto

La empresa dedicada a la implementación de software a la medida está regida por un su propia metodología de desarrollo, pero todo proyecto debe contar con las principales etapas: inicio, planificación, ejecución, seguimiento y cierre del proyecto. A su vez cada etapa consta de ciertas tareas que se deben de cumplir, esta información es requisito para la implementación de la propuesta de solución sugerida.

Los proyectos están sujetos a usar técnicas para dar seguimiento a sus desarrollos. Para este artículo se describen dos que son Kanban y Just in Time (JIT).

El JIT es una metodología de organización de la producción que tiene implicaciones en todo el sistema productivo. Además de proporcionar métodos para la planificación y el control de la producción, incide en muchos

otros aspectos de los sistemas de fabricación, como son, entre otros, el diseño de producto, los recursos humanos, el sistema de mantenimiento o la calidad. Se define como una filosofía de producción consistente en producir la cantidad de producto exigida por el mercado al nivel de calidad deseado, en el momento preciso (Albizu, 1996).

KANBAN cuenta con dos funciones principales: control de la producción y mejora de procesos. Por control de la producción se entiende la integración de los diferentes procesos y el desarrollo de un sistema JIT. La función de mejora continua de los procesos se entiende por la facilitación de mejora en las diferentes actividades, así como la organización del área de trabajo, mantenimiento preventivo y productivo. El kanban es una tarjeta que contiene información importante sobre la fabricación principalmente. Se utiliza para indicar que se requiere de más trabajo o una siguiente tarea (Albizu, 1996).

Para la propuesta de solución se implementará un módulo que se encargará de valorar un proyecto que se encuentra en desarrollo para determinar si es necesario un cambio en el equipo o una re-planeación en sus tareas, para llegar al objetivo planeado. Se utilizará el método de valor ganado para determinar el porcentaje que se tiene hasta la actualidad comparada con un valor planeado dando como resultado una diferencia positiva, en el caso de que el proyecto lleve menos tiempo del planeado, y negativa para cuando el proyecto vaya atrasado, es decir que lleve más tiempo del planeado, dando oportunidad a la toma de decisiones convenientes para que el desarrollo llegue a su término, en la fecha establecida, o bien para determinar si es necesaria una modificación en los tiempos de entrega.

El análisis del valor ganado se realiza para medir el estado del proyecto por medio de la respuesta a las tres siguientes preguntas:

¿Qué tanto trabajo se planificó?

¿Qué tanto trabajo actualmente se ha completado?

¿Qué tanto ha costado completar el trabajo actual?

El Valor Planeado (PV) es una función del tiempo y representa el valor económico que será invertido en un proyecto.

El Valor Ganado (EV) es una función del tiempo y representa el avance de un proyecto.

En la figura 1 se muestra el algoritmo de la propuesta de solución en el cual se puede observar de manera general lo que se realizará, para dar el resultado deseado. Este será implementado como un módulo de una aplicación ya existente, siendo invocada en el momento que el líder de proyecto lo requiera, en un proyecto que se encuentre en desarrollo (Granda, 2012).

Entre las herramientas de implementación NetBeans será el compilador de desarrollo de la aplicación siendo Java el lenguaje de programación, ya que esta herramienta se adapta a cualquier otro lenguaje, es rápida y es muy fácil para desarrollar aplicaciones de escritorio, móviles y web. Además de ser gratuito, de código abierto, y tiene una comunidad mundial de usuarios y desarrolladores (NetBeans IDE, 2017).

Maven se utiliza en la gestión y construcción de software. Posee la capacidad de realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. Es decir, hace posible la creación de software con dependencias incluidas dentro de la estructura del JAR. Es necesario definir todas las dependencias del proyecto (librerías externas utilizadas) en un fichero propio de todo proyecto Maven, el POM (Project Object Model), es un archivo en formato XML que contiene todo lo necesario para que a la hora de generar el fichero ejecutable de una aplicación este contenga todo lo que necesita para su ejecución en su interior.

Sin embargo, la característica más importante de Maven es su capacidad de trabajar en red. Cuando se define las dependencias de Maven, este sistema se encargará de ubicar las librerías que se desean utilizar en Maven Central, el cual es un repositorio que contiene cientos de librerías constantemente actualizadas por sus creadores. Maven permite incluso buscar versiones más recientes o más antiguas de un código dado y agregarlas al proyecto. Todo se hará de forma automática sin que el usuario tenga que hacer nada más que definir las dependencias. (MAVEN, 2017).

En el diagrama de flujo de la figura 1 se muestra el proceso que se siguió para la implementación con respecto al análisis realizado anteriormente, dando orden cronológico de las actividades a seguir para lograr dar la satisfacción necesaria a las personas interesadas en el correcto funcionamiento de la herramienta que facilite el trabajo y toma de decisiones con respecto de un proyecto en ejecución.

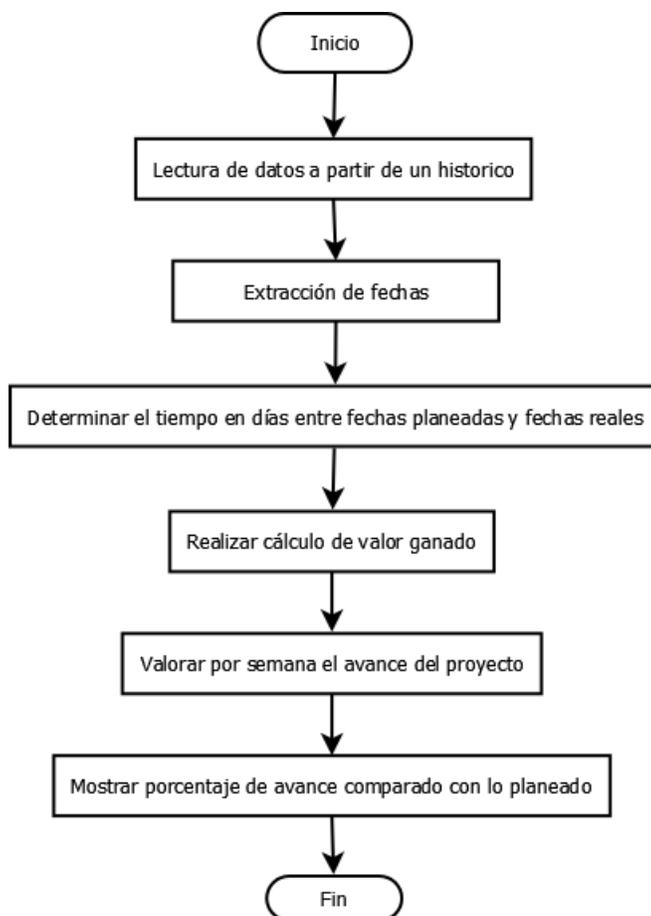


Figura 1. Algoritmo de propuesta de implementación de API para el seguimiento de un proyecto.

Implementación

Para la implementación de esta API es necesario definir los requerimientos con los cuales se podrá generar el módulo antes mencionado, para este desarrollo sólo se requiere de la base de datos conformada de la siguiente manera: tarea, responsable, fecha de inicio planeada, fecha de fin planeada, fecha de inicio real, fecha de fin real, y el estatus de cada tarea. Un ejemplo de registro de la base de datos se observa en el cuadro 2, en el algoritmo presentado se indica como histórico.

Tarea ID	Inicio Planeado	Inicio Real	Fin Planeado	Fin Real	ID Responsable	Estatus
1	21/12/2015 00:00	22/12/2015 09:50	29/02/2016 00:00	22/12/2015 09:50	67	1
2	28/12/2015 00:00	23/12/2015 13:11	31/01/2016 00:00	23/12/2015 13:11	68	1
3	24/12/2015 00:00	26/12/2015 09:57	26/01/2016 00:00	26/12/2015 09:57	70	1
4	21/12/2015 00:00	27/12/2015 20:40	29/01/2016 00:00	27/12/2015 20:40	71	0
5	11/01/2016 00:00	29/12/2015 10:00	16/03/2016 00:00	29/12/2015 10:00	73	0

Cuadro 2. Ejemplo de base de datos

Teniendo como entrada esta base de datos se procederá a realizar la extracción de fecha como primera instancia para determinar el tiempo de duración para cada tarea en días, después de obtener estos datos se procede al llenado de una primera tabla la cual consta de los siguientes atributos: tarea, responsable, tiempo planeado, acumulado del tiempo planeado, semana planeado, valor planeado, acumulado del valor planeado, tiempo actual,

semana actual, valor ganado y acumulado del valor ganado. En el cuadro 3 se puede observar un ejemplo del llenado de la primera tabla.

Para presentar un resultado que se pueda entender se recurre al llenado de una segunda tabla utilizando datos ya calculados en la primera tabla, los atributos que contiene son: semana, tiempo planeado, tiempo actual, acumulado de tiempo actual, acumulado del valor planeado y acumulado del valor ganado. En esta última tabla se puede visualizar la comparación de ambos valores, planeado y ganado, dando como resultado es estado del proyecto actual en desarrollo. El cuadro 4 muestra un ejemplo de la segunda tabla creada en la API.

Tarea ID	ID Responsable	Tiempo Estimado	Acumulado Tiempo Estimado	Semana Planeada	Valor Planeado	Acumulado Valor Planeado	Tiempo Actual	Semana Actual	Valor Ganado	Acumulado Valor Ganado
1	67	70	70	12	13.06	13.06	7	2	1.31	1.31
2	68	34	104	18	6.34	19.4	8	2	1.49	2.8
3	70	33	137	23	6.16	25.56	31	6	5.78	8.58
4	71	39	176	30	7.28	32.84	33	6	6.16	14.74
5	73	65	241	41	12.13	44.97	78	13	14.55	29.29
6	74	97	338	57	18.1	63.07	42	7	7.84	37.13
7	75	60	398	67	11.19	74.26	80	14	14.93	52.06
8	76	94	492	82	17.54	91.8	94	16		
9	77	0	492	82	0	91.8	1	1		
10	78	0	492	82	0	91.8	1	1		
11	80	44	536	90	8.21	100.01	30	5		
12	81	0	536	90	0	100.01	12	2		
13	82	0	536	90	0	100.01	10	2		
14	83	0	536	90	0	100.01	11	2		
15	85	0	536	90	0	100.01	2	1		
16	86	0	536	90	0	100.01	2	1		
17	87	0	536	90	0	100.01	1	1		

Cuadro 3. Ejemplo del llenado de la primera tabla

Semana	Tiempo Estimado	Tiempo Actual	Acumulado de tiempo actuales	Acumulado Valor Planeado	Acumulado Valor Ganado
12	70	7	7	13.06	1.31
18	34	8	15	19.4	2.8
23	33	31	46	25.56	8.58
30	39	33	79	32.84	14.74
41	65	78	157	44.97	29.29
57	97	42	199	63.07	37.13
67	60	80	279	74.26	52.06
82	94	95	374	91.8	
90	44	67	441	100.01	

Cuadro 4. Ejemplo del llenado de la segunda tabla

Para mayor visualización de resultados se presenta una gráfica en la figura 2; donde la línea de guiones representa el acumulado del valor planeado y la línea continua el acumulado del valor ganado.

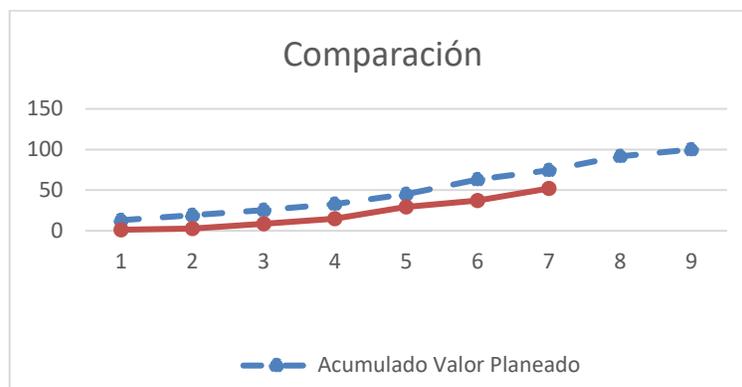


Figura 2. Gráfica de comparación entre valores planeado y ganado.

Conclusión y resultados

En este artículo se presentó la propuesta de solución e implementación de una API, la cual cumplió con los requisitos planteados al inicio de su desarrollo, así como con las necesidades de los clientes, las cuales se tomaron a partir del levantamiento de requerimientos y de las encuestas realizadas. El sistema facilita la toma de decisiones para los líderes de proyecto dando opciones para lograr el objetivo propuesto durante la planeación del proyecto a desarrollar, dando cumplimiento a sus contratos en el desarrollo de software a la medida de sus clientes.

Es necesario realizar pruebas con datos reales en la industria de desarrollo de software ya que debido a la confidencialidad, las pruebas realizadas solo se dieron con bases de datos sintéticas. El sistema será ajustado después de las pruebas realizadas y se pondrá en funcionamiento.

También se presentó un cuadro comparativo, cuadro 1, en el cual se puede observar que existe gran variedad de software para la administración de proyectos. Es cierto que estas herramientas son muy completas pero en ocasiones las empresas no pueden acoplarse a la manera de trabajo con estas, ya sea por su diseño gráfico poco intuitivo o por falta de herramientas que ayuden a controlar un proyecto en ejecución. La herramienta desarrollada cuenta con el cálculo del método de valor planeado y valor ganado.

El módulo fue diseñado para complementar una herramienta de gestión de proyectos, la cual ha sido diseñada a medida de las necesidades de una empresa, este complemento fue desarrollado para mejorar el flujo de ejecución de tareas dando como ayuda al líder de proyecto tomar decisiones en el momento que se requiera para evitar incurrir en fechas de entrega posteriores a las pactadas con sus clientes.

Referencias

- Asana. Disponible en <https://asana.com/>. Último acceso en febrero 2017.
- Base Camp. Disponible en <https://basecamp.com/>. Último acceso en febrero 2017.
- Brightpod. Disponible en <http://www.brightpod.com/>. Último acceso en febrero 2017.
- Eneka Albizu Gallastegi, "La producción flexible Just in Time vs. organización científica del trabajo". Revista de dirección de empresas, N°4, Noviembre 1996.
- JIRA Software. Disponible en <https://es.atlassian.com/software/jira>. Último acceso en febrero 2017.
- José Manuel Granda Gutiérrez, "Valoración de la aplicación del método del valor ganado a proyectos industriales", Oviedo, España, 2012.
- KANBAN Tool. Disponible en <https://kanbantool.com/es/>. Último acceso en febrero 2017.
- MAVEN. Disponible en <http://maven.apache.org/index.html>. Último acceso en febrero 2017.
- NetBeans IDE. Disponible en <https://netbeans.org/>. Último acceso en febrero 2017.
- Profesor Alexander Roberts y el Dr. William Wallace. "Gestión de proyectos", Gran Bretaña, 2014.
- Teambox. Disponible en <https://redbooth.com/?ref=teambox>. Último acceso en febrero 2017.
- Trello. Disponible en <https://trello.com/>. Último acceso en febrero 2017.

API PARA LA ADMINISTRACIÓN DE RECURSOS EN EMPRESAS DE DESARROLLO DE SOFTWARE BASADO EN UN MÓDULO DE SELECCIÓN DE PATRONES GRADUALES FRECUENTES DIFUSOS

Ing. Lorena Hernández Carmona¹, Lic. Laura Reyes Briones²,
Dr. José Crispín Hernández Hernández³, Lic. José Antonio Cruz Zamora⁴ y Lic. José Ruperto Rodríguez Lezama⁵

Resumen—La lógica difusa es útil como un lenguaje que permite traducir información en lenguaje natural a un lenguaje matemático formal, aun en casos donde el mismo concepto puede tener diferentes grados de imprecisión en diferentes contextos. En este artículo se presenta la implementación de un módulo de selección de patrones graduales frecuentes difusos obtenidos de los históricos de administración de recursos en empresas de desarrollo de software. Se generan las frecuencias de tiempos de cada fase de desarrollo, que permiten aumentar o disminuir los tiempos de desarrollo totales de proyectos software a partir de las posibles relaciones concordantes entre fases observadas.

Palabras clave—Patrones graduales frecuentes, lógica difusa, grados concordantes.

Introducción

Si bien en la industria de desarrollo de software, es importante tener en cuenta los tiempos de desarrollo de cada etapa de implementación, para medir el rendimiento de cada integrante de un equipo de trabajo y así determinar un nivel de conocimiento, el cual dé a la empresa, una idea de cómo podrían conformarse con probabilidad de llegar al objetivo de desarrollo en tiempo y forma con el equipo de trabajo adecuado, sin necesidad de tener pérdidas en recursos, ya sean económicos o en tiempo.

La problemática que se puede encontrar en empresas dedicadas al desarrollo de software, son los constantes atrasos que se tienen en la entrega de un sistema, causando tiempos perdidos y hasta ganancias perdidas, a veces disminuyendo la credibilidad y creando desconfianza en los clientes (Mary P., et al. 2013).

La solución propuesta consiste en buscar coincidencias en un histórico existente, para determinar cuáles son los errores cometidos frecuentemente, y así poder dar una posible solución para enfrentar un problema similar o igual.

En el presente trabajo se desarrolla una propuesta de solución e implementación de un módulo de minería de datos basada en extracción de patrones graduales frecuentes para determinar las tareas que deben tener prioridad en la implementación con el fin de evitar tomar más tiempo del necesario en la ejecución en una etapa de desarrollo dentro de un proyecto.

El artículo se encuentra organizado de la siguiente manera. La sección descripción del método describe las metodologías de desarrollo utilizadas para el diseño de la API. La sección análisis y propuesta de implementación del algoritmo de extracción de patrones graduales frecuentes difusos proporciona el análisis y propuesta de la solución. En la sección conclusión y resultados se describen las pruebas e interpretación de la salida de la aplicación, así como las conclusiones.

Descripción del método

Patrones graduales frecuentes

Se definen para medir una variación en la relación existente entre dos tipos de objetos, permiten expresar una “tendencia” en los datos en lugar de ver solamente cuándo se satisface la relación de similitud (Hüllermeier E., 2002).

Normalización

La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional (Edgar F., 1970).

¹ La Ing. Lorena Hernández Carmona es Estudiante de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, Tlaxcala, México. lorena_h_c@hotmail.com

² La Lic. Laura Reyes Briones es Estudiante de Maestría en Sistemas Computacionales del Instituto Tecnológico de Apizaco, Tlaxcala, México. lrb8008@hotmail.com

³ El Dr. José Crispín Hernández Hernández es docente del Instituto Tecnológico de Apizaco, Tlaxcala, México. josechh@yahoo.com

⁴ El Lic. José Antonio Cruz Zamora es docente del Instituto Tecnológico de Apizaco, Tlaxcala, México. tonocz2001@hotmail.com

⁵ El Lic. José Ruperto Rodríguez Lezama es docente del Instituto Tecnológico de Apizaco, Tlaxcala, México. rodriguezlj0172@itapizaco.edu.mx

Las bases de datos relacionales se normalizan para: evitar la redundancia de los datos, disminuir problemas de actualización de los datos en las tablas y proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones: cada tabla debe tener un nombre único, no puede haber dos filas iguales, no se permiten los duplicados y todos los datos en una columna deben ser del mismo tipo.

La normalización consiste en transformar los datos reales en una escala de cero a uno, en este artículo se utiliza el método de normalización basado en unidad, el cual consiste en tomar cada columna para encontrar de esta el número máximo y mínimo, para poder realizar la operación correspondiente con cada elemento que se encuentra en la columna, prosiguiendo a realizar este cálculo por cada columna que contenga la base de datos.

Normalización basada en unidad o Función escalar

Es un método utilizado para estandarizar la gama de variables o características de datos independientes en el procesamiento de datos, que también se conoce como la normalización de datos y, en general se lleva a cabo durante la etapa de procesamiento previo de datos.

El método más sencillo es cambiar la escala de la gama de características para escalar el rango de [0, 1] o [-1, 1]. Selección de la gama de destino depende de la naturaleza de los datos. La ecuación general es dada como:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

dónde x es un valor original, x' es el valor normalizado.

El resultado de combinar los ordenamientos difusos y la medida de correlación de rango es un robusto coeficiente de correlación de rangos idealmente adecuado para medir la correlación de rangos para datos numéricos perturbados por el ruido (Bodenhofer et Klawonn, 2008). Este innovador coeficiente de correlación se conoce como medida de correlación de rango basada en el orden fuzzy.

Ordenamiento fuertemente completo

Se calcula la diferencia entre el atributo en la posición j y el atributo en la posición i , multiplicando el resultado por $1 - (1/r)$, enseguida se calcula el valor máximo entre 0 y el resultado de la multiplicación, finalmente se calcula el valor mínimo entre 1 y el resultado anterior. Su definición normal es:

$$Lr_{i,j} = \min \left(1, \max \left(0, \left(1 - \left(\frac{1}{r} \right) \right) * (Ax_j - Ax_i) \right) \right) \quad (1)$$

Ordenamiento estricto

Se calcula la diferencia entre 1 y el valor resultante del ordenamiento fuertemente completo en su posición i, j . Su definición normal es:

$$R(i, j) = 1 - Lr(i, j) \quad (3)$$

T-norms

Las T-norms utilizadas en lógica difusa son una familia de lógicas no clásicas, de manera informal delimitada por tener una semántica que toma el intervalo unitario real [0, 1], para el sistema de valores de verdad y las funciones llamadas t-norms para interpretaciones admisibles de conjunción. Se utilizan principalmente en aplicaciones de lógica difusa y teoría de conjuntos difusos como base teórica para el razonamiento aproximado. Su definición formal para cada una de estas es la siguiente:

$$T - \text{norm mínimo} = \min(Ax_{ij}, Ax_{ij}) \quad (2)$$

$$T - \text{norm producto} = Ax_{ij} * Ax_{ij} \quad (3)$$

$$T - \text{norm Lukasiewicz} = \max((Ax_{ij} + Ax_{ij} - 1), 0) \quad (4)$$

$$T - \text{norm drástico} = \begin{cases} x & \text{si } y = 1 \\ y & \text{si } x = 1 \\ 0 & \text{en otro caso} \end{cases} \quad (5)$$

Análisis y propuesta de implementación del algoritmo de extracción de patrones graduales frecuentes difusos

Análisis

Para la implementación es importante considerar algunos datos manejados por la empresa, derivado de esto se analizó como es que se adecuará la propuesta. Con este análisis se considera evaluar los tiempos de los equipos de trabajo en base a un histórico de cada una de las etapas de desarrollo con las que cuenta la empresa.

Se diseñó un prototipo en lenguaje Scilab para ejemplificar la propuesta y presentarla a la empresa, una vez aprobada esta propuesta se realizó una implementación en lenguaje Java, para ambas versiones se utilizaron datos sintéticos con el fin de realizar pruebas durante el desarrollo del código, estos datos sintéticos se ilustran en el Cuadro 1 y Cuadro 2.

A1	A2	A3
20,2	3,16	0,76
19,5	2,2	0,66
12,7	5,12	0,89
12,2	5,5	0,25

Cuadro 1. Base de datos sintética.

A1	A2	A3	A4	A5	A6	A7	A8	A9
1	76	4	96	0	42	0	0,58	0,61
5	0	48	25	72	85	36	0,99	0
8	0	13	0	0	0	93	0,86	1
79	0	75	0	31	0	83	0,16	0
1	21	36	70	42	97	0	0,22	0,22
63	0	0	86	77	0	19	0,55	0,18
57	97	18	54	23	0	63	0	0,59
80	82	86	91	0	81	96	0,65	0
75	91	70	94	38	26	0	0,73	0

Cuadro 2. Base de datos sintética.

Para utilizar la base de datos proporcionada por la empresa se realizaron adecuaciones por parte de esta, debido a que sólo se trabajará con los tiempos de cada fase, se acordó que tendría los caracteres “T_” al inicio de la leyenda de la columna para identificar que se trata de tiempos.

Para aplicar el método estudiado con anterioridad se propuso el siguiente algoritmo general para su implementación, este puede observarse en la figura 1, en el cual se muestra cada uno de los procesos que se aplican a una base de datos inicial para lograr obtener una concordancia entre los elementos tratados y así tomar la mejor decisión sobre el campo de aplicación. Para esta investigación se toman en cuenta etapas de desarrollo de software y sus respectivos tiempos dentro de un histórico y de esta manera poder determinar cuál es la frecuencia en la que un caso es exitoso o un fracaso total, invirtiendo mayor o menor tiempo en una tarea.

En la presentación de resultados se pretende realizar la interpretación de la correlación entre dos o más atributos con las respectivas combinaciones entre estos, según la tendencia que les ha sido asignada, la cual puede ser de incremento o decremento para cada atributo.

La salida presentará el porcentaje de concordancia obtenido entre cada una de las combinaciones que se realizan con los atributos obtenidos de la base de datos.

Propuesta de solución

Obtenida la base de datos se aplica el método de normalización denominado escalamiento o basado en unidad, la extracción de los patrones graduales frecuentes se realizó a partir de la entrada que se solicitó al usuario para indicar el número de atributos y entre cuales se realizaría el cálculo de la dependencia gradual.

Los ordenamientos difusos que se implementaron fueron: ordenamiento fuertemente completo y ordenamiento estricto, para los cuales se ocuparon los patrones graduales frecuentes extraídos y se aplicó la fórmula 2 y fórmula 3

respectivamente, con la finalidad de obtener los ordenamientos difusos. La variable r para esta implementación fue asignada con valor de 0.2 para las pruebas realizadas, el cual puede ser modificado adaptándolo a la necesidad del usuario.

Para realizar este proceso se omite la diagonal principal ya que representa los valores iguales, por este motivo se generó una diagonal de 0 con la finalidad de omitir estos cálculos.

Una vez obtenido el ordenamiento estricto de todos los atributos se realiza el cálculo de los diferentes tipos de T-norms con sus respectivas formulas: 4, 5, 6 y 7.

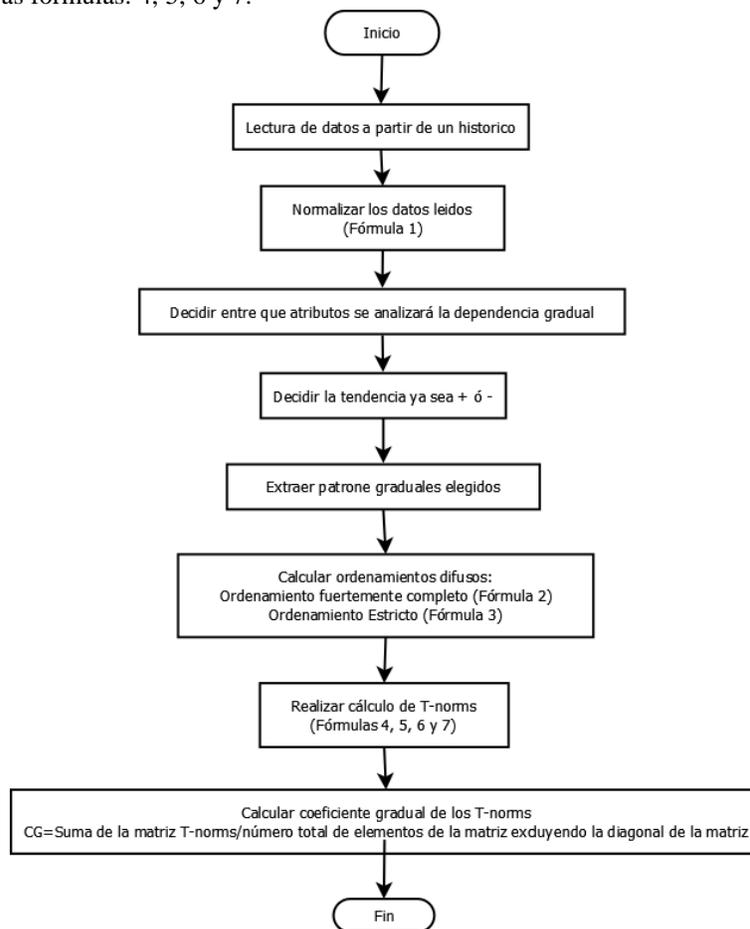


Figura 1. Algoritmo propuesto.

Conclusión y resultados

Resultados

Considerando los datos del Cuadro 3 como datos históricos de proyectos ya desarrollados, se tiene:

T_Backlog	T_Análisis	T_Diseño	T_Desarrollo	T_Pruebas
20	15	37	15	87
95	43	78	225	42
41	98	118	142	14
28	26	35	25	114
18	105	32	529	25
16	48	55	86	27
9	38	14	91	79

Cuadro 3. Datos históricos de proyectos.

Dentro del conjunto de datos se procede a identificar las tendencias en términos de correlaciones por medio de los atributos o columnas que representan los tiempos que han durado las fases en su desarrollo para proyectos anteriores. La ejecución de la API, arroja los resultados que se muestran en la Figura 2.

```
Tiempo en T_Backlog aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 18.29
Tiempo en T_Backlog aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 19.2
Tiempo en T_Analisis aumenta, tiempo en T_Diseño aumenta, el porcentaje de correlación es de 14.79
Tiempo en T_Diseño aumenta, tiempo en T_Desarrollo aumenta, el porcentaje de correlación es de 18.72
Tiempo en T_Backlog disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 18.29
Tiempo en T_Analisis disminuye, tiempo en T_Diseño disminuye, el porcentaje de correlación es de 14.79
Tiempo en T_Diseño disminuye, tiempo en T_Desarrollo disminuye, el porcentaje de correlación es de 18.72
```

Figura 2. Resultado de la ejecución de la API.

Con esta información es posible interpretar que a un aumento de tiempo en Backlog, el tiempo en Diseño aumenta, con un porcentaje de soporte del 18.29%, este resultado se gráfica y puede observarse en la Figura 3.

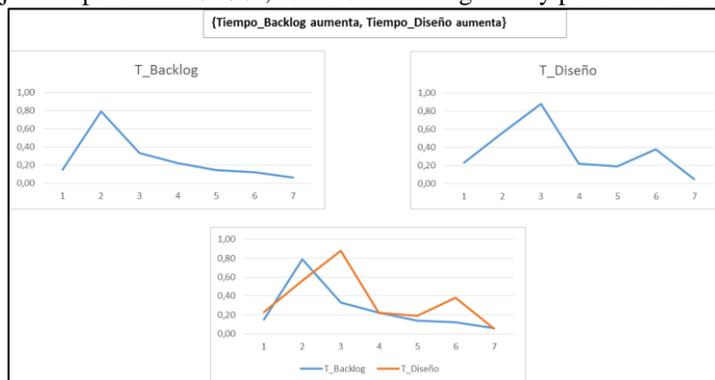


Figura 3. Gráfico de resultado.

En este caso el patrón gradual frecuente indica que con una frecuencia del 18.29% de acuerdo con los datos históricos, ocurre que entre más tiempo en backlog, se aumenta el tiempo en diseño. Para estas pruebas, el umbral se definió en 15, pero este puede ser modificado según la consideración del usuario.

Conclusión

El uso de las tecnologías conlleva en primer lugar a la recolección de datos, lo que a través del tiempo genera grandes volúmenes de información, la cual al ser procesada con patrones graduales frecuentes puede generar un porcentaje de concordancia, en segundo lugar se busca encontrar dentro del conjunto de datos, los patrones graduales frecuentes que aparecen, quedando sólo la información más relevante.

En este trabajo se ha mostrado tan solo una aplicación del uso de la extracción de patrones graduales frecuentes como una herramienta de conocimiento e interpretación para la toma de decisiones en el desarrollo de proyectos. Existen diferentes algoritmos que han sido propuestos para descubrir tales patrones, muchos de ellos para trabajar en cómputo paralelo cuando se consideran grandes volúmenes de información.

Para el estudio realizado, se concluye que la relación que existe entre una fase de desarrollo de un proyecto, se encuentra relacionada con las fases posteriores, de tal manera que el incremento o decremento del tiempo que se invierte en el desarrollo de una fase, influye en el desarrollo de la fase posterior. Este resultado puede considerarse como valioso en futuras planificaciones para desarrollo de proyectos, pues el tiempo que se invierte a una fase modificara el tiempo de desempeño de las fases siguientes.

Referencias

Bodenhof et F. Klawonn. "Robust rank correlation coefficients on the basis of fuzzy orderings: Initial steps. Mathware & Soft Computing", 2008.

Edgar F., "A Relational Model of Data for Large Shared Databanks", Communications of the ACM, 1970.

Hüllermeier E., "Association rules for expressing gradual dependencies", Lecture Notes in Computer Science, 2002.

Mary Poppendieck, Tom Poppendieck, "Lean Software Developmen", Addison Wesley, 2013.

Lautaro G., A. Bedini G., "Gestión de Proyectos de Software", Valparaíso: UTFSM, 2005.



Congreso Internacional de Investigación Academia Journals
Tabasco 2017

AcademiaJournals.com

CERTIFICADO

otorgado a

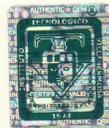
Lorena Hernández Carmona
Laura Reyes Briones
José Crispín Hernández Hernández
Edmundo Bonilla Huerta

por su artículo intitulado

Implementación de una API para el seguimiento de proyectos basados en las metodologías Just in time y Kanban

(Artículo No. Tab632)

el cual fue presentado en el Congreso desarrollado del 29 al 31 de marzo 2017 en Villahermosa, Tabasco, México, y publicado en el portal de Internet AcademiaJournals.com, con ISSN 1946-5351, Vol. 9 #3, 2017 y en el libro electrónico online con ISBN intitulado "Aplicación Del Saber: Casos y Experiencias", Vol. 3



Dr. Humberto José Cervera Brito
Director
Instituto Tecnológico de Villahermosa

Dr. Rafael Moras
Editor, Academia Journals
Profesor de Ing. Industrial, St. Mary's University



Congreso Internacional de Investigación Academia Journals
Tabasco 2017

AcademiaJournals.com

CERTIFICADO

otorgado a

Lorena Hernández Carmona
Laura Reyes Briones
José Crispín Hernández Hernández

por su artículo intitulado

API para la administración de recursos en empresas de desarrollo de software basado en
un módulo de selección de patrones graduales frecuentes difusos
(Artículo No. Tab659)

el cual fue presentado en el Congreso desarrollado del 29 al 31 de marzo 2017 en Villahermosa, Tabasco, México,
y publicado en el portal de Internet AcademiaJournals.com, con ISSN 1946-5351, Vol. 9 #3, 2017
y en el libro electrónico online con ISBN intitulado "Aplicación Del Saber: Casos y Experiencias", Vol. 3



Dr. Humberto José Cervera Brito
Director
Instituto Tecnológico de Villahermosa

Dr. Rafael Moras
Editor, Academia Journals
Profesor de Ing. Industrial, St. Mary's University

Apéndice B

Carta de liberación

San Andrés Cholula, Puebla a 16 de enero de 2017.

ASUNTO: Carta de liberación.

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
P R E S E N T E.

Por medio de la presente hago constar que la **Lic. Lorena Hernández Carmona** alumna de la **Maestría es Sistemas Computacionales** del **Instituto Tecnológico de Apizaco**, con número de control **M10370133** ha cumplido satisfactoriamente su estancia de investigación con base a los puntos contenidos en el cronograma de actividades presentado, elaborando el proyecto:

“IMPLEMENTACIÓN DE UN MÓDULO DE PREDICCIÓN DE ESCENARIOS PARA ASIGNACIÓN DE PERSONAL EN EL DESARROLLO DE PROYECTOS.”

Durante el periodo comprendido del **15 de junio de 2016 al 15 de diciembre del 2016**, el cual se desarrolló en Torre Titanium 5to. Piso oficina 501, 1er Retorno Osa Menor No. 2, en Reserva Territorial Atlixcáyotl, San Andrés Cholula Puebla.

Se extiende la presente para los efectos que al interesado convenga.

A T E N T A M E N T E



Adriana López Tuxpan

Administrador



Apéndice C

Carta de satisfacción

Sta. Ana Chiautempan, Tlax. A 20 de marzo 2017

ASUNTO: Constancia de satisfacción

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL INST. TEC. DE APIZACO
PRESENTE:

AT'N: JOSE FEDERICO CASCO VAZQUEZ
JEFE DE LA DIV. DE EST. DE POSGRADO E INV.

Por medio de la presente reciba un cordial saludo y al mismo tiempo hago constar que la C. Lorena Hernández Carmona, estudiante de la Maestría en Sistemas Computacionales con No. de control M10370133 del Instituto Tecnológico de Apizaco que usted dignamente dirige, participo como analista y desarrolladora en el proyecto denominado: **"Asistente Inteligente para la Administración de Recursos en Empresas de Desarrollo de Software"**, bajo el PROGRAMA DE ESTÍMULOS A LA INVESTIGACIÓN, DESARROLLO TECNOLÓGICO E INNOVACIÓN 2016 del CONACyT, con el registro No. 232834, cuya participación consistió en el desarrollo del módulo de Minería de patrones graduales frecuentes desde datos de PSP y TSP, contando con la supervisión del Dr. José Crispín Hernández Hernández (Asesor Científico) y el Dr. Perfecto Malaquías Quintero Flores (Responsable Técnico del Proyecto).

La asesoría y servicio especializado realizado por los investigadores antes mencionado y la participación de la desarrolladora de software antes mencionada para la realización de este nuevo producto, **cubren y satisfacen las expectativas** planteadas al inicio de este proyecto. El acuerdo, desarrollo, seguimiento, codificación y resultados del mismo se encuentran a resguardo de nuestra empresa como secreto industrial.

Agradeciendo sus atenciones quedo de ustedes.

ATENTAMENTE



M.C. OSCAR ARIANO PONCE
DIRECTOR GENERAL
SMARTSOFT AMERICA BA

Prolongación Antonio
Díaz Varela No. 222
Col. Buenos Aires

Santa Ana Chiautempan
Tlaxcala, México
Tel: (246) 466 8338

www.smartsoftamerica.com.mx

Bibliografía

- [1] Brightpod. 2017. URL <https://www.brightpod.com/>.
- [2] David J. A. *Kanban: Cambio Evolutivo Exitoso Para su Negocio de Tecnología*. Blue Hole Press, 2011.
- [3] Calum Benson, Matthias Muller-Prove, y Jiri Mzourek. Professional usability in open source projects: Gnome, openoffice. org, netbeans. En *CHI'04 extended abstracts on Human Factors in Computing Systems*, págs. 1083–1084. ACM, 2004.
- [4] Javier Carrasco Morgado y Gonzalo Vera Rodríguez. Proyges. 2013.
- [5] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [6] Glad Deschrijver y Etienne E Kerre. On the composition of intuitionistic fuzzy relations. *Fuzzy Sets and Systems*, 136(3):333–361, 2003.
- [7] Javier Donoso Díez. *Aplicación web para la gestión de proyectos software basados en repositorios de conocimiento*. B.S. thesis, 2015.
- [8] Carlos Segundo Figueroa López, Sabando Sudario, Paúl Antonio, et al. *Diseño de una aplicación móvil para la operación del servicio de alquiler de bicicletas*. B.S. thesis, Espol, 2017.
- [9] Bo Yuan George J. Klir, Ute H. St. Clair. *Fuzzy Set Theory: Foundations and Applications*. Prentice Hall, 1997.
- [10] Edward J. Hay. *Justo A Tiempo*. Grupo Editorial NORMA, 1987.

-
- [11] Eyke Hüllermeier. Association rules for expressing gradual dependencies. En *European Conference on Principles of Data Mining and Knowledge Discovery*, págs. 200–211. Springer, 2002.
- [12] Shore Labs. Kanban tool. 2009. URL <https://kanbantool.com/es/>.
- [13] Víctor Lavid Gutiérrez et al. Análisis, diseño, realización y gestión de una base de datos con elementos open source y con diversidad de dispositivos clientes. 2016.
- [14] Dustin Moskovitz y Justin Rosenstein. <https://asana.com/>. Noviembre de 2011.
- [15] Diego Navarro. Seguimiento de proyectos con el análisis del valor ganado. *Universidad Nacional de Colombia*, 2006.
- [16] FRANCESC CODINA NAVARRO. *Desarrollo de una aplicación para la gestión de calidad de los procesos en el entorno JIRA*. Tesis Doctoral, 2011.
- [17] C. G. Ordoñez. *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*. RA-MA EDITORIAL, 1995.
- [18] Rafael Ronda García. Una herramienta web para la planificación de horarios y de asignaciones de aulas. 2017.
- [19] M. S. Stachowicz y M. E. Kochanska. *Graphic interpretation of fuzzy sets and fuzzy relations, Mathematics at the Service of Man*. Universidad Politecnica Las Palmas, 1982.
- [20] Auribox Training. Curso de java frameworks (struts+ spring+ hibernate+ maven+ svn)-auribox training. 2016.
- [21] Dr. William Wallace. *Gestión de Proyecto*. 2002.
- [22] Lotfi A. Zadeh. *Similarity Relations and Fuzzy Orderings*. Information Sciences, 1971.
- [23] Hans Jurgen Zimmermann. *Fuzzy Set Theory*. Springer Science, 2001.
- [24] H.J. Zimmermann. *Fuzzy Set Theory and its Applications*. Kluwer Academic, 1993.