

El Transporte Ferroviario: Fundamentos y algunas características más sobresalientes

Dra. Aranzazu de Caballero¹, Dr. Rony Caballero George¹, Dr. Humberto Álvarez¹, Dr. José Javier Laguardia Cupertino, Dra. Delva Batista Mendieta¹, Dr. Darío Solís¹, Dr. Juan de Dios Sanz Bobi², Dr. Ramón Galán López², Ing. Juan Andrés Brunel Vásquez², Est. Krisly Guerra Guerra¹, Est. Joel Flores Pino¹.

Universidad Tecnológica de Panamá¹ Universidad Politécnica de Madrid²

Resumen— El presente artículo de divulgación tecnológica/científica tiene como propósito presentar, brevemente, algunos fundamentos y características más sobresalientes del transporte ferroviario. Adicionalmente, se presenta un resumen de los aspectos del proyecto de investigación “Metodologías e índices de desempeño para sistemas de transporte ferroviarios” financiado por la Secretaría Nacional de Ciencia, Tecnología e Innovación del Gobierno de la República de Panamá.

Palabras clave: metro, monorraíl, transporte ferroviario, tren, tren de cercanías, tren ligero.

1. Introducción - Situación Actual

La estructura de la ciudad de Panamá corresponde, actualmente, a un área metropolitana madura [1], ya que responde a un esquema policéntrico, con un núcleo central de gran atracción de viajes (Casco antiguo, Área Bancaria) y una órbita de macrozonas dormitorios (Panamá Oeste, Este y Norte) de alta densidad con una súper especialización de los usos de suelos. Atendiendo a esto, su movilidad urbana podría caracterizarse de la siguiente manera:

- Fuerte radialidad de los desplazamientos. Esta situación se agrava en el área metropolitana de Panamá porque los desplazamientos diarios tienen un componente de desplazamiento longitudinal de dirección (Este-Centro), que se suma a los desplazamientos (Norte-Centrosur) y los desplazamientos provenientes del área de Panamá Oeste hacia centro de la Ciudad.
- Concentración de orígenes destinos. La concentración de los viajes se encuentra vinculada a las macrozonas receptoras de viajes (Casco Antiguo, Centro Bancario) y las macrozonas conformadas por San Miguelito, Tocumen y Panamá Oeste, todas ellas generadoras de viajes.
- Concentración temporal de los desplazamientos. En el caso de la ciudad de Panamá, dicha concentración temporal se encuentra en función de las horas picos matutinas (6:00 a.m. a 8:00 a.m.) y las horas pico vespertinas (6:00 p.m. a 8:00 p.m.).
- Una elevada, pero decreciente participación de los modos de transporte público.

En el Área Metropolitana, la mayoría de los viajes (50.3%) se realizan en transporte público.

Según el estudio del Banco Mundial [6] se realizan 1,715,122 viajes diarios distribuidos en varios modos de transporte, donde el 50.3% corresponde a 874,164 viajes diarios en modo autobús.

- Una baja movilidad relativa personal [2]. En este sentido, al comparar la ciudad de Panamá con otras ciudades latinoamericanas, se observa que su promedio de la cantidad de viajes (1.34) es menor que el que tienen ciudades como Santiago de Chile (1.73), Lima (2,1) o Sao Paulo (1.88).

Adicionalmente, resulta importante destacar que la encuesta ENVI (Encuestas Nacionales de Niveles de Vida) muestra que la mayoría de los viajes realizados en el área metropolitana de la ciudad de Panamá están orientados al trabajo (35.4%) y los estudios (39.3%), más no a razones privadas (25.3%), a diferencia de otras capitales latinoamericanas y, de ese 25.3% un alto porcentaje del 9% corresponde a motivos de viajes por compras. En consecuencia, se evidencia que la mayoría de la población de la ciudad de Panamá sólo realiza viajes obligados debido al bajo nivel de movilidad [2].

Tabla 1. Distribución de motivos de viajes [2].

MOTIVO DE VIAJE	PANAMÁ	SANTIAGO DE CHILE	LIMA
	AÑO 2006	AÑO 2002	AÑO 2004
Trabajo	35.4%	23.3%	30.5%
Estudios	39.3%	18.0%	26.2%
Razones Privadas	25.3%	55.7%	37.5%

De acuerdo al estudio de Movilidad Urbana del Banco Mundial [2], los altos tiempos promedios de viaje en el Área Metropolitana de Panamá (AMP) reflejan la problemática de la reducida funcionalidad de la ciudad. Los tiempos promedio de viaje dentro de diferentes puntos, de la AMP son de 57 minutos a 66,5 minutos en transporte público y, 50,6 minutos en automóvil particular. El estudio ESTPUM del año 2000, por otro lado [3], arrojó los siguientes datos:

- El tiempo promedio de recorrido en el autobús es de 44 minutos para la muestra entrevistada.
- El tiempo total promedio del desplazamiento desde su punto de origen a su punto de destino de la muestra entrevistada es de 76 minutos.
- La percepción del tiempo de recorrido, por parte de los usuarios, arrojó que dos tercios de la muestra obtenida mediante la aplicación de una Encuesta de Preferencias Declaradas (EPD), consideraban que el tiempo de recorrido era muy lento o lento. Allí se evidenciaba una visión negativa de las condiciones de desplazamiento en la ciudad de Panamá por parte del usuario del transporte público colectivo, quien ya no estaba satisfecho con sus tiempos de recorrido.

En ambos estudios se muestra que existe una problemática en el sistema de transporte de la Ciudad de Panamá. Se percibe que los tiempos dedicados a movilizarse entre puntos del AMP son relativamente largos pese al tamaño de la Ciudad.

El problema de transporte descrito anteriormente, exige estudiar las diversas alternativas de sistemas de transporte público disponibles actualmente, para determinar cuál de ellas contribuye a mejorar la calidad de vida de los ciudadanos, y como resultado de esto, la economía nacional.

En los siguientes puntos, se abordan temas como las clasificaciones de los sistemas de transportes y se detallan sus principales características en forma de esquemas y tablas, las cuales permiten la comparación entre tecnologías.

Tabla 2. Clasificación del sistema de transporte según la capacidad del sistema [5].

CATEGORÍA	CARACTERÍSTICAS	EJEMPLOS
Baja capacidad	<ul style="list-style-type: none"> Operan en el viario público, a baja velocidad y con múltiples interferencias, debidas al tráfico y múltiples paradas. 	Autobús, trolebús y tranvía
Semi-rápido o de capacidad intermedia	<ul style="list-style-type: none"> Comparten parcialmente el viario con el resto de tráficos. 	Autobuses con plataforma reservada y metro ligero
Rápido o masivo	<ul style="list-style-type: none"> Su infraestructura está totalmente reservada y tienen una alta capacidad, velocidad, regularidad y seguridad. 	Metro y ferrocarril de cercanías

superiores al vehículo privado y empieza a ser atractivo con respecto a éste. La infraestructura totalmente segregada, Categoría A, permite ofrecer una excelente calidad de servicio ya que el operado puede organizar de forma independiente su programa de operación y no sufre la incidencias con otros modos de transporte.

Tabla 3. Clasificación del sistema de transporte según el volumen de viajeros [5].

CLASIFICACIÓN	VOLUMEN DE TRANSPORTE (PASAJEROS EN HORA PUNTA Y SENTIDO)
Baja Demanda	Menor de 6,000
Demanda Media	6,000 a 20,000
Alta Demanda	Mayor de 20,000

Tabla 4. Clasificación del sistema de transporte según grado de exclusividad de la infraestructura ferroviaria [5].

CATEGORÍA	VELOCIDAD DEL TRANSPORTE PÚBLICO	VIAJEROS/HORA/SENTIDO	CARACTERÍSTICAS	EJEMPLOS
A (Infraestructura totalmente segregada)	Mayor a 50 km/h	Mayor de 20,000	<ul style="list-style-type: none"> El transporte público circula por una plataforma independiente, sin cruces a nivel de vehículos y personas. Costo más elevado 	Metro convencional, ferrocarril suburbano, people – movers
B (Infraestructura parcialmente segregada)	Mayor a 20 km/h	Menor de 20,000	<ul style="list-style-type: none"> El transporte público circula por una plataforma separada del resto del tráfico pero con cruces a nivel de vehículos y peatones, incluyendo intersecciones. 	Metro ligero, autobús en sitio propio
C (Infraestructura no segregada)	Menor a 15 km/h	No superior a 6,000	<ul style="list-style-type: none"> El transporte público comparte la infraestructura con el resto del tráfico viario. 	Transporte de superficie, autobús y tranvía

Cuando el transporte público comparte la infraestructura vial con el tráfico general, (Categoría C), la calidad de la oferta de transporte ofrecida en hora punta es muy baja, sobrepasando la velocidad rara vez de los 15 Km/h en medios urbanos. Con una infraestructura parcialmente segregada, categoría B, el transporte público ya puede alcanzar velocidades un poco

2. Clasificación de los Sistemas de Transporte

Entre los parámetros para clasificar los sistemas de transporte en el ámbito urbano y metropolitano, tenemos el tipo de oferta propuesto. En base a este criterio, se presentan tres de las posibles clasificaciones establecidas por Vukan Vuchic [4]: según el volumen de viajeros, según la capacidad del sistema y según el grado de exclusividad de la infraestructura; todas éstas aparecen resumidas en las Tablas 2, 3 y 4, respectivamente.

3. Esquemas Generales de Relación de Capacidades y Velocidades Comerciales Ferroviarias

Las características típicas del transporte público en el área metropolitana, así como el volumen de pasajeros, hacen viable el uso del tren metropolitano en la Ciudad de Panamá. A continuación se presentan las figuras 1 y 2 que muestran las relaciones de velocidades comerciales y capacidades de transporte, en base a la distancia inter-estación. Estas permiten comparar el desempeño de diversos modos de transporte masivo [1], [4], [5], [6], [7] y [8] en donde se vislumbran dos claras fronteras en los límites inferior y superior: el autobús y el tren de cercanías. Entre ellos, también aparecen otros sistemas de transporte terrestre ferroviario, los cuales comparten algunos rangos de valores de sus parámetros de desempeño con sus correspondientes vecinos.

En la Figura 1, se puede observar el caso del metro pesado convencional.

Este sistema de transporte presenta velocidades comerciales entre 25 km/h y 40 km/h, en una distancia inter-estación de 500 m a 2,000 m. En la figura 2, para este mismo sistema, la capacidad de pasajeros/hora/sentido está entre 20,000 a 80,000, considerando la misma distancia entre estaciones.

En el caso de la ciudad capital, ésta presenta altos niveles de congestión vehicular a lo largo de los corredores principales tanto longitudinales como transversales, debido a que las velocidades vehiculares medias en la hora pico de la mañana son de 18 km/h y, en hora pico de la tarde, de 16 km/h [2]. Esto evidencia que el AMP de la Ciudad de Panamá impera prácticamente un sistema de transporte categoría C de muy baja calidad. Para lograr el nivel de desempeño de un sistema de transporte categoría A, se requiere el uso de tecnologías de transporte ferroviario ya que las mismas no solamente permiten el transporte masivo de pasajeros, sino también a altas velocidades y en infraestructura dedicada.

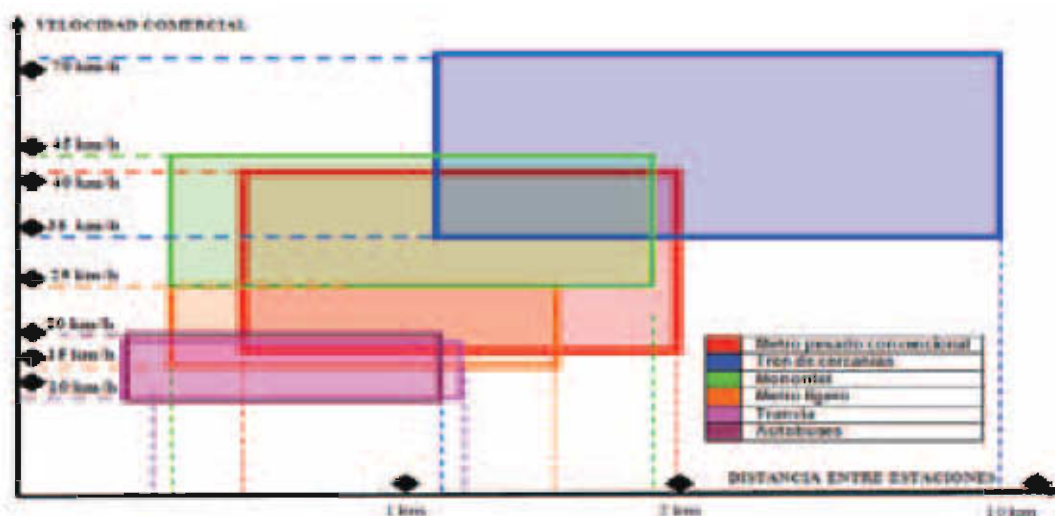


Figura 1. Esquema general de velocidades en los modos de transporte ferroviario. Elaboración conjunta. UTP – CITEF

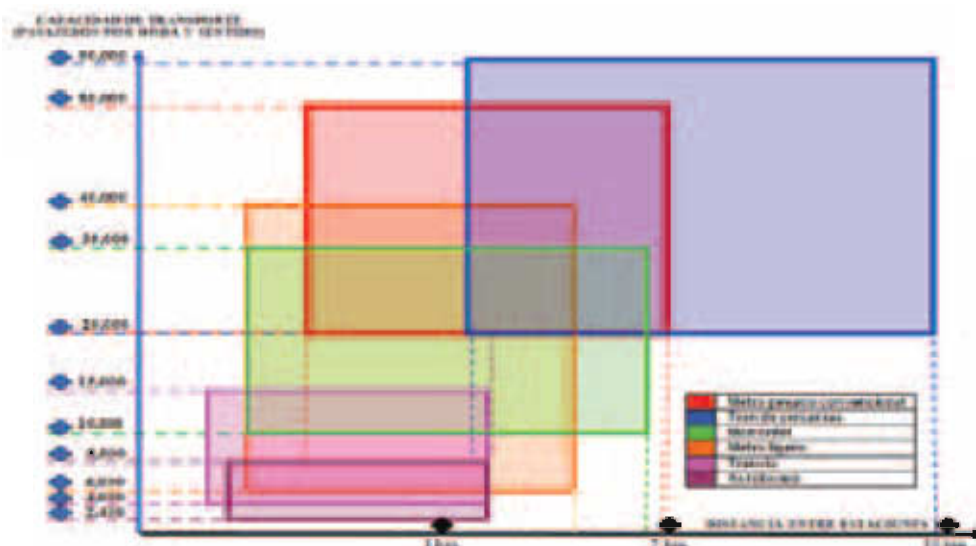


Figura 2. Esquema general de capacidades en los modos de transporte ferroviario. Elaboración conjunta. UTP – CITEF

4. Aspectos Ferroviarios

En la Tabla 5 se presenta una comparación entre tecnologías de transporte ferroviarias y vehiculares. Cabe señalar que los valores presentados en dicha tabla son sólo de carácter orientativo.

Existen diversas tecnologías, cada una con diferentes alcances en términos de capacidad, velocidad, modo de operación, entre otros. Las especificaciones de sendas tecnologías permitirán determinar cuál de éstas es la más adecuada para satisfacer los requerimientos del sistema de transporte de una ciudad.

Como se puede observar en la Tabla 5, el metro convencional y el tren de cercanías ofrecen las mejores prestaciones, en cuanto a

capacidad por sentido (pasajeros), regularidad, velocidades comerciales, entre otros factores. La principal diferencia entre estas tecnologías, aparte de las mostradas en la tabla, consiste en la zona de utilización del sistema, ya sea en el centro urbano o su periferia.

La selección de uno de estos dos sistemas de transporte para la ciudad dependerá, principalmente, de la densidad de tráfico (sucesión de trenes) y la cantidad de paradas (densidad de población); por eso donde la línea tiene más paradas la velocidad es baja y donde hay menos paradas, más alta.

Tabla 5. Comparación entre tecnologías de transporte ferroviarias y vehiculares. Elaboración conjunta. UTP – CITEF

CARACTERÍSTICAS	AUTOBÚS URBANO CONVENCIONAL	TRANVÍA CLÁSICO	MONORRAÍL JAPONÉS	METRO LIGERO TRANVÍA MODERNO	METRO PESADO O CONVENCIONAL	TREN DE CERCANÍAS
Capacidad por sentido (pasajeros)	2,400 – 8,000	4,000-15,000	10,000-35,000	6,000 – 40,000	20,000-80,000	Mayor a 20,000 90,000 (cap max.60")
Composición mínima de la unidad (carros o vagones)	1	1 – 2	1	2 – 3	2 – 10	2 – 10
Capacidad por composición (pasajeros)	40 – 120	100 – 300	724	100 – 400	400 – 1,200	750 – 2,000
Frecuencia mínima (s)	137	---	150	180	90	90
Distancia Inter-estación (m)	250 – 1,200	250 – 1,200	730 – 2,000	350 – 1,500	500 – 2,000	2,000 – 10,000
Regularidad	Baja – Media	Baja	Media Alta	Alta	Muy alta	Muy alta
Velocidad máxima (km/h)	40 – 80	50 – 70	80	60 – 90	70 – 110	70 – 140
Velocidad comercial (km/h)	10 – 20	10 – 20	25 – 45	15 – 25	25 – 40	35 – 70
Plazas totales por carro o vagón	40 – 120	100 – 180	342	110 – 250	140 – 280	150 – 250
Asiento por carro o vagón	30 – 80	22 – 40	124	25 – 80	32 – 84	40 – 120
Longitud de cada unidad (m)	8 – 12	16 – 30	15	25 – 45	32 – 150	26 – 250
Toma de corriente	Ninguna	Aérea Catenaria	Vigas de guía	Aérea catenaria	Aérea/tercer carril	Aérea/tercer carril
Accesibilidad	Parada con desnivel, excepcionalmente en subterráneo	Paradas en la acera a nivel	Estaciones elevadas, a nivel	Parada a nivel, excepcionalmente en subterráneo	Estación en subterráneo salvo particularidades	Estación a nivel o en subterráneo
Pago de peaje	Dentro del vehículo	Dentro del vehículo	En la estación	Dentro del vehículo o la estación	En la estación	Dentro del vehículo o la estación
Control del vehículo	Manual/visual	Manual/visual/señales	Manual ATO/ATC	Manual/señales ATC	Señales ATC, sistemas integrales CBTC	Protección convencional/ATP-ATO/ERTMS
Grado de segregación de la infraestructura vial	0	0 – 40%	100%	40% – 90%	100%	100% Ocasionalmente con pasos a nivel
Integración urbanística	Inmediata	Mediana	Mediana	Fácil	Independiente, ya que es subterráneo	Fácil

5. Descripción General del Proyecto de Investigación

Actualmente, la Universidad Tecnológica de Panamá, desarrolla el proyecto I+D titulado "Metodologías e Índices de Desempeño para Sistemas de Transporte Ferroviarios", financiado por la Secretaría Nacional de Ciencia, Tecnología e Innovación (SENACYT), del Gobierno de la República de Panamá.

Este proyecto de investigación se desarrolla con la sinergia de la Universidad Tecnológica de Panamá y la Universidad Politécnica de

Madrid, como colaborador internacional, a través de diferentes unidades, siendo éstas:

- Facultad de Ingeniería Eléctrica de la Universidad Tecnológica de Panamá.
- Laboratorio de Transporte de la Vicerrectoría de Investigación, postgrado y/o extensión de la Universidad Tecnológica de Panamá.
- El Departamento de Automática, Ingeniería Electrónica e Informática Industrial de la Escuela Técnica Superior de

Ingenieros Industriales en la Universidad Politécnica de Madrid (DISAM). Este centro europeo cuenta con una experiencia de más de 30 años en áreas como: control de procesos, visión por computador, robótica, simulación, etc., a través de las actividades docentes, tecnológicas, de investigación.

- El Centro de Investigaciones de Tecnologías Ferroviarias de la Escuela Técnica Superior de Ingenieros Industriales en la Universidad Politécnica de Madrid (CITEF). A modo de presentación, este centro europeo tiene como fines la investigación, innovación, experimentación, estudio y docencia dentro del área tecnológica de transporte ferroviario, siendo este sistema de redes de metro, tren ligero, trenes de cercanías, tranvía moderno, alta velocidad, mercancías, etc.

Este proyecto de investigación consta de tres etapas fundamentales, las cuales son:

Etapas I: Recopilación de datos y estudios y adquisición de equipos y software ferroviarios.

Etapas II: Modelado de las propuestas presentadas en los estudios analizados.

Etapas III: Determinación de las topologías más favorables y/o configuraciones ferroviarias.

El primer informe de avance arrojó las siguientes conclusiones preliminares:

- De acuerdo a la revisión documental de una serie de estudios de transporte realizados en la ciudad de Panamá, la línea 1 del metro de Panamá propuesta, presenta el potencial de conectar la macrozona de San Miguelito, la de mayor generación de viajes, con las macrozonas comprendidas por Casco Antiguo y Centro Bancario, de mayor atracción de estos, con lo cual se asegura que la ruta escogida tenga una demanda considerable en las horas pico. Es importante destacar que durante la hora valle, la tasa de demanda de pasajeros puede caer hasta en un 25% de la demanda pico. Es importante elegir una solución, en términos de tecnologías ferroviarias, que sea flexible y que permita retirar trenes y vagones fácilmente, de acuerdo al estado del régimen permanente de operación.

6. Conclusión

El transporte ferroviario es un componente fundamental de las grandes capitales del mundo. Aquí queda claro que la Ciudad de Panamá requiere, desde hace varios lustros, una alternativa de transporte urbano ferroviario que dinamice la economía nacional. Para altos niveles de demanda de pasajeros, el metro y el ferrocarril de cercanías siguen siendo las opciones más adecuadas.

En principio, lo ideal es elegir un sistema para el que existan diferentes suplidores, tanto para su fabricación como para el posterior mantenimiento, que pudiera ser realizado por una gama de fabricantes distintos del suplidor original.

Agradecimientos

En la Universidad Tecnológica de Panamá, deseamos agradecer al Dr. Dario Solís, a la Ing. Arelis Barahona, al Ing. Celso Spencer, a la Dra. Eliane Boulet de Cabrera y, al Prof. Garrido. A la Fundación Tecnológica de Panamá, por ser la entidad gestora de los fondos de este proyecto de investigación.

En Madrid, España, al Dr. Ramón Galán López y al Dr. Juan de Dios

Sanz- Bobi. A la Secretaría del Metro de Panamá, en especial al Ing. Roberto Roy, a la Licda. Ana Laura Morais y al Licdo. Roberto Vargas, por todo el apoyo que nos han brindado durante la ejecución de este proyecto. A la Cámara de Comercio, Industria y Agricultura de Panamá, al Ministerio de Obras Públicas y la Autoridad de Transporte y Tránsito Terrestre, por habernos facilitado varios de los estudios realizados hasta la fecha relativos al proyecto.

Referencias

- [1] M. Aymerich, J. Colomer, A. Ibeas, R. Izquierdo, J.M. Menéndez, A.Monzón, F. Robusté, M. Turro. A. Zaragoza. Transportes. Un enfoque integral. Tomo I. Transporte y Economía del Transporte. 2001. ISBN 84-380-0196-3
- [2] Banco Mundial. La movilidad urbana en el Área Metropolitana de Panamá. Elementos para una política integral del Departamento de Desarrollo Sostenible de la Región de Latinoamérica y el Caribe del Banco Mundial. 2007.
- [3] BCEOM. ESTPUM: Estudio de factibilidad de un sistema de transporte masivo en el área metropolitana de la ciudad de Panamá. 2000.
- [4] Vuchic, Vurkan R. Urban Public Transportation.1981.
- [5] C. Zamorano, J.M. Bigas, Julián Sastre. Manual de Tranvías, metros ligeros y sistemas en plataforma reservada. 2006. Diseño, Proyecto, financiación e implantación. ISBN 84-86803-61-6.
- [6] M. Melis, F. González. Ferrocarriles metropolitanos. Tranvías, metros ligeros y metros convencionales. ISBN 84-380-0287-0. 2002.
- [7] JICA. Pre feasibility study on the development of mass transit system in Panama City. Final report . March 2008 Ministry of economy trade and industry . 2008.
- [8] Jetro. Seminario taller sobre el "Monorriel y Transporte Urbano Masivo." Jetro seminar Octubre2009. Organización de Comercio Exterior de Japón, Panamá, República de Panamá. Octubre 2009.Panama
- [9] A. Berbey. Planificación en tiempo real de tráfico ferroviario. Tesis doctoral. Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid. 2008.

Anexos

Conceptos Básicos

- **Macrozonas:** Conjunto de zonas en que se divide un área durante el estudio de localización de un proyecto, tomando en cuenta distintos factores, entre los que se encuentran los técnicos, legales, impositivos, económicos y sociales.
- **Metro:** Sistema ferroviario convencional con características propias de un servicio urbano: pequeña distancia entre estaciones, pequeños radios de curvatura y pendientes importantes.
- **Monorriel:** Sistema en el que el tren se desliza sobre una única viga que descansa sobre postes.
- **Tren:** Serie de vehículos acoplados unos con otros que, remolcados por uno o varios vehículos motores (locomotora, automotor, etc.), conducen viajeros o mercancías de un punto a otro, por una vía férrea circulando de acuerdo con una marcha o en régimen especial.
- **Tren de cercanía:** Ferrocarril de corto o medio alcance que transporta gran cantidad de pasajeros diariamente, entre el centro urbano y las áreas periféricas urbanas.
- **Tren ligero:** Sistema de transporte ferroviario de pasajeros de capacidad media a escala regional y metropolitana con unidades tipo tranvía o una clase intermedia entre un tranvía y un tren, permitiendo la conexión entre núcleos urbanos y zonas rurales y creando además nuevos potenciales de desarrollo urbano.
- **Velocidad comercial:** Velocidad media que desarrolla un tren entre origen y destino, teniendo en cuenta tanto paradas comerciales como técnicas.

Sistemas de Señalización para el Control Automático de Sistemas Ferroviarios

Dra. Aránzazu de Caballero¹, Dr. Rony Caballero George¹, Dr. Humberto Álvarez¹, Dr. Juan de Dios Sanz Bobi², Dr. Ramón Galán López², Ing. Juan Andrés Brunel Vázquez², Est. Krisly Guerra¹, Est. Joel Flores¹

Universidad Tecnológica de Panamá¹
Universidad Politécnica de Madrid²

Resumen – El presente artículo de divulgación tecnológica/científica tiene como propósito presentar, brevemente, las definiciones, características y esquemas más sobresalientes de la señalización ferroviaria. Esta interesante información resulta valiosa para tener una mejor comprensión sobre el tema del metro en Panamá.

Palabras claves: señalización ferroviaria, cantón, ATP, ATO, E.R.T.M.S., ATC, Driverless.

1. Introducción

El origen de la señalización ferroviaria es la consecuencia natural de la necesidad de incrementar el número de trenes circulando dentro de una línea ferroviaria con un nivel aceptable de regularidad sobre ella y la interacción segura entre trenes consecutivos. La evolución de la señalización ferroviaria inició con banderines de diversos colores que anunciaban o denegaban el paso a los trenes; luego, surge la señalización de tipo mecánica, la cual evoluciona a la señalización eléctrica con señales de luces laterales sobre la vía hasta encontramos en nuestros días con dinámicos sistemas electrónicos de información en línea que le indican al conductor ferroviario las acciones a ejecutar a través de las pantallas dentro de la cabina de conducción. De manera general, dentro del sector ferroviario se le denomina señalización a los distintos sistemas de control y protección de trenes, es decir ha sido la técnica de control y regulación del tráfico de trenes basada en semáforos (en ferrocarril, señales), los cuales indican a un conductor de un terminado convoy (grupo de vagones) si éste puede avanzar o no y bajo qué condiciones de velocidad [1].

2. Definiciones Relevantes

Bloqueo: acción de reservar un tramo de la vía, que permite que los trenes circulen con seguridad, de tal forma que no choquen ni se alcancen durante su marcha.

Itinerario: es la secuencia de señales autorizando un determinado trayecto del tren entre dos puntos.

Cantones: tramos de línea protegidos entre dos señales donde solo puede circular un tren.

Enclavamiento: es el corazón del control de los itinerarios y desvíos, asegurando, mediante tecnologías de seguridad, que la posición de las agujas y señales no entren en conflictos y que el tren este protegido durante el paso por los cruces contra una hipotética colisión con otro tren. Esta técnica se conoce también como la creación de itinerarios seguros.

3. Señalización Convencional

La señalización convencional se basa en un sistema de localización segura de cada tren, basada en el principio de que el tren (k) cortocircuita a través de las ruedas y ejes un determinado circuito, dos de cuyos conductores son los propios carriles - véanse figuras 1 y 2.

La señalización convencional comprende dos tipos de elementos fundamentales: los circuitos de vías y los enclavamientos.

Los circuitos de vías son los sistemas encargados de la detección de un tren sobre la vía, control del intervalo adecuado entre trenes sucesivos, apoyo a ciertos sistemas de ATP mediante la transmisión de la información relativa a las señales codificadas transmitidas por los railes y, apoyo en la detección de fracturas de los carriles debido a la interrupción permanente del circuito eléctrico en corto de los dos carriles de un circuito o cantón. Existen varios tipos de circuitos de vías, siendo estos: circuitos de vías con juntas aislantes, sin juntas aislantes y aquellos de contadores de ejes. La Figura 1 presenta una sección de la vía ferroviaria comprendida entre dos juntas inductivas (cantón) que forma un circuito eléctrico, el cual se alimenta de la red de energía alterna, que en caso de cantón desocupado, provoca la excitación de un determinado relé asociado al cantón; la señal del estado del cantón se envía a las señales y los cuartos de enclavamientos.

La situación contraria se presenta en la Figura 2, cuando un tren ocupa dicho cantón, cortocircuita con el primer eje del tren (eje de acero de las primeras dos ruedas metálicas) al circuito y la corriente retorna por un camino con menor impedancia; esto implica la desexcitación del relé anterior y el cambio de sus contactos internos, así pasara la señal de luz verde a luz roja. Con el movimiento del tren sobre la vía, mediante el cortocircuito de los cantones sucesivos, va dejando una especie de rastro de señales en rojo tras de éste.

Las longitudes de los segmentos de los circuitos de vía (c.v.) varían de acuerdo al tipo de operación, sin embargo dichos tramos oscilan entre 40 m y 1000 m en trenes urbanos y en el caso de los ferrocarriles de largas distancias son de algunos kilómetros.

Todo el conjunto formado por los enclavamientos, relés de vía (electromecánicos o electrónicos), las regletas, y cableados correspondientes y los equipos electrónicos para el ATP y el ATO, los mandos de los motores de accionamiento de agujas y alimentación de la señalización están comprendidos en un espacio que se denomina cuarto de enclavamiento. El diseño de dicha instalación ferroviaria crítica debe realizarse con puertas con blindaje, sistemas electrónicos de control de entrada, condiciones de ventilación y humedad, debido a la gran responsabilidad que descansa en estos equipos ferroviarios y su personal y debido a las peligrosas implicaciones que tendría sobre la seguridad de los pasajeros en una red ferroviaria, el uso indebido de estos equipos, vandalismos, saqueos, entre otros incidentes delictivos.

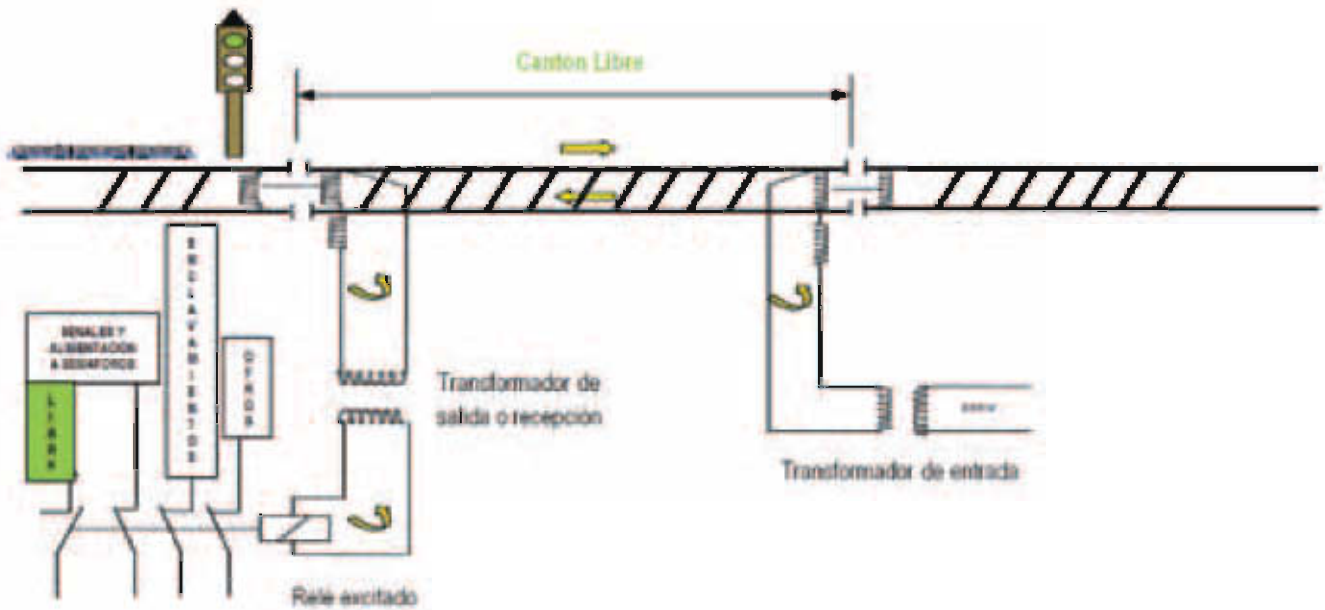


Figura 1. Esquema de circuito desocupado. Elaboración propia con imagen del tren (Cortesía del CITEF: Dr. Sanz Bobi)

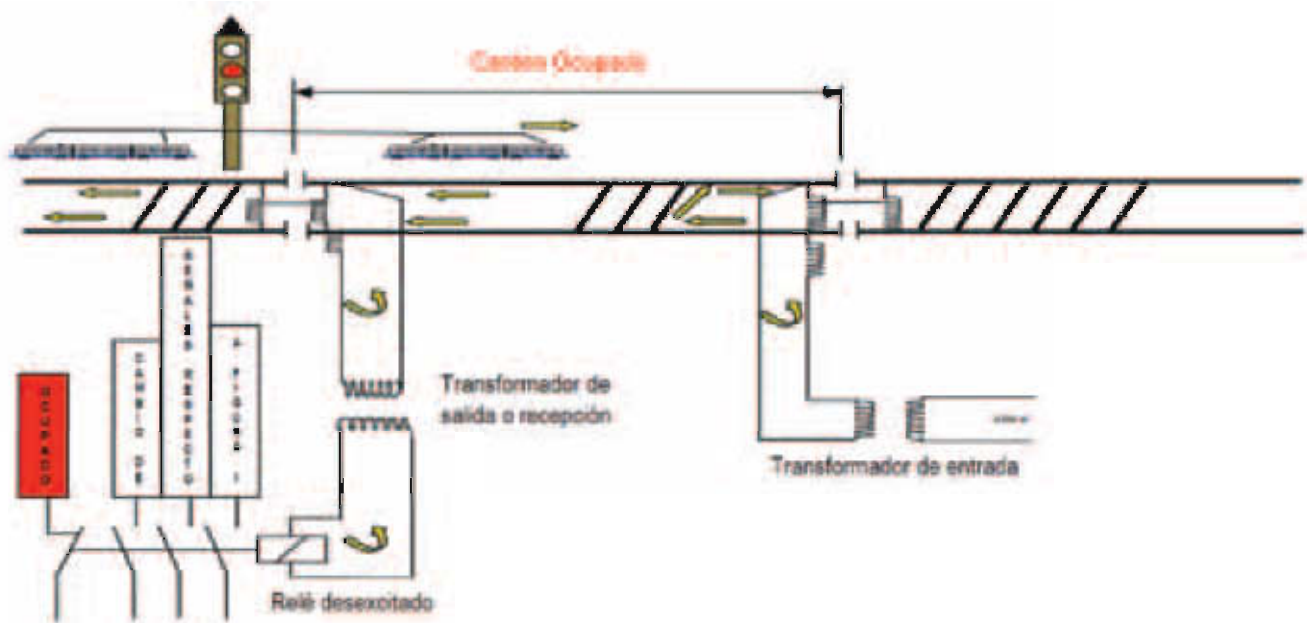


Figura 2. Esquema de circuito ocupado. Elaboración propia con imagen del tren (Cortesía del CITEF: Dr. Sanz Bobi)

De acuerdo a la Figura 3, los modos de conducción se clasifican de menor a mayor grado de automatización en: Operación Por Señales, ATP (Automatic Train Protection), ATO (Automatic Train Operations), ERTMS y sistema Driverless (sin conductor).



Figura 3. Escalera de automatización de los modos de conducción.

3.1. Operación mediante señales

La operación mediante señales constituye el primer peldaño de la escalera de control del movimiento de trenes en un sistema de transporte ferroviario y se utiliza igualmente en fallo de los sistemas ATP y ATO. Las líneas ferroviarias están divididas a efectos de señalización, en zonas controladas por equipos o sistemas denominados enclavamientos donde se centraliza el mando, el control y la supervisión de la zona; son una especie de cuartos de control por subzona ferroviaria. Las señales suelen presentar tres aspectos:

- **Vía libre (verde):** no existe ningún tren circulando en la ruta entre esa señal y la siguiente.
- **Anuncio de parada (amarilla):** indica precaución con respecto a la siguiente señal del itinerario la cual está en rojo. En sistemas de líneas de metro esta señal no suele existir, debido a que las distancias interestación son bastante cortas.
- **Orden de parada (roja):** paso no permitido. En el supuesto de que un tren esté ocupando el cantón que sigue a una señal, dicha señal estará siempre en rojo.

3.2. Conducción ATP

El sistema de protección automática ATP (en inglés Automatic Train Protection) actúa mediante señales codificadas recogidas por el tren cada cierto recorrido (normalmente al comenzar un circuito de vía), indicándole al tren la velocidad objetivo en el trayecto que comienza, hasta la recepción de la siguiente señal. Para no sobrepasar la velocidad máxima permitida, los trenes van dotados de una computadora a bordo. Ésta necesita conocer: la longitud del tren, la curva de frenado del tren, los datos de la ruta (perfiles de gradientes, máxima velocidad, etc.), y la distancia objetivo (distancia que el tren puede recorrer sin detenerse).

Con estos datos, el sistema ATP del ordenador calcula continuamente la velocidad máxima permitida y se la muestra al conductor. Si la velocidad real del tren no sobrepasa la máxima más un pequeño margen (5 km/h), el sistema no actúa.

De lo contrario, el sistema avisa al conductor mediante una señal, sonora o luminosa. Si el tren no frena, a partir de un margen mayor sobre la velocidad máxima calculada (8 km/h), el ordenador interviene sobre el equipo de frenado, el cual no podrá ser liberado hasta que la velocidad se sitúe por debajo de la permitida menos un pequeño margen. Si el conductor no libera entonces los frenos, éstos continuarán frenando el tren hasta que se detenga.

El intercambio de información puede ser por lazos de inducción, cable radiante, balizas, circuitos de vía codificados o radio.

Debido a la alta densidad de trenes en líneas de metro, se utiliza el "deslizamiento" de la señal, para evitar choques. Esto consiste en adelantar la señal al comienzo del cantón a una distancia equivalente a la distancia de frenado de seguridad. Aun así, por precaución es habitual utilizar un cantón sin ocupar como distancia de deslizamiento, sin necesidad entonces de mover las señales. Por lo general, el funcionamiento consiste en indicar al tren dos velocidades mediante la señal: la máxima durante este trayecto y, la objetivo o velocidad máxima que se espera al salir del mismo. En caso de rebasarse la velocidad objetivo, cuando el tren llegue a ese punto será frenado automáticamente. Por ejemplo, una consigna transmitida podría ser "65/50". Esta consigna ordena al control del tren que si alcanza los 65 km/h (más un margen) a partir de ese momento deje de transmitir potencia. Habitualmente se fijan unos márgenes de velocidad, tanto superior (5%), para dejar de acelerar ligeramente por encima de la velocidad máxima, como inferior (8%), para marcar el momento en el que el motor vuelve a suministrar tracción al eje para acelerar. Esta conducción se mantiene durante todo el circuito de vía hasta que el sistema automático calcula, tomando en cuenta la curva de frenado del tren, que ha llegado el momento de frenar con el objetivo de cumplir la exigencia de velocidad "a la salida", que en la consigna "65/50" sería 50 km/h.

Los sistemas ATP se dividen para los ferrocarriles urbanos metropolitanos, en cuatro categorías:

- Sistemas puntuales de velocidad máxima.
- Sistemas cantonales de velocidad máxima.
- Sistemas de distancias objetivo.
- Sistemas lineales o continuos.

Entre los **sistemas puntuales de velocidad máxima** sobresale el Sistema ASFA, el cual consiste en una baliza que se encuentra conectada con la señal y reproduce la indicación de ésta última. La antena situada en la parte inferior delantera del tren capta la información y la remite a la computadora dentro del tren, donde se analiza la información recibida y ésta la envía al tablero de control del conductor. Luego de unos instantes, si el conductor del tren no realiza las acciones pertinentes, la computadora pone en marcha las señales acústicas y vuelve a darle un margen de tiempo al conductor para que éste actúe. Si aún así este no responde (caso extremos de desmayo del chofer), la computadora emite la orden relativa al accionamiento de los frenos de emergencia y el tren frena.

En los sistemas cantonales de velocidad máxima (Figura 7), la premisa fundamental es que "sólo puede existir un tren en un determinado cantón a la vez"; aquí la información de cada c.v. se transmite codificada al tren que lo ocupa, basándose sus criterios de operación en que cada tren, según va ocupando nuevos cantones, recibe de estos la nueva velocidad máxima en dicho

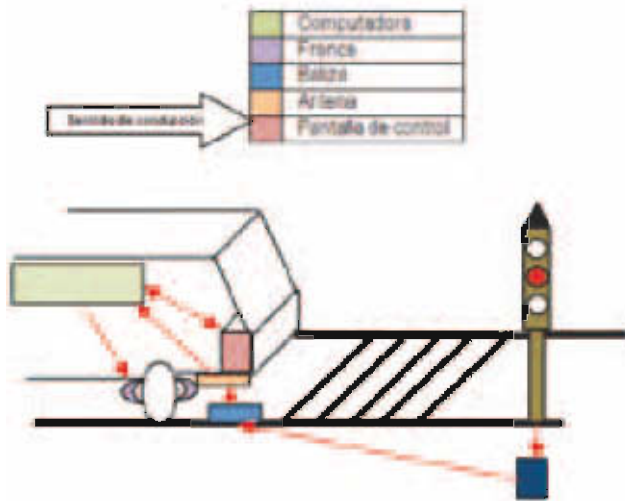


Figura 4. Sistema ASFA. Elaboración propia



Figura 5. Tablero de control dentro del tren. Imagen cortesía CITEF. (Dr. Sanz-Bobi)

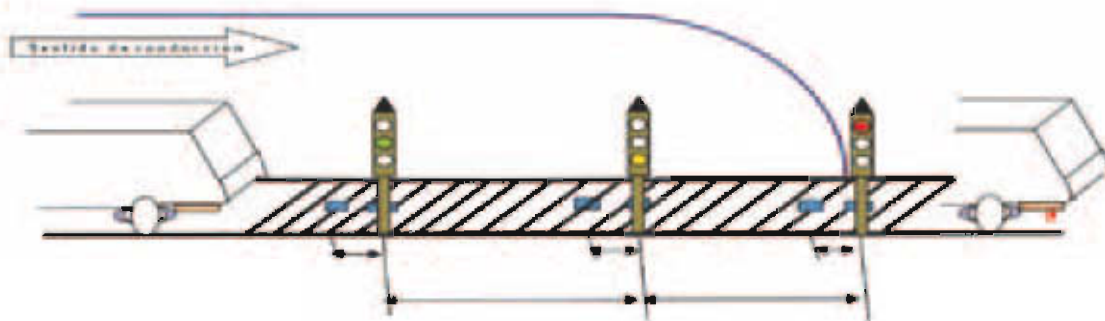


Figura 6. Movimiento de un tren en el sistema ASFA. Elaboración propia.

cantón y va dejando informaciones en el cantón que abandona; esto permite que los trenes sucesivos no puedan llegar a alcanzarle. Para dar cumplimiento a lo anterior, el sistema de protección de trenes calcula las distancias de seguridad desde el inicio del cantón hacia atrás, con las consideraciones de: distancia de frenado de tren posterior con la mínima desaceleración de seguridad posible, distancia de recorrido de los tiempos muertos de operación de los sistemas de frenos, posible distancia de seguridad

de deslizamiento con malas condiciones de adherencia rueda – carril y coeficientes de seguridad diversos.

El sistema de distancia objetivo se distingue del anterior en que cada tren conoce su posición y el cantón que ocupa el tren que le precede (no la posición dentro de dicho cantón). Esto permite que el cantón anterior al ocupado sea el cantón de freno, además de que los trenes puedan estar más cerca uno de otros y las velocidades medias sean más altas entre el trayecto interestación

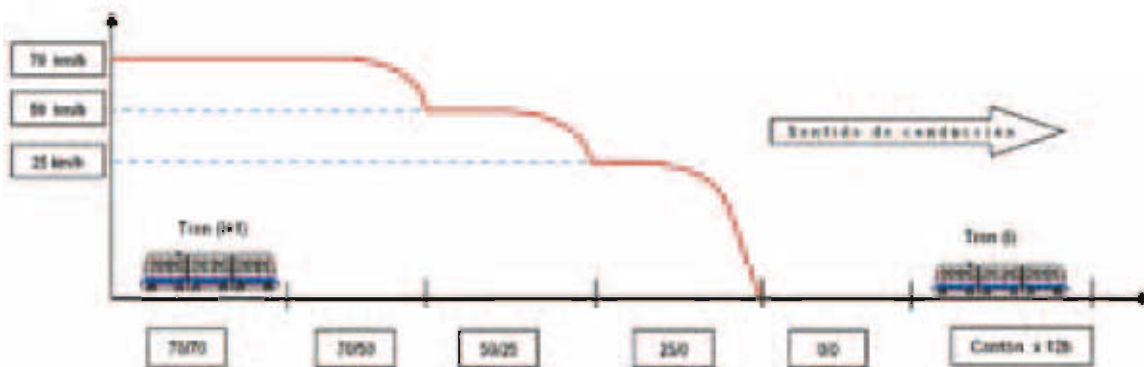


Figura 7. Sistema cantonal de velocidad máxima (sistema de códigos de velocidad)

midiendo la distancia que aún le queda por recorrer a partir del punto, hasta un punto definido de seguridad X, el cual es anterior al circuito de vía ocupado una distancia dS.

3.3. Conducción ATO

Sistema de conducción automática en el que existe una comunicación entre elementos de la vía y elementos que viajan en el tren, pudiendo mandar señales al mismo que influyan o determinen su velocidad. Con este sistema todo queda automatizado, salvo las puertas del tren y la orden de arranque. El funcionamiento del sistema ATO es muy similar al ATP, pero en el ATO tanto la aceleración como la frenada habitualmente recaen en el sistema automático, haciéndolos más suaves y precisos y eliminando la necesidad de conductor, aunque éste se mantiene por la imposibilidad del sistema de resolver averías. Para una frenada precisa, por ejemplo en una estación, el ATO se comunica con el tren mediante balizas, la primera para indicarle que se aproxima un punto de frenado, y las siguientes para reposicionar el tren y recalculando la curva de frenado. Los códigos son los mismos a lo largo de la vía, del tipo "65/50". La regulación de la velocidad se consigue definiendo la velocidad de deriva, aquella a partir de la cual el tren deja de traccionar (el motor deja de suministrar par a los ejes) y va libre y por tanto frenándose. Este sistema permite mayor regularidad en la marcha y por ello mejorar tanto los tiempos de recorrido como la capacidad total de pasajeros transportados. Se hace imprescindible instalar el ATP ya que, cuando actúa el ATO el conductor pierde el control sobre el tren. La mayoría de los sistemas ATO permiten al conductor conducir manualmente, pero aun en esta situación el conductor no podrá sobrepasar la velocidad máxima fijada por el ATP.

3.4. Conducción ATC

La combinación de los sistemas ATP y ATO se denomina ATC (Automatic Train Control), el cual incluye tanto funciones automáticas como algunas que requieren intervención manual. Existen dos formas de implementar el ATC:

a) *Bloqueo totalmente cantonado*: la transmisión de datos al tren puede ser por circuitos de vía, lazos de control o balizas. Estos datos se traducen en comandos de aceleración, frenadas o mantenimiento de la velocidad.

b) *Bloqueo móvil (Señalización basada en la Transmisión – TBS)*: la transmisión se realiza por radio. La posición del tren es determinada por un mapa a bordo, el cual se reinicia cuando el tren comienza el viaje y se verifica mediante balizas situadas a lo largo de la ruta. El área de cobertura del ATP es más grande. Si la transmisión por radio falla, éste asume que el tren que va por delante de él ha parado en su última posición conocida y parará a una distancia de seguridad por detrás de él.

3.5. Sistemas E.R.T.M.S.

El sistema unificado de control de tráfico E.R.T.M.S. (European Railway Traffic Management System) está constituido por los subsistemas E.T.C.S. y GSM – R. Este sistema puede instalarse en tres versiones distintas: nivel 1, 2 y 3, de menos a más prestaciones en cuanto a seguridad, capacidad y velocidad.

Tabla 1. Características de los niveles E.R.T.M.S.

NIVEL	CARACTERÍSTICAS
1	1. Intercambio puntual de información 2. Velocidad máxima: 300km/h 3. Intervalo entre trenes: 5'30"
2	4. Sistema de cantones fijos 5. Intercambio de información vía GSM-R. 6. Velocidad máxima: 350 km/h 7. Intervalo entre trenes: 2'30"
3	8. Sistema de cantones móviles 9. Utiliza GPS para determinar la posición del tren. 10. Aun en fase de desarrollo.

3.6. Driverless

Se refieren a sistemas realmente sin conductor o totalmente automáticos. Debido a la eliminación de personal, se requiere un equipamiento mínimo para mantener la seguridad y la disponibilidad en la conducción:

- CCTV (Circuito Cerrado de Televisión) en estaciones y trenes
- Puertas de andenes para evitar invasiones de la vía.
- Comunicación bidireccional entre los trenes y el Puesto Central
- Transmisión de informaciones del estado del equipamiento de los trenes para monitorización y diagnóstico.

En un gran número de sistemas de transporte sin conductor, los trenes se diseñan para permitir que un operario conduzca los mismos en situaciones extraordinarias o ante incidencias. Los tableros de conducción deben ser invisibles e inaccesibles a los pasajeros.

3.7. C.T.C. (Centro de Control de Tráfico)

Su primera y principal misión es la Gestión del Tráfico de trenes en las líneas, cocheras, depósitos y vías secundarias de forma segura, cumpliendo los requerimientos del tráfico.

Desde el C.T.C. también se controla la calidad del servicio ofertado mediante la regulación de los intervalos entre trenes, cumplimiento de horarios, realización de maniobras de encierres, reposición de trenes en la línea y se resuelven todos los posibles conflictos operacionales de la circulación de los trenes produciendo mínima perturbación al viajero. Además, y al mismo tiempo, el C.T.C. genera valiosa información para los viajeros, como horas de llegada o posibles desviaciones de los horarios. Este sistema permite también un ahorro energético seleccionando automáticamente marchas de trenes más económicas.

Modos de conducción	
M+20	Modo de conducción manual que no permite rebasar la velocidad de 20 km/h.
ATP	Modo de conducción que no permite rebasar el valor de la velocidad de seguridad.
ATO	Modo de conducción automática en la que el propio tren, una vez que inicia su conducción, éste desarrolla su velocidad y se detiene en el siguiente punto de parada sin intervención del conductor.
Paso de hombre	Conducción de un vehículo a una velocidad de una persona caminando.
Auto - shunt	Maniobra de inversión de la conducción que se realiza en el modo de conducción ATO.
Marcha a la vista	Modo de conducción manual de un tren a una velocidad tal que permite detenerlo, si fuese necesario, antes de llegar a cualquier obstáculo o agente que se puede encontrar en la vía.

Tabla 2. Resumen de los modos de conducción para los choferes de trenes.

Referencias

- [1] F.J. González Fernández, J. Fuentes Losa. Ingeniería Ferroviaria. Unidad Didáctica. ISBN 978-84-362-5293-4.2006.
- [2] RENFE. Nociones básicas ferroviarias. Segunda Edición Escuela Técnica Profesional. Editorial Marcombo, S.A, Ediciones Técnicas. ISBN: 978-84-267-1513-5.2008.
- [3] C. Zamorano, J.M. Bigas, Jullán Sastre. Manual de Tranvías, metros ligeros y sistemas en plataforma reservada. 2006. Diseño, proyecto, financiación e implantación. ISBN 84-86803-61-6.
- [4] M. Melis, F. González. Ferrocarriles metropolitanos. Tranvías, metros ligeros y metros convencionales. ISBN 84-380-0287-0. 2002.
- [5] Vuchic, Vurkan R. Urban Public Transportation.1981.
- [6] A. Berbey. Planificación en tiempo real de tráfico ferroviario. Tesis doctoral. Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid. 2008.
- [7] Metro de Madrid. Normas de seguridad de los agentes en relación con la circulación. Metro de Madrid, S.A. Abril 1997. M-19634-1997.
- [8] J. D. Sanz-Bobi. Señalización ferroviaria. Introducción. CITEF. Páginas 30. 2009.



FACULTAD DE INGENIERÍA INDUSTRIAL

Oferta Académica

La Universidad Tecnológica de Panamá, a la vanguardia con el incesante crecimiento de nuestro país; se ha destacado, por ser una de las universidades más sobresalientes y reconocidas internacionalmente por preparar óptimos y competentes profesionales en las áreas de ciencias y tecnología.

Pregrado

- Licenciatura en Ingeniería Industrial
- Licenciatura en Ingeniería Mecánica Industrial
- Licenciatura en Mercadeo y Comercio Internacional / Título intermedio de Técnico en Gestión de Ventas
- Licenciatura en Recursos Humanos y Gestión de la Productividad / Título intermedio de Técnico en Recursos Humanos y Gestión de la Productividad
- Licenciatura en Gestión de Producción Industrial / Título intermedio de Técnico en Ingeniería con especialización en Gestión de Producción Industrial
- Licenciatura en Logística y Transporte Multimodal / Título intermedio de Técnico en Logística y Transporte Multimodal
- Licenciatura en Gestión Administrativa / Título intermedio de Técnico en Gestión Administrativa

Para mayor información:

Coordinación de Postgrado y Maestría

Teléfonos: 560-3145 / 560-3386

Correos electrónicos:

juan.moran@utp.ac.pa luis.vega@utp.ac.pa marta.caballero@utp.ac.pa

Postgrados y Maestrías

- Postgrado en Alta Gerencia
- Postgrado en Logística
- Postgrado en Formulación, Evaluación y Gestión de Proyectos
- Maestría en Dirección de Negocios con especialización en:
 - Estrategia Gerencial
 - Mercadeo Estratégico
 - Gerencia en Recursos Humanos
 - Administración de Sistemas de Información
 - Economía de las Empresas
- Maestría en Sistemas Logísticos y Operaciones con especialización en:
 - Planificación de la Demanda
 - Centros de Distribución
- Maestría en Gestión de Proyectos con especialización en:
 - Administración
 - Evaluación

Visítenos en:

<http://www.fii.utp.ac.pa>

Base de Datos para la Gestión de Ontologías

Belén B. Bonilla M., Davis Arosemena, estudiantes

Víctor López C., MSc.

Universidad Tecnológica de Panamá

belen.bonilla@utp.ac.pa, davis.rosemena@utp.ac.pa, victor.lopez@utp.ac.pa

Resumen- Una ontología es una representación formal del conocimiento donde los conceptos, las relaciones y las restricciones conceptuales son definidos mediante formalismos en un determinado dominio. Las bases de datos utilizadas para almacenar y hacer consultas sobre ontologías se denominan Base de Datos basadas en Ontologías. En este artículo se presenta un estudio sobre las Bases de Datos basadas en Ontologías y se describen las características y propiedades de algunas de estas bases de datos.

Palabras Claves- Ontología, Base de Datos, Sesame, OntoDB, RDFSuite.

1. Introducción

Actualmente, las ontologías son utilizadas en una gran cantidad de dominios. Por ejemplo, en la Web Semántica [1], se utilizan para describir términos, recuperar información e interconectar servicios web, y en el área de base de datos, para facilitar el intercambio e integración de los datos [2].

Debido al uso incremental de las ontologías, se han propuesto una gran cantidad de modelos y lenguajes para el manejo de tales ontologías, dentro de los cuales se pueden mencionar: Resource Description Framework (RDF), Resource Description Framework Schema (RDFS) [3], Ontology Web Language (OWL) [4], entre otros. Sin embargo, resulta necesario también, poder almacenar estas ontologías y sus instancias de tal manera que se puedan acceder y manipular fácilmente.

Varios trabajos han sido desarrollados para representar tanto las ontologías como la data asociada, en bases de datos. Una base de datos que ofrece esta funcionalidad es denominada Base de Datos basada en Ontologías (OBDB – Ontology-based Database) o base de datos ontológica.

Existen diferentes bases de datos para el almacenamiento y gestión de ontologías entre las que se pueden mencionar Sesame [5], OntoDB [6], entre otras. Cada una posee características, propiedades y ventajas, y la elección de una u otra depende de las necesidades del usuario y del lenguaje de definición de las ontologías.

El objetivo de este artículo es presentar un estudio sobre las bases de datos utilizadas para el almacenamiento y consulta de ontologías, su importancia y características.

En la siguiente sección se presenta el concepto de ontologías, los beneficios de su utilización, sus aplicaciones y los lenguajes utilizados para crearlas y definirlos. En la sección 3 se presenta el concepto de base de datos basadas en ontologías o base de datos ontológicas, la necesidad de su utilización y diferentes bases de datos para la gestión de ontologías.

2. Ontologías

Las ontologías proceden del campo de la Inteligencia Artificial; son vocabularios comunes para las personas y aplicaciones que trabajan en un dominio específico [7].

Uschold y Gruninger [8], utilizan el término ontología para referirse a un entendimiento compartido de algún dominio de interés, el cual puede ser utilizado como un framework unificado para resolver problemas tales como interoperabilidad, confusión conceptual y terminológica, etc. Por otro lado, Gruber [9], definen ontología como "una especificación explícita y formal de una conceptualización compartida". El término "conceptualización" se refiere a un modelo abstracto de algún fenómeno del mundo construido mediante la identificación de los conceptos relevantes a ese fenómeno; el término "explícito" significa que los conceptos utilizados en la ontología, y las restricciones para su uso, están claramente definidos; el término "formal" se refiere al hecho de que debe ser comprensible para las máquinas, es decir, estar expresada mediante una sintaxis que permita a una computadora operar sobre ella; y finalmente, el término "compartida" refleja la noción de que contendrá conocimiento consensuado en algún grado [10].

2.1. Beneficios de las ontologías

Actualmente, las ontologías facilitan aspectos como comunicación, interoperabilidad y razonamiento automático.

Comunicación. Las ontologías reducen las confusiones de tipo conceptual y terminológico; de esta forma, habilitan el entendimiento compartido y la comunicación entre personas, organizaciones y aplicaciones con diferentes necesidades y puntos de vista sobre un contexto particular. Esto se logra ya que a través de las ontologías se pueden crear modelos normativos los cuales permiten agregar semántica a un sistema determinado y crear un modelo extensible que puede ser refinado posteriormente y que permite transformaciones semánticas entre diferentes contextos [8].

Interoperabilidad. La interoperabilidad es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos. Para el tratamiento de la interoperabilidad, las ontologías se utilizan para soportar traducciones entre diferentes lenguajes y representaciones. En términos más prácticos, una ontología puede funcionar como traductora permitiendo la interoperabilidad entre dos aplicaciones de software [8].

Razonamiento Automático. Las ontologías resultan muy útiles para facilitar el razonamiento automático, es decir, sin necesidad de la intervención humana. Partiendo de unas reglas de inferencia, un motor de razonamiento puede usar los datos de las ontologías para inferir conclusiones de los mismos.

Una de las aplicaciones más importante del razonamiento automático es la validación de datos. También se usa para establecer relaciones entre ontologías, para descubrir relaciones ocultas o inesperadas entre los datos y para integrar esquemas de bases de datos [11].

Por otro lado, el uso de ontologías, aunado a sus bondades en cuanto a comunicación, interoperabilidad y razonamiento automático, permite que los usuarios organicen y estructuren la información de

manera que los agentes de software puedan interpretar el significado y, por tanto, puedan buscar e integrar datos mucho mejor que como se hace actualmente. Gracias al conocimiento almacenado en las ontologías, las aplicaciones podrán extraer automáticamente datos de las páginas web, procesarlos y sacar conclusiones de ellos, así como tomar decisiones y negociar con otros agentes o personas [12, 13].

2.2. Aplicaciones de las ontologías

Actualmente, las ontologías son ampliamente utilizadas en actividades como: comercio electrónico [14], gestión de conocimiento [15], procesamiento de lenguaje natural [16], recuperación de información [17] y Web Semántica [18], siendo estas dos últimas las de mayor impacto y relevancia.

Por otro lado, las ontologías también son utilizadas en las bases de datos, facilitando el intercambio e integración de los datos almacenados [19].

2.3. Lenguajes

Para crear y manipular ontologías, resulta necesario contar con lenguajes y tecnologías que provean las características adecuadas para su tratamiento y definición, según las necesidades de aplicación [20].

A continuación se detallan los lenguajes más utilizados para crear y definir ontologías.

RDF/RDFS, han sido desarrolladas por el World Wide Web Consortium (W3C) y en conjunto constituyen una herramienta de representación del conocimiento de uso general que proporciona un método neutral para describir un recurso o definir una ontología [3].

Una ontología puede ser definida a través de un RDFS y las instancias de esta ontología, a través de un documento RDF.

DAML+OIL, fue desarrollado por un comité conjunto de Europa y Estados Unidos en el contexto del proyecto DARPA, para permitir interoperabilidad semántica en XML. Es una extensión de RDF así como RDF es un lenguaje basado en XML. Su objetivo es apoyar la transformación de la Web de ser un foro de presentación de la información a ser un recurso de interoperabilidad, comprensión y razonamiento [21].

DAML+OIL coloca énfasis en la semántica clara e introduce varios constructores de lenguajes que incrementan el poder de expresividad en las ontologías.

OWL es un lenguaje de marcado para publicar y compartir datos utilizando ontologías. Tiene como objetivo proporcionar un lenguaje de marcado construido sobre RDF (Resource Description Framework) y XML [4].

OWL es un lenguaje para la definición e instanciación de ontologías y puede ser usado para representar el significado de los términos en vocabularios y la relación entre esos términos. Se podría describir como un conjunto de reglas que ofrecen una forma de definir conceptos y hechos del mundo real.

3. Bases de Datos basadas en Ontologías

Con la Web Semántica y todas las funcionalidades de la Web 3.0 relacionado con la capacidad de inferir conocimiento a través de los metadatos de los datos; se presentan nuevos retos para la gestión de dicha información en la Web.

Como solución se propone la utilización de ontologías que facilita la estructuración de la información con su respectiva semántica; de esta manera, las ontologías no solo son utilizadas para funciones en la Web sino para cualquier otra función en la que se requiera significado de lo que se está procesando en la computadora.

Muchas ontologías representadas en un lenguaje como OWL conjuntamente con sus instancias, representa una dificultad a la hora de gestionar sus operaciones ya que al encontrarse en un texto plano requiere la utilización de herramientas como Protégé [22] que levante toda la información en memoria principal para la realización de dichas operaciones; sin embargo, aunque estas herramientas son fáciles de diseñar están limitadas por la disponibilidad de memoria principal. Basado en esto se propone la utilización de bases de datos que facilite la gestión de las ontologías ya que por las características de un DBMS en la que la distribución, flexibilidad, heterogeneidad representan una gran ventaja, se garantiza la disponibilidad e integridad de la información [23].

Surge entonces, el concepto de Base de Datos basada en Ontologías (Ontology-based Database OBDB). Un OBDB es el almacenamiento de un modelo ontológico y sus instancias en una base de datos para la gestión de la información que se encuentra semánticamente estructurada [23].

La estructuración de los datos comprende dos modelos: orientado a objeto y basado en entidad-relación. Las ontologías utilizan el modelo orientado a objeto, ya que se estructura en clases, subclases y propiedades. De esta manera, un OBDB es una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto lleva a la necesidad de realizar un mapeo en la que una base de datos entidad-relación almacene información estructurada de forma orientada a objetos. Existen muchos modelos o esquema de mapeo para el mismo, tales como: la representación vertical y la representación binaria [24], en la que se combina representaciones verticales y horizontales; por otro lado, existen arquitecturas que emplean estos esquemas a través de un OBDB.

La consulta de la información almacenada en los OBDB puede ser realizada a través de lenguajes como SPARQL, RQL, OWL-QL, OntoQL y otros.

A continuación se describen algunos OBDBs que pueden ser utilizados.

3.1. RDF Suite

RDF Suite es una suite de herramientas para la validación, almacenamiento y consulta de ontologías y sus instancias que son descritas en RDF/RDFs [25]. De esta manera, los datos descritos en RDFs y las instancias de los datos en RDF serán representados en una base de datos relacional. Los datos en RDFs son almacenados en 4 tablas: clases, propiedades, subclases y subpropiedades. Para almacenar la información de los namespaces se utilizará otra tabla y las instancias de datos en RDF son guardadas en tablas separadas. Los *domain* y *range* que especifican una propiedad en las ontologías no son soportados por RDF Suite por lo que presenta poca flexibilidad y menos poder de expresión que otros lenguajes de ontologías como OWL. Por otro lado, RDF Suite utiliza el lenguaje de consulta RQL.

3.2. Sesame

Es una arquitectura genérica para el almacenamiento y consulta eficiente de una gran cantidad de metadatos estructurados en RDF/RDFs. El diseño e implementación de Sesame es independiente del modelo de estructura de datos, por lo que puede soportar base de datos relacional, *triple stores* y base de datos orientado a objeto [26]. Esto se debe a que presenta una API llamada SAIL (Storage and Inference Layer) el cual se encuentra en la capa superior del DBMS. De esta manera, el traslado de un método específico de RDF a cualquier método para el modelo de estructura de datos es responsabilidad del SAIL.

Esta arquitectura es diseñada para ser flexible y extensible en la utilización de más DBMS si fuese requerido. El lenguaje de consulta soportado es RQL. Sesame puede soportar el almacenamiento de *domain* y *range* dentro de las tablas en contraste con RDF Suite; sin embargo, ambos poseen esquema de base de datos similares cuando utiliza el modelo de estructura de datos objeto relacional. Los datos de RDF son guardados en tablas separadas.

Sesame, al utilizar RDF/RDFs, carece de poder de expresión de modelos ontológicos en comparación a otros lenguajes como OWL [26].

3.3 DLDB

Se define como una base de datos extensible relacional que soporta consultas de la semántica de la información utilizada en la Web semántica [27]. El diseño de DLDB se basa en la utilización de ontologías escritas en DAML+OIL que pueden ser almacenadas de manera híbrida, donde se combina la parte vertical (una simple tabla con un predicado, sujeto y objeto para cada columna) y la horizontal (una simple tabla con una columna para cada tipo y propiedad). Para esta arquitectura cada clase de la ontología es mapeada a una tabla independiente. Por consiguiente, cuando se guarda una gran cantidad de datos ontológicos, ésta podría representar un problema debido a la cantidad de tablas necesarias para la representación.

DLDB posee su propia implementación para las consultas a través de un conjunto de clases en Java; por lo que no necesita la utilización de ningún lenguaje específico. Estas clases mantienen el nombre de predicado y el número de términos, una vez que se ha ejecutado, se obtiene la traducción o el equivalente a las tablas y campos específicos que serán luego traducidos a consultas en lenguaje SQL; de esta manera, se ejecuta directamente en el DBMS la consulta generada en SQL del nombre del predicado y términos solicitados.

Esta arquitectura posee mayor capacidad de expresión que las implementaciones anteriores debido al lenguaje ontológico utilizado; sin embargo, es menos flexible ya que no acepta ningún lenguaje de consulta, creando una dependencia a la utilización del conjunto de clases.

3.4 OntoDB

OntoDB representa un modelo de OBDB relacional en la cual las ontologías y sus instancias son almacenadas a través de una estructura propia [24].

Este modelo propone dos estructuras para el almacenamiento: el enfoque de una sola tabla y el de doble esquema.

El primero utiliza tablas verticales para guardar la descripción de las clases, propiedades y sus instancias; en el segundo, el identificador de instancias representadas, propiedades y valores de las instancias son almacenadas en una tabla que posee tres columnas para cada valor mencionado.

El lenguaje ontológico utilizado es OWL por lo que posee mayor capacidad de expresión en la representación de la semántica y su lenguaje de consulta es el OntoQL que ha sido diseñado especialmente para este modelo.

Basado en OntoDB se propone OntoDB2 [28] el cual posee algunas mejoras como: permitir trabajar con varios modelos ontológicos en una misma estructura de base de datos.

3.5 OntoMS

OntoMS es un modelo para la representación de ontologías e instancias OWL en base de datos relacional [29]. Como lenguaje de consulta utiliza el OWL-QL, el cual está basado en patrones de consultas en forma de propiedades, sujeto y objeto.

OntoMS genera una clase basada en esquema relacional donde una relación es creada para cada clase. Cada clase relacionada contiene asociado tanto propiedades como atributos.

Referencias

- [1] E. K. Jacob, "Ontologies and the Semantic Web," Bulletin of the American Society for Information Science and Technology, pp. 19-22, 2003.
- [2] T. Sellis, D. Skoutas, and K. Staikos, "Database interoperability through Web Services and ontologies," 8th IEEE International Conference on Bioinformatics and BioEngineering, 2008.
- [3] W3C. (2004) Resource Description Framework (RDF). [Online]. Disponible en: <http://www.w3.org/RDF/>
- [4] Z. zhihong and Z. mingtian, "Web Ontology Language OWL and its description logic foundation," Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies PDCAT'2003., pp. 157-160, 2003.
- [5] OpenRDF. (2002) User Guide for Sesame 2.3. [Online]. <http://www.openrdf.org/doc/sesame2/2.3.2/users/userguide.html>
- [6] S. Jean, H. Dehainsala, D. N. Xuan, G. Pierra, L. Bellatreche and Y. Aït-Ameur, "OntoDB: It Is Time to Embed Your Domain Ontology in Your Database," Advances in Databases: Concepts, Systems and Applications Lecture Notes in Computer Science, vol. 4443/2010, pp. 1119-1122, 2010
- [7] W3C. (2005) Guía Breve de Web Semántica. [Online]. Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/websemantica>
- [8] M. Uschold, and M. Gruninger, "Ontologies: Principles, Methods and Applications", Knowledge Engineering Review, vol.11, 1996.
- [9] T.R. Gruber, "A translation approach to portable ontologies", Knowledge acquisition, vol.5, 1993, pp.199-220.
- [10] A. Gómez-Pérez, and D. Manzano-Macho, "An overview of methods and tools for ontology learning from text", The knowledge engineering review, vol.19, pp. 187-212, 2005.
- [11] M. Viinikkala. (2004) Ontology in Information Systems. [Online]. Disponible en: <http://www.cs.tut.fi/~kk/webstuff/Ontology.pdf>
- [12] R. Pedraza-Jiménez, L. Codina, and C. Rovira, "Web semántica y ontologías en el procesamiento de la información documental", El profesional de la información, vol.16, pp. 569-578, 2007.
- [13] E. P. Redondo, Y. H. Montero, E. Herrera-Viedma, and J.C. Herrera, "Ontologías, metadatos y agentes: recuperación semántica de la información", 2002.
- [14] A. Malucelli, "Combining ontologies and agents to help in solving the heterogeneity problem in e-commerce negotiations," Proceedings of International Workshop on Data Engineering Issues in E-Commerce, pp. 26-35, 2005.
- [15] W. Yanmei, "Ontology-based Web knowledge management," Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific Rim Conference on Multimedia vol. 3, pp. 1859-1863, 2003.

- [16] I. Zafraayes, L. Gu, F. Giunchiglia, W. Pan, Q. Ji, M. Chi and X. Huang, "From Web Developer to Collaborative Natural Language Processing Challenges," *The Semantic Web Lecture Notes in Computer Science*, vol. 4625, pp. 693-696, 2007.
- [17] M. Eghami, "Associational agents and ontology for semantic information search," *Third International Conference on Research Challenges in Information Science*, pp. 421-425, 2008.
- [18] W3C. (2008) W3C Semantic Web Activity. [Online]. Disponible en: <http://www.w3.org/2007/sem/>
- [19] T. Galla, D. Stizman, and K. Stalova, "Database interoperability through Web Services and ontologies," *5th IEEE International Conference on Distributed and Bio-Inspired Computing*, 2009.
- [20] A. Gomez-Perez and D. Corcho, "Ontology languages for the Semantic Web," *Intelligent Systems IEEE*, vol. 17, pp. 84-90, 2002.
- [21] D. L. McGuinness, "DAML+OIL: an ontology language for the Semantic Web," *Intelligent Systems IEEE*, vol. 17, pp. 72-80, 2002.
- [22] Stanford Center for Biomedical Informatics Research. (2010) The Protégé Ontology Editor and Knowledge Acquisition System. [Online]. Disponible en: <http://protege.stanford.edu/>.
- [23] H. Dehnavi, "Benchmarking data Schemes of Ontology based Databases," in *Proceeding of database system for advanced applications*, 2007.
- [24] H. Dehnavi, "DaidDB: An Ontology-Based Database for Data Intensive Applications," in *Proceeding of database system for advanced applications*, 2007.
- [25] S. Alcock, "The RDFSuite: Managing Voluminous RDF Description Bases," in *2nd International Workshop on the Semantic Web*, pp. 1-13, 2001.
- [26] J. Simola, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schemas", in *Proceedings of the first Intl Semantic Web Conference (ISWC 2002)*, vol. 2942, pp. 84-96, 2002.
- [27] Z. Pan and J. Hertz, "QLDB: Extending Relational Databases to Support Semantic Web Queries," 2003.
- [28] C. Fensler, "OntoDB2: Support of Multiple Ontology Models within Ontology Based Database," *Proceedings of the 2005 EDBT Ph.D. Workshop*, 2005.
- [29] M. J. Peris, "An Efficient and Scalable Management of Ontology," *Proceedings of the 12th International conference on Database systems for advanced applications*, 2007.

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
Sistema de Ingreso Universitario

Fechas de Inscripción:

- **1ra. Convocatoria**
9 de abril al 13 de julio de 2012
Aplicación de la prueba PAA: 21 de julio de 2012
- **2da. Convocatoria**
23 de julio al 14 de septiembre
Aplicación de la prueba PAA: 22 de septiembre de 2012

o comunicarse a través de:

● Bocas del Toro	706-8104/810-2306	www.fch.bocastorobocas.utp.ac.pa
● Azuero	302-9822/302-9994	www.fca.azuerobocas.utp.ac.pa
● Chiriquí	774-1000/774-1000/774-1000	www.fcc.chiriquibocas.utp.ac.pa
● Coclé	897-9123/897-7323	www.fcc.cocleabocas.utp.ac.pa
● Colón	472-9125/472-9925/472-9206	www.fcc.colonbocas.utp.ac.pa
● Veraguas	300-8007	www.fcv.veraguasbocas.utp.ac.pa
● Panamá Oeste	304-8807/304-8808	www.fco.panamabocas.utp.ac.pa
● Panamá Sur	300-9271/300-3209	www.fcs.panamabocas.utp.ac.pa
● Escuela de Telecálculo	309-8137/309-9234	www.fct.utp.ac.pa

Para mayor información ingrese a
www.utp.ac.pa

Simulación con Herramientas Open Source

Leonardo E. Fields M., Victor Sánchez U.

Fac. de Ingeniería Mecánica, Universidad Tecnológica de Panamá
leonardo.fields@utp.ac.pa, victor.sanchez@utp.ac.pa

Resumen- Este artículo brinda una visión general en cuanto a la simulación de entornos tridimensionales por computadora utilizando herramientas de código libre. La tecnología para simulación ha experimentado un crecimiento acelerado en los últimos años gracias a las exigencias del mundo del entretenimiento, entre otras cosas. La existencia de herramientas libres para crear este tipo de entornos resulta útil a la comunidad científica, investigadores, estudiantes y aficionados.

Palabras Claves- Computación, Simulación, Videojuegos, Código Abierto.

1. Introducción

La simulación de entornos 3D es un área de la computación que ha experimentado un crecimiento acelerado en los últimos años, debido entre otras cosas a las necesidades de la industria del entretenimiento. Esto ha ocasionado la aparición de múltiples herramientas comerciales y gratuitas, restringidas y de código libre que brindan soluciones de simulación a distintos niveles. Esta tecnología ha captado la atención de un grupo muy variado de individuos y entidades entre los que hay desde científicos hasta aficionados.

Existen muchos tipos de simuladores según su objetivo y el nivel de detalle de los mismos; pero a grandes rasgos, un simulador necesita:

- Selección del lenguaje de programación.
- Plataforma de desarrollo.
- Motor gráfico.
- Interacción con dispositivos de entrada y salida.

Otros componentes adicionales pueden incluir: motor físico, acceso a multimedia y acceso a redes.

2. Lenguaje

Existen tres lenguajes principales, ampliamente usados para la creación de simuladores: C/C++, Java y C#.

Java es un lenguaje orientado a objetos que utiliza una máquina virtual para compilar el código y luego hacerlo ejecutable. Es ampliamente usado en aplicaciones web, pero es muy criticado por la lentitud con que se inicializan las aplicaciones debido a su metodología de trabajo, problema que las últimas versiones de Java Virtual Machine han tratado de resolver.

C# es un lenguaje orientado a objetos derivado de C/C++, desarrollado por Microsoft para su plataforma NET. Es utilizado ampliamente en el desarrollo de videojuegos en el set de herramientas XNA de Microsoft.

C inició como un lenguaje estructurado, pero su predecesor C++ puede ser utilizado en aplicaciones orientadas a objetos.

Es un lenguaje universal presente en casi cualquier medio. Muchas aplicaciones de todos los niveles son escritas en C/C++, incluyendo simuladores, videojuegos y sistemas operativos.

3. Código Abierto

A diferencia del software libre, el código abierto no se concentra en las implicaciones filosóficas de compartir el código fuente, más bien en los beneficios prácticos de hacerlo. Aún así podemos encontrar muchas aplicaciones de software libre y de código abierto que comparten las mismas licencias.

Entre los beneficios prácticos de usar este tipo de herramientas tenemos:

- Posibilidad de estudiar y modificar el código fuente.
- La mayoría de las aplicaciones y código fuente se distribuyen gratuitamente.
- Amplias comunidades de usuarios activos que brindan respuestas a consultas.

4. Herramientas

Existen muchas formas distintas de crear un simulador, y muchas herramientas disponibles para hacerlo. Por su naturaleza, podemos clasificar las formas de hacer un simulador en dos grandes grupos: usar un motor de simulación o integrar librerías.

4.1 Motor de simulación

Es una sola aplicación que contiene todas las herramientas necesarias para crear simulaciones. Muy similares a los Motores de Videojuegos (Game Engine) que están orientados principalmente a la creación de videojuegos.

Un motor de simulación puede integrar varias herramientas diferentes como motor gráfico, motor físico y acceso a dispositivos de entrada y salida en una plataforma de alto nivel que facilite el desarrollo de proyectos en esta línea. Estos motores pueden contar además con una interfaz gráfica de usuario (para hacer el desarrollo menos orientado a la programación), y con un modelador 3D.

Cry Engine 3 y Unreal Engine son algunos motores de código cerrado para videojuegos que han sido la base de muchos videojuegos comerciales populares.

Blender es un modelador 3D gratuito de código abierto que puede utilizarse como motor para simulación y videojuegos. Tiene una interfaz poco intuitiva, pero sus múltiples características lo hacen muy utilizado.

4.2 Librerías

Algunos desarrolladores evitarán el uso de motores de simulación y videojuegos debido a que desean usar herramientas específicas, no necesitan hacer uso de todas las características que estos proveen, prefieren desarrollar sobre una aplicación más ligera o desean tener más control sobre el código fuente de su proyecto.

Existen muchas librerías de código libre que controlan partes específicas de la simulación, o varias de ellas, haciendo referencias a las mencionadas en la introducción de este documento. Estas variantes pueden incluir librerías que controlen:

- Plataforma de desarrollo.
- Video, audio.
- Video, audio, dispositivos E/S.

- Video, dispositivos E/S.
 - Leyes físicas.
- Algunas de estas librerías y sus funciones principales son:
- Plataformas de desarrollo: Qt, GTK+, wxWidgets, FOXTOOLKIT.
 - Motores gráficos: Ogre, Irrlicht, OpenSceneGraph.
 - Motores físicos: Open Dynamics Engine, Bullet Physics.
 - Programación de videojuegos: Allegro, SDL, SFML.

Si el desarrollador no desea construir los modelos 3D de la escena usando el motor gráfico directamente, puede hacer uso de modeladores 3D como Blender, Art of Illusion, K-3D.

5. Integración

Para poder utilizar distintas librerías en un solo simulador es necesario integrarlas. El grado de conocimiento necesario para realizar una integración depende de factores como si las herramientas están escritas en el mismo lenguaje, y la forma en que están construidas. No se logra de igual forma para todos los proyectos pues depende de las necesidades de los mismos.

La integración es posible debido a que las distintas herramientas tienen elementos en común que les permiten comunicarse. Por ejemplo los tipos de datos (números enteros, números flotantes y caracteres) y los punteros (si los permite el lenguaje). Incluso si las librerías estuviesen escritas en lenguajes distintos, muchas de ellas pueden trabajar con otros lenguajes mediante el uso de "bindings".

Delta3D es un ejemplo de un motor de simulación que integra distintas librerías, las cuales son: OpenSceneGraph, OpenDynamicsEngine, Character Animation Library, y OpenAL.

Otro ejemplo de integración es la de las librerías Qt, OpenSceneGraph, SDL y Bullet Physics; el cual se explicará en más detalle en los siguientes puntos.

5.1 Las herramientas

Qt es una librería escrita en C/C++ para el desarrollo de aplicaciones con o sin interfaz gráfica de usuario. Cuenta además con un completo entorno de desarrollo con interfaz gráfica de usuario, y complementa su lenguaje nativo C/C++ con métodos, clases y tipos de datos de alto nivel propios de QT.

OpenSceneGraph (OSG) es un motor 3D orientado a objetos que trabaja usando grafos de escena, lo cual evita al programador tener que utilizar rutinas de bajo nivel para construir el entorno. Soporta además varios formatos de modelos 3D y texturas.

SDL es una herramienta para desarrollo de videojuegos. No es un Game Engine, pero provee de varias funcionalidades necesarias para una simulación tales como: video 2D y 3D, audio, acceso a dispositivos E/S, acceso a imágenes, entre otras.

Bullet Physics es un motor físico. Permite simular el comportamiento de los objetos al ser sometidos a las leyes físicas. Permite simular colisiones entre objetos, restricciones, motores y soporta tanto objetos rígidos como blandos.

5.2 La plataforma y el motor gráfico

La ventaja de integrar un motor físico dentro de una plataforma de desarrollo es que se tiene acceso a todas las características de dicha plataforma y pueden utilizarse para robustecer el simulador. Qt posee un módulo de compatibilidad con OpenGL, que le permite integrar en una ventana hecha en Qt, un escenario tridimensional.

OpenSceneGraph es un derivado de alto nivel de OpenGL y cuenta con clases que le permiten integrar escenarios 3D en ventanas de otras herramientas. Existe un ejemplo de cómo integrar ambas herramientas en el sitio Web de OSG - véase la Figura 1.



Figura 1. Integración de Qt y OSG.

5.3 Dispositivos e/s y audio

Aunque tanto Qt como OSG tienen acceso a lecturas desde mouse y teclado, algunos simuladores pueden requerir acceso a otros dispositivos tales como los *joysticks*. Es por eso que para la comunicación con los dispositivos E/S se utiliza SDL.

Para lograr esta comunicación sólo es necesario saber el estado del dispositivo (hacia dónde se mueve o qué botón fue presionado). SDL puede retomar estos valores en forma de números o caracteres.

Reproducir audio también es posible desde SDL, y es recomendable hacerlo desde esta herramienta ya que es una de sus características principales.

5.4 Motor gráfico y motor físico

OSG y Bullet funcionan paralelamente en la ejecución de una aplicación, pero realizan tareas distintas. Uno dibuja el escenario, otro simula la física del mismo.

Para integrar ambas funciones primero debemos darle a todos los objetos del motor gráfico, un equivalente en el motor físico. Este equivalente no necesita ser idéntico en la forma, pero debe conservar un aspecto similar (Figura 2), tener la misma ubicación y orientación en el espacio.

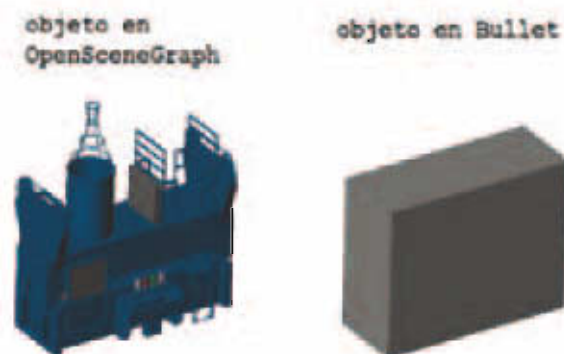


Figura 2. Objeto en OSG y su equivalente en Bullet.

Tanto los objetos de OSG como los de Bullet son colocados en el espacio mediante el uso de transformadas, las cuales son conjuntos de vectores que indican las coordenadas del objeto, su rotación, entre otras cosas.

Para igualar las transformadas de ambos objetos en tiempo de ejecución, puede utilizarse un `CallBack` de OSG que permite actualizar los valores de la transformada del objeto gráfico, según cambie la transformada del objeto físico. Un ejemplo de esta integración puede verse en el tutorial del sitio web de OSG, "Bullet demo" escrito por Jan Ciger.

6. Nivel de Detalle

Las simulaciones son aplicaciones que consumen gran cantidad de recursos de los computadores. Incluso hoy en día, las exigencias del mundo del entretenimiento ponen a prueba el desempeño de los equipos más modernos, para producir escenarios 3D de alto detalle y animaciones realistas.

Pero muchos desarrolladores y usuarios no tienen acceso a tecnología de última generación lo cual los obliga a usar de forma inteligente los recursos disponibles.

Los desarrolladores de simuladores pueden valerse de algunas técnicas sencillas para mejorar el desempeño de la aplicación en computadoras convencionales sin sacrificar mucho el nivel de detalle:

- Reducir el número de polígonos de los objetos 3D y compensar con texturas más detalladas.

- Reducir el detalle de los objetos en el motor físico y restringirse a usar sólo los necesarios.
- Reducir la frecuencia de actualización de la simulación.
- Asegurarse de optimizar el código mediante eliminar variables y punteros innecesarios; utilizar características de alto nivel de las herramientas y reutilizar.

7. El Hardware

Si se está por adquirir una computadora para desarrollar simuladores con herramientas open source, y se pretende desarrollar bajo un sistema operativo tipo Linux, se debe prestar atención al hardware que se utilizará.

Distribuciones de Linux como Ubuntu usan *drivers* de código abierto, y no los *drivers* que distribuye el fabricante del hardware, debido a que muchos de estos son exclusivos para sistemas Microsoft Windows. Esto puede causar problemas inesperados, especialmente cuando se trata de las tarjetas de video. Ati y Nvidia, algunos de los núcleos más populares para tarjetas de video tienen *drivers* tanto para Windows como Linux. Sin embargo, comentarios en múltiples foros en la web, parecen indicar que los núcleos Nvidia tienen por lo general un mejor desempeño en Linux.

Referencias

- [1] F. Leonardo, "Simulador de Grúas STS con herramientas Open Source", Tesis de Ingeniería, Universidad Tecnológica de Panamá, Panamá, Panamá, Mar. 2011.

Diccionario de Datos

Un enfoque semántico, de seguridad y usabilidad

Felipe McCalla, Iris González, Isabel Leguías,

Xavier Medianero, estudiantes

Victor López, MSc.

Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

felipe.mccalla@utp.ac.pa, iris.gonzalez@utp.ac.pa, isabel.leguias@utp.ac.pa

xavier.medianero@utp.ac.pa, victor.lopez@utp.ac.pa

Resumen- Este artículo trata sobre los diccionarios de datos enfocados en su utilidad semántica, léxica sobre las bases de datos y los esquemas relacionales; la manera en cómo debe ser estructurados y diseñados para su integración con los algoritmos ontológicos y con los sistemas de interfaces de usuarios. Además, el trabajo incluye la forma de extracción de datos implementando medidas de seguridad y a su vez medidas para la interpretación multilinguaje para la traducción del diccionario utilizando términos de base de datos. El artículo abstrae trabajos actuales sobre estas áreas, los integra y aporta nuevas ideas al tema.

Palabras Claves - Diccionario de Datos, Multilinguaje, Ontología.

1. Introducción

Las bases de datos tienen como propósito el servir de repositorio para datos, mediante la implementación de modelos de DBMS (Sistema Manejador de Base de Datos) se pueden representar los datos por diferentes modelos, como por ejemplo: modelo relacional, jerárquico, de redes o de objetos, que ayudan a capturar la heterogeneidad de los datos. Los datos almacenados pueden ser observados por diferentes usuarios (usuarios de múltiples culturas) con diferentes niveles de abstracción, los cuales no manejan la misma terminología pudiendo causar malas interpretaciones de los mismos [1]. Además, los sistemas actuales son grandes y complejos, difíciles de controlar y manejar; para facilitar estas tareas se utilizan los denominadas "Diccionarios de Datos" los cuales presentan las siguientes ventajas:

- Un diccionario de datos es un conjunto de metadatos que contienen las características lógicas y puntuales de los datos que se van a utilizar en el sistema, incluyendo descripción, alias, contenido y organización [2].
- Los diccionarios se desarrollan durante el análisis de flujo de datos y ayudan a los analistas que participan en la determinación de los requerimientos del sistema, evitando así las ambigüedades, su contenido también se emplea durante el diseño del proyecto. Los elementos más importantes son los flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos. Los diccionarios de datos son utilizados porque permiten [3]:
 - Manejar los detalles en sistemas muy grandes, ya que tienen enormes cantidades de datos, aún en los sistemas más chicos hay gran cantidad de datos.

- Facilitar el detalle de las características de las bases de datos relacionadas o del sistema en general y si son necesarias nuevas características.
- Determinar dónde efectuar cambios en el sistema.
- Localizar errores, omisiones en el sistema y detectar dificultades.
- Aplicar un significado y terminología común para todos los elementos del sistema. Los diccionarios de datos proporcionan asistencia para asegurar significados comunes para los elementos y actividades del sistema.

Dada la importancia que presentan los diccionarios de datos en la representación y análisis de sistemas para base de datos grandes y complejos, su influencia sobre el flujo y tratamiento semántico de los datos, además de las características y aplicaciones mencionadas, se reconoce el impacto que tienen estos diccionarios.

En la aplicación de las mismas, es fundamental la integración de elementos críticos que permite aumentar la importancia y la utilidad de los diccionarios de datos (sobre todo para usuarios de culturas diferentes para la que fue hecha; esto se debe a que dada la expansión de la web se puede acceder desde cualquier lugar a la información.) Estos elementos corresponden a la utilización de ontologías, multilinguaje, técnicas eficientes de extracción de datos y mecanismos de seguridad, de tal manera es posible aumentar los beneficios que proporcionan, expandiéndolos a:

- Agilización en procedimientos de extracción de dato [4].
- Mayor integridad e interoperabilidad con las bases de datos [5].
- Brindar la capacidad de soportar multilinguaje (capacidad de cambiar de idioma pero con un solo código) [6].
- Características ontológicas y semánticas para el significado de la información [7], [8].
- Características de protección de los datos del diccionario y a su vez de las bases de datos relacionadas.
- Diseño de Interfaces para la representación de los elementos de los diccionarios a los usuarios [9].

Este trabajo tiene como objetivo la integración de los elementos presentados, recopilando los beneficios, ventajas y facilidades de cada uno de ellos e integrándolos bajo un solo esquema; además de incorporar a cada componente características extras que permiten mejorar su utilidad en cada enfoque, para beneficio de una diversidad de usuarios siempre y cuando estos elementos presentados sean usables y amigables.

Este artículo está conformado de la siguiente forma: se explican los trabajos relacionados sobre las diferentes áreas a incorporar, posteriormente se integran aportando ideas para el mejoramiento de los diccionarios en cada área. Luego se discuten las ventajas enfocando la importancia de cada elemento.

2. Trabajos Relacionados

Debido a la importancia de la utilización de Diccionarios de Datos para facilitar el control de los datos tanto como su estructura, organización, significado semántico y seguridad de los mismos, se han elaborado diferentes trabajos, entre ellos los presentados a continuación:

2.1. Diccionario con Terminologías Multilinguaje

Lee y Park en [6], proponen métodos para remover la ambigüedad

en la interpretación de consultas en lenguaje natural a través de un *framework* denominado Gramática Categórica Combinatoria (CCG).

El método que utiliza CCG consiste en combinaciones extendidas de gramáticas categorizadas basadas en reglas de descomposición léxica y reducción en partes.

Presenta dos modelos de traducción: uno directo y uno indirecto, la representación en estos modelos no debe coincidir con objetos o elementos de la base de datos. La traducción basada en CCG deriva las expresiones base del lenguaje objetivo, ya sean instrucciones SQL, TSQL, lenguaje intermedio a partir del lenguaje base. El procedimiento de traducción utiliza una representación intermedia, lo cual proporciona diferentes ventajas, entre ellas: la independencia lingüística, la separación por dominios de conocimiento, la facilidad para determinar las inferencias y la reducción de la ambigüedad.

El *framework* CCG permite una traducción directa ente dos lenguajes, además de apoyarse en una representación intermedia. Las consultas que se realizan finalmente pueden ser formuladas en múltiples lenguajes proporcionándolo en el lenguaje base.

2.2. Interoperabilidad con Base de Datos

La interoperabilidad de los sistemas de base de datos está cobrando más importancia, debido a que las organizaciones pueden aumentar el número de sistemas operativos y añadir nuevos sistemas de apoyo a las decisiones [7],[5]. La construcción, operación y mantenimiento de estos sistemas es complicado, el tiempo y la complejidad crece más rápidamente si aumenta el número de sistemas [10].

El mayor problema de interoperabilidad consiste en combinar dos o más sistemas de bases de datos en un sistema coherente e integrado; ya que proporcionar la interoperabilidad entre los sistemas es mucho más difícil que la construcción de un sistema de base de datos distribuido.

En [5] se propone la integración automatizada por medio de la captura de datos utilizando la semántica de un diccionario normalizado. Esta arquitectura incorpora vistas locales definidas independientemente de la semántica de la base de datos, utilizando únicamente un diccionario predefinido entre sitios de integración. Mediante la normalización del diccionario en forma de cláusulas, se eliminan los conflictos de nombre, al igual que se reducen los conflictos semánticos.

La Base de Datos Semántica captura de forma independiente los documentos XML que son llamados X-Specs en los cuales se almacenan los nombres semánticos para los elementos de esquemas que determinan conceptos iguales en los sistemas. Luego se construye una vista integrada de conceptos que es transparente a las consultas que realiza el usuario. El procesador de consulta traduce las consultas semánticas a expresiones estructurales e integra los resultados.

2.3. Extracción Morfema usando Diccionarios

En [4], Nakamura y Yukishita investigan la extracción morfema de una cadena de fonemas por medio de las características de los diccionarios de datos. Para realizar la extracción de morfema, el diccionario de datos utiliza un método de acceso, una estructura de árbol de índice y una técnica en paralelo de índice. Los resultados obtenidos de estas técnicas son muy importantes para la prevención

del cuello de botella en el procesamiento del lenguaje natural que se utilizan en la entrada de los datos, al igual que en la extracción de morfemas.

2.4. Interface para la Usabilidad de Diccionarios

En los trabajos presentados en [9] enfocan la importancia que presentan las interfaces dentro de los diccionarios de datos. Se establece un modelo basado en redes jerárquicas en el cual la información se relaciona por significado léxico con otra información del mismo diccionario, en donde se asocian todas las dependencias de un elemento padre con sus elementos hijos.

Cada red jerárquica contiene niveles de abstracción para cada uno de sus elementos. Un elemento identificado como tigre, se relaciona en niveles diferentes de transparencia con un elemento gato; el objetivo principal es la traducción de lenguajes determinando dinámicamente qué nodos y cuáles caminos conectan con la información que se desea trasladar.

2.5. Diccionarios Léxicos

La investigación en [8] presenta un sistema experto llamado lexicográfico cuyo objetivo es de suplir al usuario con diferentes informaciones sobre las palabras rusas, incluyendo información bibliográfica concernientes a artículos del léxico individual. El sistema se concibe como una ayuda tanto en el ámbito del procesamiento del lenguaje natural y en la lexicografía tradicional, componiéndose así de dos componentes básicos: una base de datos bibliográfica y un lenguaje.

Las bases de datos léxicas son consideradas como un instrumento para predecir y calcular todo tipo de palabras de tipo semánticas. La base de datos léxica muestra una ventaja si la comparamos con un diccionario tradicional ya que una base de datos posibilita la presentación semántica de los datos utilizando un formato que facilita al computador arrojar una lista de palabras que poseen las mismas características comúnmente.

2.6. Seguridad en Base de Datos

Actualmente las bases de datos han evolucionado en diferentes aspectos y esto incluye la seguridad. La seguridad en la base de datos [11] está basada en tres aspectos importantes: confidencialidad, integridad y disponibilidad. Además de estos aspectos hay otros que deben ser tomados en cuenta, como control de acceso, acceso a las aplicaciones, vulnerabilidades y los mecanismos de auditoría.

El método que generalmente se utiliza para brindar seguridad a los datos es restringir el acceso a los mismos, y se realiza a través del control de acceso, autenticación y autorización. Aunque estos mecanismos son diferentes, se utilizan en combinación con el control de acceso de granularidad para la asignación de derechos a objetos y usuarios. Dentro de una base de datos, estos objetos pueden ser tablas, vistas, filas y columnas. La limitación del acceso a los objetos se realiza a través de mecanismos de control de acceso Grant/Revoke. El control de acceso se especifica de tres maneras: Control de Acceso Obligatorio (MAC), Control de Acceso Discrecional (DAC) y Control de Acceso basado en Roles (RBAC). Tanto MAC como DAC proporcionan privilegios a los usuarios como a grupos asignados. Los roles son similares a las funciones

de trabajo. El objetivo principal de las funciones es la identificación de las operaciones y los objetos a los que estas operaciones necesitan tener acceso.

Políticas de Seguridad de base de datos

Las políticas de seguridad proporcionan una serie de directrices que soportan y orientan el proceso de seguridad en las bases de datos [12].

Las políticas sobre administración de seguridad en donde se destacan dos puntos importantes: el control centralizado, donde un solo administrador o grupo controla todos los aspectos de seguridad de la base de datos vs control descentralizado, diferentes administradores tienen control sobre diferentes partes de la BD, frecuentemente siguiendo lineamientos que se aplican a toda la BD. En el caso de propietario vs administrador, el propietario muchas veces se considera el responsable de los datos, pero cuando las BD son compartidas se requiere de un administrador cuyo objetivo es definir los datos compartidos por los usuarios y los controles de su uso.

Políticas para la especificación del control de acceso

Entre las cuales se encuentra las políticas del menor privilegio, máxima compartición de datos, sistemas abiertos y cerrados, control de acceso independiente del nombre, tipos de acceso, control independiente de la historia.

3. Discusión

Los diccionarios de datos tienen un papel relevante en el detalle y representación de los datos, detección de problemas y asociación de terminologías en relación a las bases de datos a las cuales están asociados. Su utilidad es incrementada cuando, en su diseño, se incorporan las siguientes características:

Seguridad: Debido a que en un diccionario de datos, los datos representan información del sistema que cubren, cualquier ataque que pueda revelar datos e información del mismo, proporcionará de forma indirecta información de cómo es el sistema, lo cual permitirá realizar un ataque más concreto y directo sobre las base de datos del sistema. De este hecho radica la importancia de asegurar la información y el acceso hacia los diccionarios de datos.

Usabilidad: Debido a que los datos presentes en los diccionarios de datos deben ser analizados y estudiados, las interfaces por las cuales se accede a esta información deben permitir el control completo de los datos, por lo que en este procedimiento deben aplicarse también los patrones de diseño de interfaces, para garantizar la usabilidad.

Además, es importante resaltar que los datos que se almacenan en los diccionarios de datos deben representar información relevante sobre las características e información extra que se necesite en el sistema.

Multilinguaje: La creación de diccionarios multilingües es muy valiosa ya que son una poderosa herramienta para almacenar datos. Mejor aún, si se contemplan varios idiomas para el mismo diccionario se podría entonces utilizar por una diversidad de usuario. Para lograr una producción de calidad en el diccionario multilinguaje es propicio integrar las herramientas informáticas dominado por lexicógrafos, que son esencialmente los procesadores de palabras.

Semántica: Los elementos almacenados deben otorgar un valor semántico al diccionario de datos como se muestra en la sección de usabilidad. De esta forma mediante las interfaces no sólo se buscarán elementos puntuales (léxicos), sino que facilitará la búsqueda ontológica de elementos (semántica) lo cual permitirá al usuario la búsqueda de otras características.

Interoperabilidad: Otro elemento a considerar dentro de los diccionarios, es la interoperabilidad que existe entre el mismo y las bases de datos. Por eso en [5] se menciona que la integración de fuentes de datos implica la combinación de sus conceptos y el conocimiento en una visión integrada que aísla a los usuarios de la organización del sistema. Sin embargo, en [7] señalan los criterios y técnicas a seguir para apoyar el establecimiento de un diccionario semántico.

Las características presentadas, aumentan la utilidad que presentan los diccionarios de datos e incorporan ventajas extras a la aplicación de diccionarios dentro de un sistema.

El tratamiento y la traducción de frases o sentencias en lenguaje natural a otros lenguajes, es una de las tareas actuales que deben ser incorporadas dentro de los sistemas debido al fenómeno moderno de internalización de la información.

Procedimientos de descomposición léxica del idioma base en elementos indicadores como sujetos, predicados y verbos, en conjunto con el valor semántico que pueden agregar los mismos permite que el proceso de traducción sea más eficiente pues se identifican las capas de las mismas.

Además, es necesario manifestar que en las Interfaces de usuarios no sólo es importante la información que se va a presentar y la usabilidad de la misma, sino que también es necesario aplicar patrones de multiculturalidad tanto en las Interfaces, como al momento de manipulación de los datos del diccionario, debido a que el significado de los mismos puede variar el sentido real causando inconsistencia.

4. Comentarios Finales

Cuando un sistema amerita la integración de Diccionarios de Datos, no sólo se deben enfocar los aspectos y ventajas propias del mismo como el detalle de las características del sistema, localización de errores, entre otros; sino también incluir aspectos de seguridad para protección de datos de forma directa e indirecta, es decir, proteger los datos del diccionario y cualquier información que se pueda obtener del sistema a través de la visualización de los mismos, aspectos de multilinguaje orientados a multiculturalidad, de tal forma que la información recabada en los diccionarios sea más útil para los actores que utilizan la misma, sin olvidar que las interfaces de consulta a los mismos también deben integrar esos patrones.

Además, la efectividad de la utilización de los diccionarios, puede ser aumentada incorporando técnicas semánticas y ontológicas, pues permite no sólo captar datos aislados, sino comprender el objetivo de los mismos, por lo que también añadirá ventajas en búsqueda y consultas.

Referencias

- [1] C. Bartini, et al., Diseño Conceptual de Base de Datos. Massachusetts, United States Addison-Wesley Iberoamericana, S.A., 1994.
- [2] A. Silberschatz, et al., Fundamentos de Base de Datos, 4 ed., 2001.
- [3] M. P. Meza. (Consultado: 5 de octubre de 2010, Tutorial de Administración de Base de Datos. Available: http://sistemas.itp.edu.mx/tutoriales/admonbasedat/tema5_1.htm
- [4] O. Nakamura and M. Yikishita, "A High-speed Morpheme-Extraction System Using Dictionary Database," in Proceedings. Fourth International Conference Data Engineering, Los Angeles, CA, 1988, pp. 488-495.
- [5] R. Lawrence and K. Baker, "Integrating Relational Database Schemes using a Standardized Dictionary," in Simposio sobre Informática Aplicada Actas del simposio de ACM New York, EE.UU., 2001, pp. 225-230.
- [6] H. Lee and J. Parque, "Automatic Augmentation of Translation Dictionary with Database Terminologies in Multilingual Query Interpretation," in Actas del taller sobre Tecnología del Lenguaje Humano y Gestión del Conocimiento, Toulouse, Francia, 2001.
- [7] S. Castano and V. D. Antonellis, "Semantic Dictionary Design for Database Interoperability," in Data Engineering, Proceedings, 13th International Conference Birmingham, UK, 1997, pp. 43-54.
- [8] E. Paducheva, et al., "Semantic dictionary viewed as a lexical database," in Proceedings of the 14th conference on Computational linguistics, Actas de COLING-92, Nantes, Francia, 1992, pp. 1294-1298.
- [9] H. Ozawa, et al., "DIS: A User Interface System design for the dictionary's database," Languages for Automation: Symbiotic and Intelligent Robots, IEEE Workshop pp. 164-169, 1988.
- [10] R. Lawrence, "Automatic Conflict Resolution to Integrate Relational Schema," Ph.D, Doctor of Philosophy in Computer Science, UNIVERSITY of MANITOBA, Canada, 2001.
- [11] M. Coffin, "Database Security: What Students Need to Know," Journal of Information Technology Education, vol. 9-2010, 2010.
- [12] M. López, "Diccionario/Directorio y Seguridad de Datos," Centro de investigación en Sistemas de Información.

LACCEI 2012

Participa de la:

10ª Conferencia Internacional del Consorcio Latinoamericano y del Caribe de Escuelas de Ingeniería

MEGAPROYECTOS:

Construyendo Infraestructura mediante colaboración en:

- ▶ Ingeniería
- ▶ Integración eficiente y efectiva
- ▶ Planificación Innovadora

"Educación, Innovación, Tecnología, Diseño y Práctica"

Del 23 al 27 de julio de 2012

Organizado por:
 LACCEI - Consorcio Latinoamericano y del Caribe de Escuelas de Ingeniería
 UTP - Universidad Tecnológica de Panamá

www.laccei2012panama.org.pa

Evaluación de la Eficiencia del Algoritmo de Kruskal para la Construcción de Árboles Cobertores Mínimos

María T. Ortega, Mario A. Ramos, estudiantes

Universidad Tecnológica de Panamá

maria.ortega2@utp.ac.pa, mario.ramos@utp.ac.pa

Resumen- Este artículo brinda un análisis de la eficiencia de cuatro variantes del algoritmo de Kruskal para la construcción de árboles cobertores mínimos de un grafo dado y el impacto de la utilización de algoritmos de búsqueda óptimos en el rendimiento esperado del algoritmo.

Palabras Clave - Árbol cobertor mínimo, Algoritmos, Eficiencia, IQS, Kruskal, QuickSort.

1. Introducción

El desarrollo de software ha tenido un gran crecimiento en las últimas décadas, apoyándose en gran parte de los avances en las tecnologías de construcción de computadores. Esto ha permitido la creación de algoritmos y programas que resuelven problemáticas en diferentes dominios, apoyándose en la capacidad de procesamiento de los computadores y dejando de lado la eficiencia del propio algoritmo. Sin embargo, buenos algoritmos implementados, haciendo énfasis en la eficiencia, permiten sacar mayor partido al recurso computacional.

Una forma de evaluar la importancia de la implementación de algoritmos eficientes para la solución de una problemática en particular, es utilizar diferentes algoritmos que resuelvan la misma problemática y comparar sus características de tiempo de ejecución, consumo de recursos, complejidad de implementación, entre otras; lo cual puede servir de métrica para decidir la opción que mejor se adecúe a las necesidades requeridas.

Si se tienen algoritmos que resuelven la misma problemática de manera distinta, ¿cómo decidir cual algoritmo es mejor?, posibles respuestas a esta interrogante serían: el que conlleve menor tiempo de ejecución, menor complejidad de implementación y/o menor consumo de recursos.

Uno de los problemas clásicos de las ciencias de la computación consiste en la búsqueda del árbol de cobertura de costo mínimo (MST, *Minimum Spanning Tree*) de un grafo conexo, no dirigido y con aristas con peso $G(V,E)$, en que V corresponde al conjunto de nodos y E al de aristas (o arcos). En este breve artículo se analizarán experimentalmente algunas soluciones para este problema.

2. Árbol Cobertor Mínimo

Sea un grafo conexo, no dirigido y con aristas con peso $G(V,E)$, en que V corresponde al conjunto de nodos y E al de aristas.

El árbol de cobertura de costo mínimo (también llamado árbol de cobertura de mínimo costo) es un árbol que conecta a todos los vértices en V mediante las aristas contenidas en E con un costo total mínimo [1]. Siguiendo la notación estándar se considerará que los tamaños de los conjuntos de nodos y aristas son $n = |V|$ y $m = |E|$, respectivamente.

El MST es un árbol porque es conexo y acíclico, es de cobertura porque cubre todos los vértices y es de costo mínimo porque la suma de los costos de las aristas del MST es la menor posible.

En general, cualquier grafo no dirigido (sea conexo o no) tiene un bosque de cobertura de costo mínimo. Este bosque está compuesto por la unión de los distintos MSTs de cada una de las componentes del grafo procesadas independientemente.

El MST puede no ser único para un grafo G dado. Note que el número de aristas del MST es $|V| - 1$, por lo que un grafo con una alta densidad de aristas puede tener múltiples árboles cobertores mínimos de igual coste total.

Los algoritmos básicos para resolver este problema son de tipo *greedy* (algoritmos avaros), y los más conocidos son el algoritmo de Kruskal y el de Prim [1]. Usando el algoritmo de Prim, el cálculo del MST toma tiempo $O(n^2)$, lo cual lo hace más conveniente para grafos densos. A su vez, el algoritmo de Kruskal permite calcular el MST en tiempo $O(m \log m)$, que es apropiado para grafos dispersos. A continuación nos concentraremos en el algoritmo de Kruskal y algunas optimizaciones que se le pueden hacer de modo de mejorar su desempeño incluso en el caso de grafos densos.

3. Algoritmo de Kruskal

El algoritmo de Kruskal para construir árboles cobertores mínimos inicia su funcionamiento con n componentes que contienen un solo nodo, que son mezclados hasta producir una única componente conexa. Para la creación de esta única componente conexa, el algoritmo de Kruskal utiliza un par de estructuras auxiliares: la primera es C , que es una instancia del Tipo de Datos Abstractos (TDA) *UnionFind* [1], la que será explicada a continuación. La segunda estructura auxiliar es ACM que se inicializa en (V, \emptyset) , esto es un bosque de n árboles de un nodo, que representa al MST en construcción.

Kruskal opera iterativamente, y en cada iteración verifica si es posible insertar una nueva arista. El criterio de selección de arcos es escoger incrementalmente el arco de menor costo en E tal que no produce un ciclo con las aristas previamente almacenadas en ACM; esto significa, que sólo agrega arcos cuyos vértices pertenezcan a dos componentes distintas (es decir, dos árboles distintos del bosque). Una vez agregada la arista ambas componentes se fusionan en una sola (restando un árbol al bosque). Cuando el proceso termina, la estructura ACM contiene al árbol cobertor mínimo del grafo $G(V,E)$ como se muestra en la Figura 1.

El TDA *UnionFind* permite administrar eficientemente una colección de conjuntos disjuntos permitiendo consultar a que conjunto pertenece un elemento dado (*Find*) o unir dos conjuntos distintos (*Union*). Durante la ejecución de Kruskal, se tiene exactamente este caso. Se inicia con conjuntos disjuntos y cada vez que se analiza una arista se determina si sus vértices pertenecen a componentes distintas, es decir, a conjuntos distintos para luego unir las componentes.

```

Kruskal (Grafo G(V,E))
UnionFind C ← {[v] | v ∈ V} // el conjunto de las componentes conexas
ACM ← ∅ // árbol coberter mínimo
Ordenar las aristas de E en orden creciente de peso
while (C) > 1 do
  Sea e = (u,v) la siguiente arista en orden creciente de peso
  if C.find(u) ≠ C.find(v) then
    ACM ← ACM ∪ {e}
    C.union(u, v)
return ACM
    
```

Figura 1. Algoritmo de Kruskal Básico. Adaptado de [1]

La versión estándar del TDA UnionFind se presenta en la Figura 2. Con esta estructura, la operación Find consiste en seguir los punteros (hacia el padre) hasta llegar a la raíz del árbol al que pertenece el nodo. En este caso la raíz es un nodo que se apunta a sí mismo. Una vez que se conoce la raíz, se puede actualizar el puntero de cada nodo en el camino hacia la raíz para dejarlo apuntando directamente a la raíz, lo que se conoce como compresión de caminos [1].

La operación Union fusiona las componentes. En donde el peor caso posible podría producir caminos de largo proporcional a la cantidad de nodos. Para prohibir la ocurrencia del peor caso, la idea es garantizar que la altura de los árboles sea logarítmica es decir, que contenga caminos relativamente cortos. La estrategia de fusión consiste en hacer que la componente que tenga menos elementos quede como hija del representante de la componente más numerosa. Tras fusionar se actualiza el tamaño de la componente resultante.

La implementación estándar de la estructura UnionFind considera dos valores (r,c) para cada identificador de nodo (Figura 2), en que r es el representante del nodo y c indica la cantidad de elementos en la componente.

4. Optimizaciones del Algoritmo de Kruskal

Para propósitos de este estudio se utilizarán cuatro variantes eficientes del algoritmo de Kruskal, a saber, Kruskal0, Kruskal1, Kruskal2 y Kruskal3. Kruskal0 corresponde a la implementación básica de Kruskal (Figura 1) en que las aristas se ordenan con Quicksort (Figura 4), se utiliza la versión de UnionFind mostrada en la Figura 2, y se deja explícita la condición de término del ciclo While con una operación que es más fácil de codificar. El pseudocódigo de la versión básica se muestra en la Figura 3.

La segunda variante del Kruskal, denominada Kruskal1, incorpora el algoritmo de ordenación incremental IQS (Incremental QuickSort) (Figura 5) [2]. IQS permite obtener los primeros k elementos en orden creciente de un conjunto desordenado A de tamaño m para cualquier valor de $k \leq m$, en tiempo óptimo $O(m + k \log k)$.

El algoritmo Kruskal1 al igual que Kruskal0 hace uso de la estructura estándar UnionFind (Figura 2). La variante del algoritmo de Kruskal1 con las adecuaciones necesarias para utilizar el algoritmo IQS se muestra en la Figura 6.

La tercera y cuarta variante del algoritmo de Kruskal, denominadas Kruskal2 y Kruskal3, utilizan los algoritmos de búsqueda Quicksort

```

UnionFind (int n) // inicializa la estructura
1  int[] r ← int[n], c ← int[n] // basta con dos arreglos
2  For i ← 0 to n-1 Do r[i] ← i, c[i] ← 1

Find(int x) // calcula el representante de x y comprime el camino
// busca el representante y comprimo camino
1  if x = r[x] Then r[x] ← Find(r[x])
2  Return(x)

Union (int x, int y) // asume que x e y son componentes distintas
// y tiene más nodos, luego y sigue como
1  if r[x] < c[y] Then //raíz y actualizo su tamaño
2  r[x] ← y, c[y] ← c[x] + c[y]
3  Else //raíz, x es igual o más numerosa que la de y, luego x
//sigue como raíz y actualizo su tamaño
4  r[y] ← x, c[x] ← c[x] + c[y]
    
```

Figura 2. Estructura UnionFind estándar. Adaptado de [1]

```

Kruskal0 (Grafo G(V,E))
UnionFind C ← {[v] | v ∈ V}
max ← 0 //el árbol coberter mínimo en construcción
QuickSort (E, 0, E-1)
While [max] < n - 1 Do //inserta n-1 aristas
  (e = (u,v)) ← E[k], k ← k + 1
  if C.find(u) ≠ C.find(v) Then
    max ← max ∪ {e}, C.union(u,v)
Return max
    
```

Figura 3. Algoritmo Kruskal0

```

QuickSort (A, p, r)
if p < r then
q=Particion(A,p,r)
QuickSort(A,p,q)
QuickSort(A,q+1,r)
    
```

Figura 4. Algoritmo QuickSort [3]

```

IQS (Ser A, índice de Stack S)
//idx es el índice del elemento que queremos recuperar
//Precondición: idx ≤ S.top()
1  if idx = S.top() Then S.pop() Return A[idx]
2  idx ← random[idx, S.top()-1]
3  idx' ← partition(A, A[idx], idx, S.top()-1)
4  S.push(idx')
5  Return IQS(A, idx, S)
    
```

Figura 5. Algoritmo Incremental IQS [2]

(Figura 4) e IQS (Figura 5) respectivamente, pero utilizan una implementación alternativa de la estructura UnionFind (Figura 7). Esta variante utiliza una variante no recursiva de la función Find, para ello utiliza una Pila que permita actualizar los índices de los representantes de cada nodo.


```

Kruskal (Graph G(V,E))
UnionFind C ← ([v] v ∈ V)
mst ← ∅ // el árbol cobertor mínimo en construcción
Stack S: S.push(E); k ← 0
While (mst) < n - 1 Do // inserta n-1 aristas
    (e = (u,v)) ← QS(E,k,S); k ← k + 1
    If C.Find(u) ≠ C.Find(v) Then
        mst ← mst ∪ {e}; C.Union(u,v)
Return mst
    
```

Figura 6. Algoritmo Kruskal1

```

FindInd (int x)
    While(x!=r[x])
        S.Agregar(x)
    x ← r[x] then
        While(S.Top() != null)
            r[S.Sacar]=x
    retorna x;
    
```

Figura 7. Estructura alternativa de Find. Adaptado de [1]

5. Metodología Experimental

En este artículo se evalúa el impacto de implementaciones eficientes de las estructuras UnionFind en conjunto con los algoritmos de búsqueda en los tiempos de construcción del árbol cobertor mínimo utilizando el algoritmo de Kruskal.

Para el estudio experimental se utilizarán grafos Euclidianos conexos no dirigidos representados bajo una lista de arcos.

Los grafos Euclidianos son grafos ponderados en los cuales, los vértices están conformados por puntos pertenecientes al plano Euclidiano R² [4], y cuyas aristas son ponderadas en relación a las distancias Euclidianas entre dos vértices del grafo.

Para propósitos de este estudio se considera un grafo euclidiano no dirigido G(V,E), como un conjunto de puntos (x,y) dentro del cuadrado unitario [0,1] × [0,1] donde el costo de los arcos del grafo equivale a la distancia euclidiana d (1) entre dos vértices v, denotados por vk(xk,yk), del grafo dado. Los arcos considerados en el grafo son elegidos aleatoriamente, iniciando con grafos con baja densidad de arcos hasta obtener el grafo completo.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

El tamaño del conjunto de arcos E (2) pertenecientes a un grafo G, estará delimitado por la densidad de arcos p, en donde un grafo con una densidad p = 0.01 contendrá el 1% del total de arcos posibles, mientras que un grafo con densidad p = 1 contendrá el 100% del total de arcos posibles es decir, el grafo completo.

$$|E| = p \frac{n(n-1)}{2} \tag{2}$$

El análisis realizado en este artículo se basa en un estudio comparativo de las cuatro variantes del algoritmo Kruskal, explorando distintos valores de densidad de arcos y número de nodos. Se midió el tiempo de construcción del MST para cada una de las versiones de Kruskal, identificando la más eficiente en tiempo de ejecución.

6. Resultados y Conclusiones

Las diversas variantes del algoritmo de Kruskal, junto con la representación del grafo Euclidiano, fueron implementadas en el lenguaje Java, y las pruebas experimentales se realizaron en una máquina con las siguientes especificaciones: procesador Intel Pentium M 740 2.66GHZ, memoria 2 GB DDR2 RAM, sistema operativo Windows 7 Ultimate, Java SE Development Kit 6 Update 17, NetBeans IDE 6.7.1 y se configuró la máquina virtual de java con un heap size de = -Xms500m -Xmx1536m.

Se realizaron pruebas con densidad de arcos p correspondientes a 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.0, y una cantidad de nodos variable de 1,000 a 10,000 nodos. Las pruebas ejecutadas con densidades de 0.01 a 0.16 fueron ejecutadas hasta un máximo de 10,000 nodos; para la densidad de 0.32 las pruebas alcanzaron un tope con 7,000 nodos, con la densidad 0.64 se alcanzó ejecutar hasta un máximo de 5,000 nodos y con la densidad de 1.0 alcanzó el límite con 4,000 nodos. Estos límites corresponden al máximo nivel de memoria y procesamiento soportados utilizando un computador con las especificaciones anteriormente mencionadas.

La Figura 8 muestra los resultados obtenidos de la prueba de las cuatro variantes de Kruskal, con una densidad de 0.01 y valores de n variables de 1,000 a 10,000 nodos con un incremento de 1,000. La gráfica muestra una representación por tiempo vs nodos, ponderados en valores de milisegundos y múltiplos de 1,000 respectivamente. Las implementaciones Kruskal 1 y Kruskal 3 son notablemente más rápidas que las implementaciones de Kruskal 0 y Kruskal 2. Esto se debe a la influencia directa del algoritmo de ordenamiento utilizado, puesto que Kruskal 1 y 3, utilizan el algoritmo IQS, el cual es más eficiente que el algoritmo Quicksort para ordenar las aristas que son necesarias para construir el MST.

Otro punto importante es comparar las diferencias entre las implementaciones de la estructura UnionFind. Para ello se muestran los resultados obtenidos de la comparación de los algoritmos Kruskal1 y Kruskal3 con densidad de 0.32 (Figura 9). Puesto que ambos utilizan el algoritmo de búsqueda IQS, la pequeña diferencia de tiempos se debe a la implementación de la estructura UnionFind. Puesto que el algoritmo Kruskal3 utiliza la implementación alternativa FindInd, puede concluirse que esta variante es más eficiente que la estructura estándar.

Las implementaciones de Kruskal 1 y 3, resultaron ser las más eficientes debido a la utilización del algoritmo incremental IQS. En el peor de los casos, cuando se necesita ordenar el conjunto completo, el costo derivado de ejecutar IQS resulta ser de O(m logm). Sin embargo, debido a que en la práctica Kruskal sólo utiliza una cantidad pequeña de aristas para conectar a los nodos (en efecto, de acuerdo a [5], sólo requiere revisar las primeras k = O(n ln(n)) aristas) en promedio IQS consume O(m + log2n) [2], que en la práctica es bastante menor que O(m logm).



Figura 8. Resultados de prueba con densidad de 0.01

Las tendencias observadas en las pruebas con cada una de las densidades de arcos y con cada bloque de nodos utilizado denotan que los tiempos menores de creación del MST corresponden a las implementaciones que utilizan el IQS (Fig. 9) como algoritmo de ordenamiento, siendo el ordenamiento la operación de mayor costo del algoritmo y un punto importante de optimización. Esto es consistente con los resultados obtenidos en [6], en donde también IQS muestra el mejor resultado para este tipo de aplicaciones.

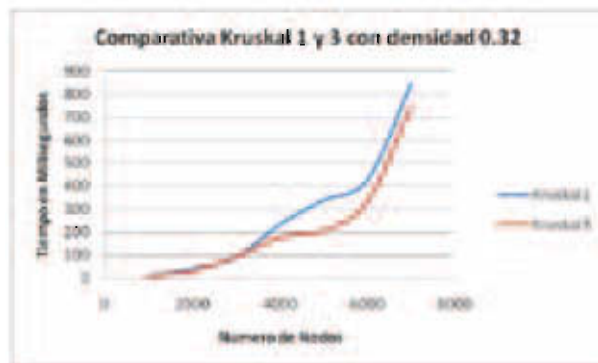


Figura 9. Resultados de prueba con densidad 0.32

Las implementaciones de las estructuras UnionFind, influyen en menor grado en la eficiencia de los algoritmos siendo ligeramente más eficientes los que utilizan la estructura alternativa no recursiva

(FindInd) a la compresión de caminos estándar (UnionFind) siempre y cuando utilicen el mismo algoritmo de ordenamiento. Esta tendencia pudo observarse a lo largo de las pruebas para todas las densidades de arcos y cantidades de nodos (vértices), tal como se muestra en la Figura 10.



Figura 10. Resultados de pruebas con densidad 1.0

Agradecimientos

Deseamos agradecer al Profesor Rodrigo Paredes por sus comentarios sobre este trabajo y su colaboración en la redacción del presente artículo.

Referencias

- [1] T. H. Cormen, C. E. Leiserson and R. L. Rivest, and C. Stein. "Introduction to Algorithms". 2^a ed. McGraw Hill 2001.
- [2] R. Paredes. "Graphs for Metric Space Searching". Universidad de Chile, Santiago, Chile, Julio 2008.
- [3] R. Guerequeta and A. Vallecillo, Técnicas de diseño de algoritmos, 2a ed., Publicaciones de la Universidad de Málaga, Mayo 2000.
- [4] J.B. Hayet, "Camino más cortos", Centro de Investigación en Matemáticas, Guanajuato, México, 2008.
- [5] S. Janson, D.Knuth, T. Łuczak and B. Pittel. "The birth of the giant component." Random Structures & Algorithms, 4(3):233-358,1993.
- [6] R. Paredes, "Aplicación de Ordenamiento en Línea: Construcción Eficiente del Árbol Cobertor Mínimo", in Proc. Conference on Informatics (CLEI'03), 2003.

M-TAR: Metodología para el Tratamiento de Aspectos en Requerimientos

Xavier S. Medianero P.¹, Sérgio Crespo², Clifton Clunie³

¹ Estudiante, ² Profesor Investigador, Universidad Tecnológica de Panamá

² Profesor Investigador en UNISINOS

^{1,3}{xavier.medianero, clifton.clunie}@utp.ac.pa,

²crespo.unisinos@gmail.com

Resumen- Este artículo propone una metodología cuyo propósito es el tratamiento de aspectos en la etapa de diseño de la Ingeniería de Requerimientos, mediante procesos de identificación, integración y especificación de los componentes de los aspectos, encapsulación de funcionalidades y resolución de conflictos. La metodología propuesta presenta una normativa basada en otros métodos de Ingeniería de Requerimientos Orientadas en Aspectos, incorporando elementos de los enfoques de puntos de vistas, por casos de uso, orientación a objetivos, entre otros. Este trabajo contiene una versión inicial para la metodología basada en cuatro etapas que permite la integración de los aspectos y sus componentes como extracción de identificadores, influencias y dependencias presentes en los modelos rutinarios de la especificación de requerimientos.

Palabras Claves- Ingeniería de Requerimientos; Programación Orientada a Aspectos; Desarrollo de Software.

1. Introducción

En todo proceso de desarrollo de software se tiene como prioridad cumplir con un nivel de calidad sin perjudicar el costo o el tiempo del mismo, incluyendo tareas de análisis, diseño e implementación del mismo. La ingeniería de software es una disciplina que nace como medida para tratar de garantizar un producto de calidad que cumpla con los estándares exigidos mediante un procedimiento estructurado y cíclico de continua mejora de software tanto en diseño, implementación y mantenimiento del mismo.

En el desarrollo de software basado en programación orientada a aspectos (POA), para obtener un software que cumpla con estándares de calidad, es necesario analizar y diseñar el software enfocado a la identificación temprana de aspectos y de esta manera incluirlo en todos los componentes que sean necesarios.

La Ingeniería de Software consta de varias etapas, cada una de ellas presenta un grado significativo de importancia para el proceso, siendo importante destacar el fuerte impacto de la Ingeniería de

Requerimientos en todo el proceso de desarrollo de software en las etapas posteriores y en los resultados finales, por tal motivo este trabajo se enfoca a esta etapa del proceso.

En la Ingeniería de Requerimientos se realizan las interpretaciones de las necesidades y requerimientos del cliente, se crean los documentos con la información detallada del software, se desarrolla el

modelo lógico del sistema entre otras actividades.

La especificación correcta de funcionalidades y la descripción de los componentes del software en la Ingeniería de Requerimientos permitirá crear una base robusta para las etapas posteriores reduciendo la complejidad de las mismas. La incorporación de la Programación Orientada a Aspectos, uno de los actuales paradigmas de programación cuya función es la encapsulación de funciones en elementos denominados aspectos en beneficio de procesos de mantenimiento, depuración y reducción del acoplamiento, en la ingeniería de requerimientos, permitirá incorporar desde etapas tempranas este paradigma y sus beneficios al diseño y productos de la fase.

Según [1], el problema radica en la necesidad de una metodología con normas más acopladas que constituyan un procedimiento para el mejoramiento del diseño y estructura del sistema integrando los aspectos desde una etapa temprana, de tal forma que sea posible obtener un diseño más robusto que sirva de base para las etapas futuras, mejorando de esta forma la calidad a través de un procedimiento sistemático. La metodología que se presenta en este trabajo radica en la necesidad de incorporar a las etapas de análisis y diseño de sistemas de software atributos de la programación orientada a aspectos y de esta forma, abstraer todas las ventajas de la misma para consolidar una base sólida para etapas posteriores de la Ingeniería de Software.

En este artículo se propone una primera versión de una metodología para la consideración de aspectos en la etapa inicial de la Ingeniería de Software, la misma combina métodos y pasos provenientes de otras metodologías como la Ingeniería de Requerimiento Orientada a Aspectos, a Componentes y Aspectos, Modelos de Puntos de Vistas, Ingeniería de orientación a Aspectos utilizando Casos de Uso.

La separación de los asuntos y requerimientos del sistema en aspectos es uno de los aportes más importantes del paradigma de POA [2], debido a esto se han creado disciplinas como lo son la AORE [3], SLAI [1], AOCRE [4] que proponen mecanismos para el análisis de los aspectos en la etapa de Ingeniería de Requerimientos.

El procedimiento que aplica M-TAR consta de cuatro etapas cíclicas para la identificación y especificación; el propósito de la misma es el de establecer una serie de pasos que refuerzan el tratamiento y la abstracción de los elementos de los aspectos dentro del proceso de diseño del sistema de la Ingeniería de Requerimientos en la aplicación de la Ingeniería de Software, mediante la utilización de prácticas rutinarias como casos de Uso, enfoque a objetivos y puntos de vistas.

2. Desarrollo de Software Orientado a Aspectos

El proceso de separación de asuntos es un procedimiento indispensable en la etapa de análisis y diseño de la ingeniería de requerimientos [5]; Esta técnica consiste en la separación de dominios del problema, logrando los siguientes beneficios: Disminuir fallas y evitar o facilitar la resolución de conflictos que podrán presentarse en etapas posteriores causadas por el mal entendimiento de las necesidades y objetivos del proyecto, reducir el grado de complejidad de las actividades de desarrollo [6].

De esta integración nace la disciplina AOSD (Desarrollo de Software Orientado a Aspectos) la cual fue creada para hacer frente a la

dificultad del código transversal debido a su complejidad y su dificultad para entenderse. Las principales ventajas que se logran mediante la integración de AOSD según [7], se tienen: reducción de esfuerzo en el desarrollo del software, un diseño más elaborado y robusto, facilita el mantenimiento y actividades de pruebas y depuración de código, abstrae y separa las funcionalidades encapsulándolas en aspectos, entre otras.

La disciplina AOSD se enfoca en la identificación, separación, representación y la composición de los asuntos de carácter transversal [1]. Estos asuntos son capturados en módulos independientes denominados aspectos.

A continuación se detalla la programación orientada a aspectos y los diversos enfoques de cómo son tratados en la AOSD.

3. Trabajos Relacionados

Trabajos relacionados en el área de la AOSD que intentan suplir la necesidad de una metodología sistemática para el tratamiento de aspectos en la Ingeniería de Requerimientos, entre ellos se tienen:

AORE (Ingeniería de Requerimiento Orientada a Aspectos)

AORE utiliza una metodología en la cual trata los asuntos como un conjunto coherente de requerimientos y donde los aspectos transversales son matrices conformado por los primeros elementos [6], de esta forma es posible determinar cuántas funcionalidades trabajan de manera negativa o positiva frente a otros asuntos.

Esta metodología basada en la manipulación de aspectos cuenta con las siguientes etapas [6, 7]:

Identificación de Asuntos: Separa los requerimientos funcionales y no funcionales del software. En esta etapa se utilizan Vistas, Enfoque a Casos de Uso, Orientación a Objetivos, Orientación a Aspectos, entre otras.

Especificación de Asuntos: Se evalúan la compatibilidad de cada uno de ellos como paso previo para su clasificación por funcionalidad.

Identificación y Especificación de Influencias: Los asuntos presenten en las matrices son alineados y clasificados según su nivel de influencia en otros elementos, creando un enfoque de vistas. En esta etapa se identifican aspectos candidatos, se especifican y priorizan los aspectos resultantes.

Manejo de Conflictos entre asuntos: Conforman las actividades que se llevarán a cabo para la solución de conflictos que surgen por las relaciones entre los asuntos.

Especificación de las dimensiones de los asuntos: Define cómo los aspectos serán soportados en la arquitectura según influencia y mapeo. En esta etapa se evalúa y se predice cómo se adaptarán los aspectos a las siguientes áreas del proceso ingenieril, en especial, frente al ciclo de cambio y redefinición de requerimientos; en este punto es necesario tomar en consideración los aspectos, las funciones y las decisiones para cada etapa del ciclo de vida.

SLAI (Léxico Estructurado para la Identificación de Aspectos)

Esta metodología tiene como objetivo principal la identificación de aspectos potenciales del software en la etapa de diseño de la Ingeniería de Software [1].

Su meta es la de reconocer los posibles puntos transversales de los requerimientos funcionales y no funcionales del software.

Su enfoque está basado en la utilización de casos de uso para el

proceso de integración de aspectos.

La Metodología de SLAI se basa en procesos separados para el manejo de requerimientos y así llevar a cabo la identificación de aspectos.

Para el primer grupo de requerimiento se identifican los actores, usuarios y sistemas que interactúan en el mismo, se crea un repositorio de identificadores determinantes considerados como palabras claves que denotan acciones. Al agregar funcionalidades se verifica si existen en el repositorio, de ser así, la misma puede llegar a considerarse como un aspecto candidato. Este procedimiento es cíclico y busca en optimizar los resultados.

Para el segundo grupo de requerimientos se utilizan funciones ya reconocidas e identificadas, se analizan las necesidades del cliente para extraer cualquier otra y luego la identificación es similar al de los requerimientos funcionales.

AOCRE (Ingeniería de Requerimientos y Componentes Orientada a Aspectos)

Se enfoca a la identificación y especificación de funciones transversales encapsulando los requerimientos en aspectos y a su vez relacionando cada uno de los componentes del sistema [4].

Dentro de sus objetivos está el análisis del comportamiento de los componentes basados en los aspectos que lo cortan, el enfoque de los aspectos dentro del sistema encapsulados por componentes. Se basa en un procedimiento de especialización desde lo más grande a lo más pequeño [4] es decir, un procedimiento de descomposición de los aspectos.

La metodología de AOCRE se basa en la identificación y determinación de aspectos mediante el ciclo de especialización jerárquico de funcionalidades, este proceso se realiza por cada componente, Una vez seleccionado los aspectos candidatos se comparan frente a los requerimientos funcionales y no funcionales para su última aprobación [8].

El Modelo ViewPoint (Puntos de Vista)

Está basado en la metodología de AORE a la cual se le integran los enfoques por puntos de vistas según funcionalidades y actores, para un tratamiento de aspectos más detallado y mejorado [9].

El modelo por puntos de vista se basa en procedimientos de extracción de funciones a partir de los requerimientos del sistema, para luego clasificar las vistas por estas funcionalidades, realizar diagrama de casos y escenarios, como también, la identificación de los requerimientos no funcionales. Posteriormente se realiza una resolución de conflictos para determinar qué función debe tener privilegios frente a otra dependiendo de las influencias, dependencias y estimaciones de los *stakeholders*.

La integración de vistas y aspectos hace que estos enfoques creen una estructura más sólida para el tratamiento de los requerimientos [9].

4. Metodología

MTAR es una propuesta sistemática para la identificación y especificación de aspectos en la Ingeniería de Requerimientos, que consta de cuatro etapas y está basada en metodologías como: AORE [3, 6], AOCRE [4], SLAI [1] y Modelo de Puntos de Vistas [9]. El proceso de identificación de aspectos está basado, por ende, en

Casos de Uso, Puntos de Vistas y enfoque a objetivos.

Cada una de estas etapas forman un procedimiento cíclico, que consta de los siguientes pasos y normativas:

4.1 Definición de Requerimientos

En esta etapa se llevan a cabo los siguientes pasos:

Enfoque a Vistas e Identificación de elementos claves: Involucra la orientación a vistas como primer requisito. Dependiendo de los requerimientos funcionales del sistema y de los actores que se ven involucrados en cada una de las funcionalidades, ya sean: usuarios, sistemas casi o completamente automatizados, se crea un enfoque a vistas por objetivos. De tal forma que de paralelo al procedimiento de creación de vistas se diseña una tabla con los elementos detallados en la Tabla 1.

Tabla 1. Elementos a tomar en consideración de la Orientación a Vistas.

Aspectos	Intervenciones	Acciones	Funciones Afectadas	Condiciones	Función Transversal
Nombre	Inicio Durante Fin	Acción Acción Acción específica	Función Caso de Uso	Parte del Proceso Condiciones	Función Global

Dependiendo a la concurrencia de los actores en el sistema, principalmente usuario se contabiliza en escala de 1-9 su relevancia sobre las funcionalidades del sistema. Una funcionalidad puede tener más de una vista asociada.

Elaboración de Casos de Uso: Con las vistas elaboradas se diseñan los casos de uso. Este procedimiento se realiza de la forma usual.

Contabilización de identificadores determinante de asuntos. Aplicando el método en [11], los identificadores (verbos y sustantivos significativos en el nombre de los casos de uso) se almacenan en un repositorio, de tal forma que sea posible determinar la frecuencia de cada uno de ellos dentro del procedimiento. Además, se incorpora una influencia de los identificadores sobre la cardinalidad de las funcionalidades y vistas, de tal forma que sea posible asociar también cuántos cortes transversales serán posibles realizar a una funcionalidad determinada. Retomando las etapas de definición de requerimientos, a partir de la segunda ejecución de la misma se realizan las siguientes actividades:

Actualización de Casos de Uso. Consiste en adaptar los casos de uso de la primera etapa incorporando el resultado final de las otras etapas del proceso en su primer fase. Este procedimiento es necesario debido a la segmentación de aspectos en sub-aspectos mediante el enfoque de especialización (top-down). El procedimiento cíclico involucra que se realice la sub-etapa de contabilización de identificadores determinante de asuntos de manera similar.

4.2 Identificación de Aspectos

Identificación de Influencias Mediante las influencias determinadas en la identificación de vistas se profundiza la identificación de influencias basadas en los diagramas de caso de uso, de tal forma, obtener qué función influye sobre otra.

A diferencia del procedimiento de AORE, las influencias en este

paso son de manera más específicas en lugar de globales, se determinan por los diagramas.

Aplicación de la matriz de descomposición. Una vez determinadas las influencias, se crea la matriz de descomposición, la cual determinará la cardinalidad de dependencias de un asunto determinado, siendo así posible, determinar los cortes transversales y los posibles aspectos candidatos. El procedimiento para determinar esta matriz será a partir de la regla de descomposición detallada dentro de la revisión bibliográfica del trabajo.

Determinación de Requerimientos No Funcionales Los requerimientos no funcionales se determinarán por su adaptabilidad en relación con el sistema en desarrollo. Este procedimiento se complementa con la identificación de RNF (Requerimientos no funcionales) a partir de licitaciones con los usuarios del sistema.

A continuación un listado de posibles funcionalidades que representan, generalmente, RNF:

- Disponibilidad de Servicios
- Nivel de Seguridad
- Rendimiento del Sistema
- Tiempo de Servicio
- Confiabilidad de Procesos
- Compatibilidad
- Capacidad de Usuarios Múltiples
- Casos Legales
- Adaptabilidad a la Red

En esta etapa se deben identificar los aspectos por nombres, determinar sus funciones, verificar los casos de uso que son impactados por los aspectos y especificar la función transversal (cross-cutting).

Considerando los elementos de los aspectos, podemos ubicar ciertos puntos críticos en la primera etapa (una vez los elementos de la POO se encuentren identificados).

Como apoyo a la regla de descomposición se utiliza el diccionario léxico: mediante la frecuencia de cada identificador se pueden determinar aspectos candidatos, este procedimiento tiene más relevancia para sistemas complejos.

A partir de la segunda intervención, la Determinación de RNF y la matriz de descomposición no se aplican debido a que su relevancia es mínima.

4.3 Especificación de Aspectos

Fragmentación de Aspectos La fragmentación de aspectos por el enfoque de especialización (top-down) se lleva a cabo después de la identificación de los cortes transversales, consiste en que, una vez identificados los aspectos en la etapa ii, se crea un listado de cada uno de ellos y de su funcionalidad asociada, se intenta descomponer las funciones en funciones más pequeñas y se trata de encontrar un aspecto asociado; de ser posible se segmenta el aspecto. En la Tabla 2 se muestran elementos que deben especificarse de los aspectos para facilitar el procedimiento de fragmentación.

Tabla 2. Modelo de especificación de aspectos.

Aspectos	Intervenciones	Acciones	Funciones Afectadas	Condiciones	Función Transversal
Nombre	Inicio Durante Fin	Acción Acción Acción específica	Función Caso de Uso	Parte del Proceso Condiciones	Función Global

Como otro método de segmentación de aspectos está su división (representación y especificación de *advices*, puntos de corte, puntos de unión) que se hace a partir de los siguientes pasos: Especificación de componentes de los aspectos: Como se muestra en la Figura [1], la especificación de los *advices*, puntos de corte y puntos de unión se lleva a cabo a través del siguiente procedimiento:

Cross-cutting: se visualizan en la relación directa entre los aspectos y los casos de uso que interceptan, es decir, dependiendo de la funcionalidad del caso de uso se conceptualizan dentro de una función transversal; representa la funcionalidad transversal en cada uno de los casos de uso al elemento que la genera.

Join-Point: Se asocian con un aspecto determinado, contiene los métodos que forman parte de la funcionalidad transversal. Se propone su identificación mediante la detección del diagrama de caso de uso involucrado al aspecto determinado, su ubicación debe estar entre el caso de uso afectado por el crosscutting y el anterior.

Cut-Point: Son el conjunto de los join-point, pero colocados en todos los casos de uso que intervienen en el proceso, es decir, la ubicación de los puntos de corte se dan en todos los casos de uso que ejecutarán en su siguiente etapa el corte transversal. También es necesario analizar los elementos anteriores al elemento de caso de uso analizado.

Advice: deben ubicarse en el caso de uso al que el aspecto interviene, los *advices* serán colocados de forma intermediaria entre los predecesores del caso de uso y el mismo, para los eventos *before()*, *in()* y *after()* en caso de ser necesarios, si no es así, simplemente se prescinden de ellos.

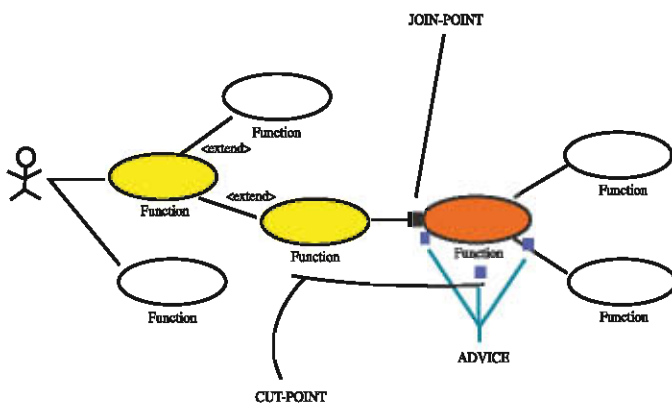


Figura 1. Ubicación de los puntos de corte dentro de un diagrama de casos de uso.

Clasificación de Aspectos: La clasificación de aspectos se realiza para filtrar los aspectos reales de posibles funciones y decisiones. A este proceso se le denomina "Mapeo".

En el *mapeo*, a cada aspecto ya definido, se le asocia las funcionalidades sobre las que influye y se separan de las funciones y decisiones del proceso.

Control de Redundancia: Una vez segmentado los aspectos, es necesario revisar si estos no causan redundancia con otros aspectos identificados, de ser así, se eliminan. Esta redundancia se puede visualizar con el listado de funcionalidades de cada aspecto. A partir de la segunda intervención, los procedimientos de segmentación de aspectos y especificación no son contemplados, ya que, la relevancia es mínima.

4.4 Resolución de Conflictos

En la resolución de conflictos se utiliza la "tabla de conflictos para *stakeholders*" aplicada en la metodología AORE y ViewPoint, modificada por el enfoque a actores del sistema.

La Tabla [3] es la tabla de conflictos para *stakeholders* con valoración de actores involucrados en el sistema, pero más significativo para esta evaluación son las influencias y dependencias de las funcionalidades en conflicto. Un porcentaje del máximo de influencias actuales se aplicará a cada funcionalidad en conflicto, de tal forma, que quien tenga más funciones dependientes merecerá más prioridad; en este punto es importante resaltar, que la magnitud de significancia es restada si una función en conflicto es predecesora de la misma, debido a que la otra función se ejecutará primero.

Tabla 3. Para cualquier intervención de esta etapa se aplica la misma metodología.

Funciones	Stakeholders	Actores	Número de	¿Antecede?
1	2	Valoración	Dependencias	

5. Conclusión

M-TAR es una metodología propuesta que está incorporando modelos para el tratamiento de aspectos en la Ingeniería de Requerimientos basada en AORE, SLAI, AOCRE y el Modelo ViewPoint, que son procedimientos donde se incluyen aspectos en el análisis de requisitos no funcionales y funcionales del sistema para el diseño del software.

Su enfoque está basado en el análisis de diagramas elaborados con casos de uso, componentes, orientación a objetivos y vistas; técnicas de descomposición de aspectos, abstracción de funcionalidades, entre otras.

La incorporación del paradigma de programación orientada a aspectos dentro de las etapas de diseño de la Ingeniería de Requerimientos permite agregar al modelo una estructura más sólida debido a que se podrá contar con todos los beneficios de la misma desde fases tempranas.

Las normas establecidas y la metodología conformada por la unión de características de otros modelos ha resultado un método alternativo para el tratamiento de aspectos.

Consta de procesos como Identificación de aspectos en casos de uso, segmentación de vistas, resolución de conflictos, método determinación de puntos de corte, puntos de unión, advice, entre otros elementos.

Referencias

[1] C. C. Budwell and F. J. Mitropoulos, "The SLAI Methodology: An Aspect Oriented Requirement Identification Process," in 2008 International Conference on Computer Science and Software Engineering, Wuhan, Hubei, 2008, pp. 296 - 301

- [2] H. Hu, et al., "An AOP Framework and Its Implementation Based on Conceptual Model," ISECS International Colloquium on Computing, Communication, Control and Management, p. 4, 2009 2009.
- [3] M. Aoyama and A. Yoshino, "AORE (Aspect-Oriented Requirements Engineering) Methodology for Automotive Software Product Lines," presented at the Software Engineering Conference, 2008. APSEC '08. 15th Asia-Pacific, Beijing, 2008.
- [4] J. Grundy, "Aspect-oriented requirements engineering for component-based software systems," in IEEE International Symposium on Requirements Engineering, 1999., Limerick, 1999, pp. 84 - 91.
- [5] D. Dahiya and R. J. Sachdeva, "Understanding Requirements: Aspect Oriented Software Development," in Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06), Chicago, IL, 2006, pp. 303 - 308.
- [6] M. Tabares, et al., "Aspect Oriented Software Engineering: An Experience of Application in Help Desk Systems," Dyna rev.fac.nac.minas, vol. 147, 2007.
- [7] A. Rashid, et al., "Early aspects: a model for aspect-oriented requirements engineering," presented at the Proceedings. IEEE Joint International Conference on Requirements Engineering, 2002, 2002.
- [8] A. M. Reina and J. Torres. Components + Aspects: A General Overview. Revista Colombiana de Computación.
- [9] P. Yu-Ning and L. Qiang, "A Viewpoint-Oriented Requirements Elicitation Integrated with Aspects," presented at the World Congress on Computer Science and Information Engineering, 2009 WRI., Los Angeles, CA 2009



10^a Conferencia Internacional del Consorcio Latinoamericano y del Caribe de Escuelas de Ingeniería



Megaproyectos: Construyendo Infraestructura mediante Colaboración en Ingeniería, Integración Eficiente y Efectiva y Planificación Innovadora

Educación, Innovación, Tecnología, Diseño y Práctica

Julio 23-27, 2012

Ciudad de Panamá, Panamá

Organizado por:
LACCEI Consorcio Latinoamericano y del Caribe de Escuelas
de Ingeniería y la Universidad Tecnológica de Panamá

www.laccei2012panama.org.pa

Biografías

Lic. José Beltrán, Dr.-Ing. Carlos A. Medina C.

Universidad Tecnológica de Panamá

En esta edición de Biografías, presentamos la vida y obra de dos mentes grandiosas, cuyas ideas estuvieron muy por delante de su tiempo, anticipándose a la era de las computadoras digitales y la informática.

Charles Babbage

Nació en Totnes, Inglaterra, en 1791, hijo de un rico banquero. Ya de niño se interesó por los artefactos mecánicos y aprendió matemáticas de forma autodidacta.

En 1810 entró a la Universidad de Cambridge, donde en ese tiempo, la enseñanza de las matemáticas consistía principalmente en el estudio de los trabajos del científico inglés Isaac Newton, relegando las investigaciones más recientes de Euler, Lacroix y otros matemáticos europeos. Con el propósito de reformar la enseñanza de las matemáticas en Inglaterra, Babbage y sus contemporáneos, entre los que se contaba su amigo de toda la vida, el astrónomo John Herschel, fundaron la Sociedad Analítica. En 1816 tradujeron el tratado de Lacroix sobre el cálculo, una importante forma de resolver muchos problemas matemáticos, seguido en 1820 por un trabajo conjunto: Ejemplos del Cálculo Diferencial e Integral. Fue durante este tiempo, trabajando en estos proyectos, que Babbage consideró la posibilidad de usar máquinas para realizar algunos cálculos.

Por ejemplo, se dio cuenta de que en las tablas astronómicas de la época había gran cantidad de errores, y que éstos se debían a fallas de cálculo por dejadez o aburrimiento de las personas que los desarrollaban. Babbage decidió que la única solución para arreglar los errores en las tablas era construyendo una máquina que hiciera los cálculos de forma automática – una calculadora. Sería una máquina infalible, infatigable y multiusos. Así que empezó a experimentar y, logrando la ayuda del gobierno inglés, inició la construcción de lo que llamó una máquina diferencial (Difference Engine) para calcular las tablas de navegación y otras tablas más. Esta máquina debía ser capaz de calcular cifras de hasta 20 dígitos, almacenar una serie de números y efectuar sumas con estos, presentando los resultados en forma impresa para eliminar el error humano. Esta máquina funcionaba con ruedas dentadas y en sistema decimal (no binario, como las calculadoras y computadoras actuales), pero su complejidad superaba la capacidad técnica de la época. Había muchos problemas con los rozamientos de los engranajes y las vibraciones. Aún cuando Babbage tenía una tremenda capacidad de improvisación y resolvía los problemas con ideas brillantes, a medida que la máquina crecía, la misma no pudo finalizarse.

A pesar de ello, en 1827 había utilizado la única parte operativa de su máquina, formada por apenas 2 000 piezas, para calcular



tablas de logaritmos de 1 a 108 000. Esa parte de la Máquina Diferencial se considera hoy día como la primera calculadora automática.

Después de varios años de trabajo, y a pesar de perder el apoyo del gobierno, desarrolló un concepto aún mejor, la “Máquina Analítica”, una máquina calculadora mucho más compleja y que, aunque mecánica, incorporaba muchas de las características de las computadoras actuales. La máquina analítica tenía dispositivos de entrada basados en tarjetas perforadas, un procesador aritmético que calculaba números, una unidad de control que determinaba qué tarea debía ser realizada, un mecanismo de salida y una memoria donde los números podían ser almacenados hasta ser procesados. Lo interesante de la Máquina Analítica es que sería “programable”. Esto, ¡hace más de 180 años!

Hoy se considera que la máquina analítica de Babbage fue la primera computadora del mundo. Un diseño inicial plenamente funcional de ella fue terminado en 1835. Sin embargo, debido a problemas similares a los de la máquina diferencial, la máquina analítica nunca fue terminada.

Babbage fue una figura enérgica y controvertida. Muchos de sus contemporáneos le consideraban un demente. Pero, no se puede pensar así de una persona que, además de idear la primera computadora de la historia, inventó el velocímetro, el dinamómetro, el oftalmoscopio, los códigos para navegación usados con los faros, y fue el precursor de la Dedicronología, al descubrir que la anchura de los anillos de los árboles dependía del clima producido aquel año y que por tanto, podían utilizarse como registro meteorológico. Además, se interesó por la política, la economía, la filosofía y la astronomía. También se dedicó a la criptografía y algunos de sus trabajos fueron considerados, durante un tiempo, como secreto militar. También enunció el Principio de Babbage, en el que se afirma que el trabajo es mejor producido en tareas simples para los operarios y no que un mismo operario haga todas las tareas.

En 1816 fue elegido miembro de la Royal Society y en 1824 le concedieron la Medalla de Oro de la Royal Astronomical Society por su “Invención de una máquina de cálculo matemático y tablas astronómicas”.

El año 1827 fue fatídico para él, pues murieron su padre, su esposa y algunos de sus ocho hijos, de los cuales sólo sobrevivieron tres hasta edad adulta.

Entre 1828 y 1839 fue elegido para ocupar la cátedra Lucasiana de matemáticas de Cambridge, que en su día tuvieron personajes como Newton y Dirac. En esa época, contribuyó a numerosas revistas astronómicas y tuvo un papel fundamental en la fundación de la Sociedad Astronómica y la Sociedad de Estadística.

En 1833, completó su “máquina diferencial” y formuló los fundamentos teóricos de cualquier autómatas de cálculo.

Durante los últimos años de su vida, se dedicó a la construcción de máquinas capaces de la ejecución de operaciones aritméticas y cálculos algebraicos. Así, Charles Babbage sentó los principios básicos de las computadoras modernas.

Babbage sufrió un duro golpe con la muerte de su esposa y nunca volvió a casarse. Esto, unido a que los tres hijos que le sobrevivieron se habían marchado de Inglaterra, hizo que muriera en total soledad el 18 de octubre de 1871.

Aún cuando su obra permaneció olvidada durante largo tiempo,

hoy se reconocen ampliamente sus trabajos y sus grandes aportes, quedando inscrito en la historia como el padre de la Informática.

Augusta Ada Byron, Condesa de Lovelace

Augusta Ada Byron era hija del famoso poeta Lord Byron y de Anabella Milbanke Byron. Nació el 10 de diciembre de 1815 en Piccadilly Terrace, Londres. Sus padres se separaron al poco tiempo de haber nacido la niña y la custodia la obtuvo la madre. Para evitar que su hija imitara el camino de su padre, Lady Byron se encargó de inclinar los intereses de su hija hacia la ciencia y las matemáticas.



La niña, que tenía una capacidad intelectual privilegiada, respondió al estímulo y de hecho fue una autodidacta en geometría, la cual llegó a dominar con gran soltura, al igual que la astronomía y la matemática. Fue ayudada en sus estudios avanzados por Augustus De Morgan, el primer profesor de matemáticas de la Universidad de Londres.

No obstante, ya fuera por su naturaleza o por un deseo de acercarse a su padre, Ada no pudo negarse a la seducción de la poesía y la música. En los números y fórmulas que la rodeaban en su vida de investigación y estudio, Ada siempre encontró el lado poético, y tal vez por esta razón es que sus concepciones e ideas eran diferentes y se enriquecían con una gran imaginación.

A sus diecisiete años, la niña de marcada inteligencia y de una ilimitada fantasía entró en contacto con Mary Somerville, la traductora de los trabajos del astrónomo Simone Laplace. Ella alentó y apoyó a Ada en sus estudios científicos.

Fue en una cena en casa de Somerville, el año de 1843, que la joven Ada escuchó por primera vez a Charles Babbage exponer sus ideas acerca de la posibilidad de una máquina calculadora llamada el motor analítico. Gracias a su sensibilidad y su inteligencia perceptiva, Ada se sintió profundamente atraída por esta idea, que no despertó gran revuelo en ese momento.

Años antes, en 1835, Ada había contraído matrimonio con William King, conde de Lovelace, y en 1838 se le otorgó el título de condesa (Lady) de Lovelace. Tuvo tres hijos. Pero nada de esto fue un obstáculo para la joven, cuya pasión por la ciencia y la tecnología no tenía límites.

Babbage realizó una conferencia sobre sus trabajos en un seminario de Turín (Italia) en 1840, y un matemático e ingeniero italiano llamado Luigi Federico Menabrea escribió un artículo en francés sobre todo lo dicho por Babbage.

Ada se avocó a traducir, en 1843, el artículo escrito por Menabrea: "Nociones sobre la máquina analítica de Charles Babbage", elaborando y detallando anotaciones sobre el mencionado artículo; en especial, fue excelente su descripción de cómo el propuesto "motor analítico" podría ser programado para computar los números de Bernoulli.

A partir de ahí, se inicia una larga relación de colegas entre la joven y el inventor. El motor que Babbage había visualizado y que pudo concretar gracias a los aportes de Ada, es el antecesor de nuestras computadoras.

Ada sugirió a Babbage escribir un "plan" para que la máquina calculase números de Bernoulli. Este "plan" es considerado el primer "programa de ordenador", y por ello se considera a Ada la primera programadora de la historia.

Sus anotaciones hicieron un profundo análisis del invento y permitieron que el mismo fuera desarrollado. El motor analítico calculaba cualquier función algebraica y almacenaba números. El programa que permitía su funcionamiento no estaba guardado en la máquina como hoy en día, sino que era introducido en la máquina por medio de tarjetas.

En su primera publicación, la traducción del trabajo de Babbage, la condesa de Lovelace hace la predicción de que una máquina como ésta podría llegar a componer música, producir gráficos, y que podría llegar a ser utilizada tanto en el ámbito científico como en la vida diaria. Hablaba de las computadoras de hoy.

Con amigos de muy alta envergadura como Sir David Brewster, el inventor del caleidoscopio, y el escritor Charles Dickens, Ada alimentaba su ansia de conocimiento. Sus intereses no se enfocaban únicamente en la máquina o en la ciencia, sino que abarcaba la música, los caballos y la astronomía, entre otras disciplinas.

Pero su mayor logro no fue ayudar a Babbage. Ada diseñó por sí misma un programa para la máquina de Babbage que calculaba los números de Bernoulli. Éste es el primer programa de computadora que se reconoce, es una especie de software demasiado avanzado para la época en que fue concebido.

Ada murió muy joven al igual que su padre, a la edad de 36 años, de un cáncer de matriz, en 1852. Su trabajo se mantuvo oculto por muchos años, pero cuando la computación se convirtió en una disciplina en desarrollo, fue sacado a la luz, reconocido y altamente respetado.

Existe un lenguaje de programación desarrollado por el Departamento de Defensa de EE.UU. en 1979 que lleva su nombre: ADA, para recordar su trabajo y darle el crédito que se merece.

Sus ideas fueron retomadas por Alan M. Turing y John Von Neumann, dos de los padres de las modernas tecnologías de la computación.

Sin duda, es a Lady Lovelace y a su fecunda capacidad imaginativa que le debemos nuestra era informática.

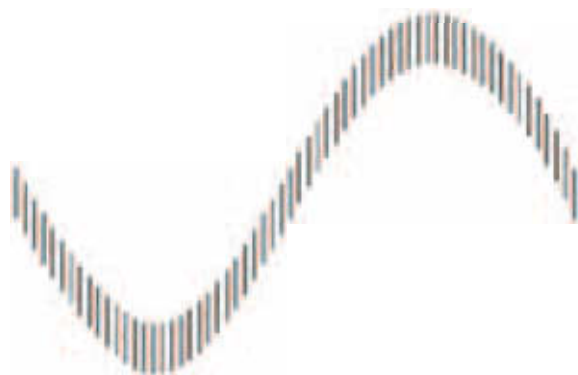
Referencias

- [1] Charles Babbage, Historia de la Ciencia, <http://www.historiasdelaciencia.com/?p=314>, Citado 4 de mayo de 2011
- [2] Charles Babbage, Biografías y Vidas, <http://www.biografiasyvidas.com/>, Citado 4 de mayo de 2011
- [3] The New How It Works Encyclopedia, Babbage, Charles, Vol.24, pp. 3214, Publ. H.S. Stuttman Inc., U.S.A. 1989
- [4] Ada Byron, http://www.dma.eui.upm.es/historia_informatica/Doc/personajes.htm, Citado 4 de mayo de 2011
- [5] Ada Byron, <http://centros5.pntic.mec.es/ies.juan.de.mairena/adabyron.html>, Citado 4 de mayo de 2011

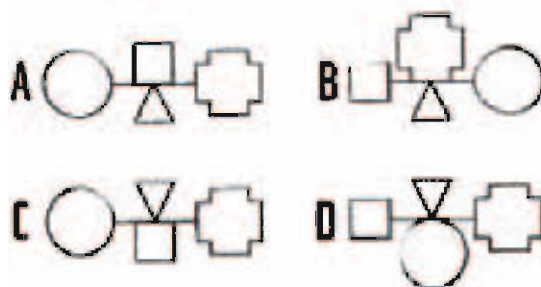
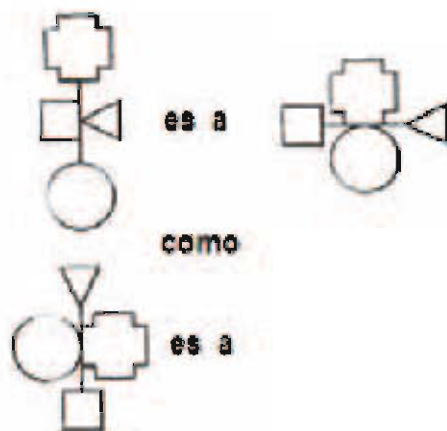
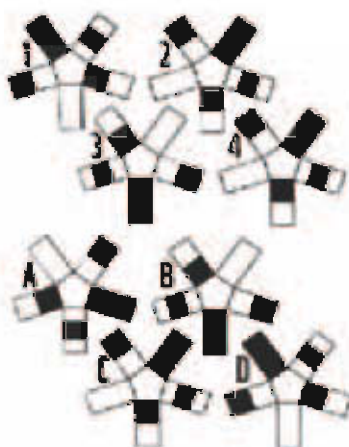
¿Puede encontrar al hombre oculto en esta escena?



¿Puede decir dónde en esta curva las líneas son más largas, y dónde son más cortas?



¿Cuál de las figuras de abajo – A, B, C o D – siguen a la número 4?



Multiplicando y sumando

Intente llenar los números que faltan. Use números del 1 al 25 para completar las ecuaciones. Cada número se usa una sola vez. Cada fila y cada columna es una ecuación. Recuerde que la multiplicación y la división se realizan primero que la adición y la sustracción.

23	+		-		x	+		-184
-		x		x		-		+
	x		+		-		/	463
-		+		-		x		-
	-		-	11	+		x	131
x		+		+		-		-
	-		-		x	-		-104
+		+		-		+		/
	-		-		x	+	6	-88
-49		542		78		-186		10



**UNIVERSIDAD TECNOLÓGICA
DE PANAMÁ**

OFERTA ACADÉMICA

La Universidad Tecnológica de Panamá, a la vanguardia con el incesante crecimiento de nuestro país; se ha destacado, por ser una de las universidades

Actualmente la UTP posee una oferta académica de gran relevancia y variedad al ofrecer, en Panamá, novedosas carreras en Programas de Maestrías y Postgrados dirigidos hacia las áreas Administrativas y Financieras de la Industria, Ingeniería, Ciencias y Tecnología.

www.utp.ac.pa

Tels.: (507) 560-3000

Campus Universitario
Víctor Levi Sasso
Ciudad de Panamá

@utppanama



Programas de Postgrados y Maestrías

FACULTAD DE INGENIERÍA CIVIL

Maestría y Postgrado en Ingeniería Estructural
Maestría y Postgrado en Ingeniería Ambiental
Maestría y Postgrado en Administración de
Proyectos de Construcción
Maestría y Postgrado en Desarrollo Urbano y
Regional
Postgrado en Sistemas de Información Geográfica

Para mayor información:

Ing. Nelson Cedeño / Tel.: 560-3006
Ing. Viccelda Domínguez
nelson.cedeno@utp.ac.pa
viccelda.dominguez@utp.ac.pa
Visítenos en: <http://www.fic.utp.ac.pa>

FACULTAD DE INGENIERÍA INDUSTRIAL

Postgrado en Alta Gerencia
Postgrado en Logística
Postgrado en Formulación, Evaluación y Gestión de
Proyectos
Maestría en Dirección de Negocios con especializa-
ción en:

- Estrategia Gerencial
- Mercadeo Estratégico
- Gerencia en Recursos Humanos
- Administración de Sistemas de Información
- Economía de las Empresas

Maestría en Sistemas Logísticos y Operaciones con
especialización en:

- Planificación de la Demanda
- Centros de Distribución

Maestría en Gestión de Proyectos con especializa-
ción en:

- Administración
- Evaluación

Para mayor información:

Coordinación de Postgrado y Maestría
Tels.: 560-3145 / 560-3386
juan.moran@utp.ac.pa
luis.vega@utp.ac.pa
marta.caballero@utp.ac.pa
Visítenos en: <http://www.fii.utp.ac.pa>

FACULTAD DE INGENIERÍA ELÉCTRICA

Postgrado en Ingeniería Electrónica Digital
Postgrado en Telecomunicaciones
Postgrado en Ingeniería Eléctrica Industrial
Maestría en Ingeniería Eléctrica con especialización
en:

- Potencia Eléctrica
- Electrónica Digital y Automatización
- Telecomunicaciones

Para mayor información:

Coordinación de Postgrado y Maestría
Tel.: 560-3050
carlos.medina@utp.ac.pa Vicedecano de Inv. y
Extensión
fernando.merchan@utp.ac.pa Coordinador
Visítenos en: <http://www.fie.utp.ac.pa>

FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES.

Maestría de Gestión de Servicios de Información
Documental
Maestría y Postgrado en Redes de Comunicación de
Datos
Maestría y Postgrado en Informática
Maestría y Postgrado en Ingeniería del Software
Aplicada
Maestría y Postgrado en Auditoría de Sistemas y
Evaluación de Control Informático
Maestría y Postgrado en Informática Educativa
Maestría en Ciencias de la Tecnología de la Informa-
ción y Comunicación (Diurno)

Para mayor información:

Vicedecano de Investigación, Postgrado y Extensión
Tels.: 560-3606 / 560-3657
julio.lezcano@utp.ac.pa
victor.fuentes@utp.ac.pa
Visítenos en: <http://www.fisc.utp.ac.pa>

FACULTAD DE INGENIERÍA MECÁNICA

Maestría en Ingeniería de Plantas y Mantenimiento de
Plantas
Maestría en Energía Renovable y Ambiente
Maestría en Ciencias de la Ingeniería Mecánica con
especialización en:

- Manufactura y Materiales, o
- Automatización y Robótica

Para mayor información:

Coordinación de Postgrado y Maestría
Dr. Dimas Portillo
Tel.: 560-3105
Correo electrónico: postgrado.fim@utp.ac.pa
Visítenos en: <http://www.fim.utp.ac.pa>

FACULTAD DE CIENCIAS Y TECNOLOGÍA

Profesorado en Educación Media y Premedia en
Ciencias y Tecnología con especialización en el área
Maestría en Docencia Superior

Para mayor información:

Coordinación de Postgrado y Maestría
Tels.: 560-3637 / 560-3677
Ildeman.abrego@utp.ac.pa
celmibeth.herrera@utp.ac.pa
Visítenos en: <http://www.fct.utp.ac.pa>

COORDINACIÓN DE DOCTORADO

Coordinador Administrativo Ing. Gabriel Vergara
gabriel.vergara@utp.ac.pa Tel: 560-3351
Coordinador Académico Dr. Ramfis Miguelena
ramfis.miguelena@utp.ac.pa

“Camino a la excelencia a través del mejoramiento continuo”.



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ



*“Camino a la excelencia
a través del mejoramiento continuo”.*



Correspondencia y comentarios: prisma@utp.ac.pa