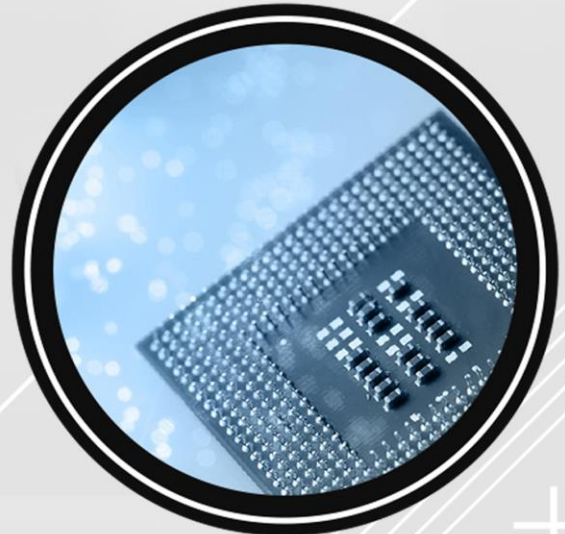


  
**RECIBE**  
Revista electrónica  
DE COMPUTACION, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA



# Índice

## Computación e Informática

Optimización del desempeño de un sistema de recomendación de documentos de texto basado en la configuración de los servidores.....I

Rodrigo Vences Nava

Víctor Hugo Menéndez Domínguez

Alfredo Zapata González

Building an ontological modeling mechanism for virtual cities .....II

Luis Casillas

Gilberto Vargas

Miguel Lares

## Electrónica

Reconocimiento de micro partículas de polen con algoritmos de procesamiento de imágenes

implementados en dispositivos reconfigurables .....III

Juan Manuel Ruiz Varela

Susana Ortega Cisneros

Adrian Pedroza de la Cruz

Juan José Raygoza Panduro

Jorge Rivera Domínguez



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

# Optimización del desempeño de un sistema de recomendación de documentos de texto basado en la configuración de los servidores.

Rodrigo Vences Nava  
Universidad Autónoma de Yucatán  
[vnava@correo.uady.mx](mailto:vnava@correo.uady.mx)

Víctor Hugo Menéndez Domínguez  
Universidad Autónoma de Yucatán  
[mdoming@correo.uady.mx](mailto:mdoming@correo.uady.mx)

Alfredo Zapata González  
Universidad Autónoma de Yucatán  
[zgonzal@correo.uady.mx](mailto:zgonzal@correo.uady.mx)

**Resumen:** Cuando se habla de sistemas informáticos en la Web, una de las configuraciones más utilizadas por los desarrolladores es la combinación de Apache y MySql, independientemente del sistema operativo sobre el cuál se desarrolle. En este trabajo se muestra cómo optimizar los tiempos de procesamiento y despliegue de información en el navegador Web de un Sistema de Recomendación de trabajos de titulación, aplicando sencillas configuraciones en el servidor Apache y siguiendo algunas buenas prácticas de programación al momento de ejecutar llamadas a la

base de datos. Las configuraciones realizadas son específicamente en cuanto a la compresión de datos y el manejo de caché. Este ahorro de tiempo repercute directamente en una mejor percepción por parte del usuario con respecto a la usabilidad del sistema.

**Palabras clave:** optimización, servidor Web, Apache, MySQL.

## **Improvement performance of a recommendation system for text documents based on servers configuration**

**Abstract:** When we speak about computer systems on the Web, one of the configurations more used by developers is the combination of Apache and MySQL, regardless of operating system on which it develops. This paper shows how to optimize processing times and display of information in the Web browser for a recommendation system of graduation thesis using simple settings in the Apache server and following some good programming practices when executing calls to the database. The settings made are specifically data compression and cache management. This time saving directly affects in a better perception by the user regarding the usability of the system.

**Keywords:** Keywords: optimization, Web server, Apache, MySQL

# 1. Introducción

Los Sistemas de Recomendación (SR) recopilan información sobre las preferencias de los usuarios sobre un conjunto de elementos o ítems (películas, canciones, libros, etc.), desde diferentes fuentes de información para proporcionar a los usuarios predicciones y recomendaciones de elementos similares, tratando de equilibrar factores como exactitud, novedad, dispersión y estabilidad en las recomendaciones (Bobadilla, Ortega, Hernando, & Gutiérrez, 2013).

Dentro de los esfuerzos que se han realizado para gestionar la información textual se encuentran principalmente los del área de recuperación de información (RI), la cual trata con la representación, almacenamiento, organización y acceso de

elementos de información. Los documentos recuperados deben satisfacer las necesidades de información del usuario expresadas en lenguaje natural (Baeza-Yates & Ribeiro-Neto, 1999).

La RI significa encontrar material (usualmente documentos) de una naturaleza no estructurada (usualmente texto) que satisface una necesidad de información desde dentro de grandes colecciones (generalmente se encuentran almacenados en computadoras) (Manning & Schutze, 1999).

Existen diversas herramientas para poder manejar el gran volumen de información, por ejemplo: clasificadores temáticos de textos, sistemas de detección de plagio, detección de reputación en redes sociales. Sin embargo todos tienen en común conocer el grado de similitud textual entre un par de textos (Álvarez Carmona, 2014).

Surge entonces la necesidad de procesar automáticamente ese gran volumen de información y para ello se ha recurrido a una de las ramas de la Inteligencia Artificial: el Procesamiento de Lenguaje Natural (PLN) (Manning, Raghavan, & Schütze, 2009).

El enfoque principal de esta área consiste en crear métodos, técnicas y herramientas computacionales que permitan realizar análisis de información escrita u oral y que faciliten la búsqueda y organización de dicha información (Karin Verspoor, 2013).

Algunos de los problemas que se presentan al desarrollar soluciones para manejar grandes cantidades de información textual, son los tiempos tanto de espera en el proceso de indexación, como de respuesta de la aplicación y la adecuada configuración de los servidores donde reside la aplicación, o las características de conectividad existentes dentro de una institución (Souders, 2007), los cuales impactan en el rendimiento del sistema y la satisfacción del usuario final.

Una posible solución para mejorar el desempeño de este tipo particular de sistemas como son los sistemas de recuperación de información y los sistemas de recomendación consiste en realizar modificaciones particulares a su código fuente, según recomendaciones o directrices de uso común (Souders, 2007). Sin embargo, esto origina problemas de mantenimiento, en especial cuando se trata

de aplicaciones de libre distribución, donde las actualizaciones resultan muy importantes. Por otro lado, adecuar la conectividad no siempre es posible dentro de una institución, dadas las políticas y normativas preestablecidas.

En este trabajo se propone una solución a esta problemática utilizando técnicas (manejo de caché y compresión de datos) que se centran en la configuración de los servidores donde se ejecuta nuestro caso de prueba, que es el Sistema de Recomendación de Trabajos de Titulación de la Facultad de Ciencias Antropológicas de la Universidad Autónoma de Yucatán (Vences, Menéndez, & Zapata, 2015) y que al aplicarlas en conjunto optimizan el desempeño del mismo en poco más del 50% en cuanto a velocidad de transferencia, considerando un esquema sólo de compresión de datos y hasta un 90.8% en cuanto a volumen de transferencia considerando un esquema de compresión de datos y manejo de caché. Además de optimizar hasta en un 99.98% los accesos a la base de datos, siguiendo las recomendaciones para la ejecución de sentencias SQL. Nuestra propuesta se ha implementado en el servidor Web Apache (<https://httpd.apache.org/>) y el servidor de base de datos MySQL (<https://www.mysql.com/>).

## 2. Aspectos Relacionados

La accesibilidad de una aplicación se refiere a su utilización, en forma satisfactoria, por un mayor número de personas, sin importar sus limitaciones intrínsecas o derivadas del entorno en que se encuentran (W3C, 2010). Los tiempos de respuesta de la aplicación por lo tanto impactan en gran medida en la accesibilidad.

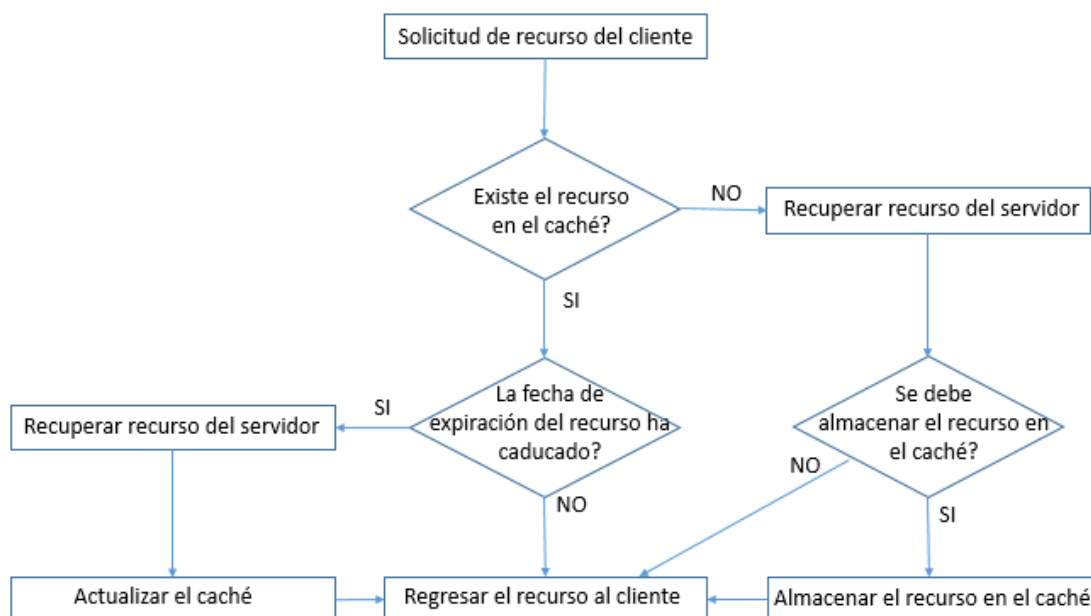
Algunas de las razones por las que una aplicación Web es lenta, por ejemplo, es que no fueron diseñadas pensando en la velocidad de carga sino en la apariencia (diseñador), o se tienen accesos a bases de datos sin considerar un tráfico pesado (Smith, 2013).

## 2.1 Optimización del servidor web Apache

La elección de la configuración correcta de los parámetros para optimizar el servidor web Apache no es un trabajo sencillo. Las mejores configuraciones dependerán del hardware, las cargas de trabajo y cualquier proceso ejecutándose simultáneamente en el sistema. (Gandhi, Tilbury, Diao, Hellerstein, & Parekh, 2002).

Algunos autores, (Gandhi et al., 2002; Jun Li & Menghan Lu, 2013) implementan una optimización de desempeño dinámica en el servidor web Apache, centrando su atención en limitar el uso del CPU y la memoria, justificando que, cualquier cambio en el archivo de configuración de Apache con el fin de optimizar el desempeño, sólo funciona cuando el servidor es reiniciado. Este trabajo por el contrario se centra en aplicar ciertas configuraciones al archivo de configuración de Apache debido a que se tiene el control total del sistema.

En el desarrollo de aplicaciones Web existen directrices (Smith, 2013) para la optimización del desempeño que impactan fuertemente en la accesibilidad de la aplicación si son tomadas en cuenta, en especial el manejo de compresión y de caché tanto del cliente como del servidor y la replicación y buenas prácticas del uso de bases de datos. La figura 1 muestra el manejo del caché.



### ***Figura 1. Manejo del caché.***

A cada petición del usuario, el navegador determina si el recurso requerido ya ha sido solicitado recientemente para recuperarlo del caché y utilizarlo siempre y cuando no haya vencido su período de caducidad. Esto elimina el tiempo de espera por recibir el recurso del servidor y hace más rápida su presentación. Un proceso similar se puede hacer del lado del servidor para el manejo del caché.

De igual forma tanto en el cliente como en el servidor se puede emplear un esquema de compresión de recursos antes de almacenarlos en el caché y de esta manera se optimizan los tiempos de procesamiento y se agiliza la descarga de los recursos solicitados.

El esquema de compresión varía dependiendo del tipo de recurso, ya que si son imágenes o documentos PDF, la compresión está dada por el mismo formato del archivo; En cambio, si son hojas de estilo, scripts o páginas estáticas es posible comprimirlas e incluso minificar el código (Smith, 2013).

Diversos estudios han demostrado la factibilidad de aplicar éste y otro tipo de configuraciones a nivel de servidores, obteniendo reducciones significativas en volumen de transferencia y despliegue de la información, llegando a ser hasta del 40% (Ahuja, Wu, & Dixit, 2003; Menéndez, Castellanos, Aguilar, & Gómez, 2013).

## **2.2 Optimización de la base de datos MySQL**

En Ryser y Glinz (1999) se define un procedimiento para la recopilación de los requisitos y su documentación a través de escenarios, en el cual destacan 15 pasos para la creación de un escenario:

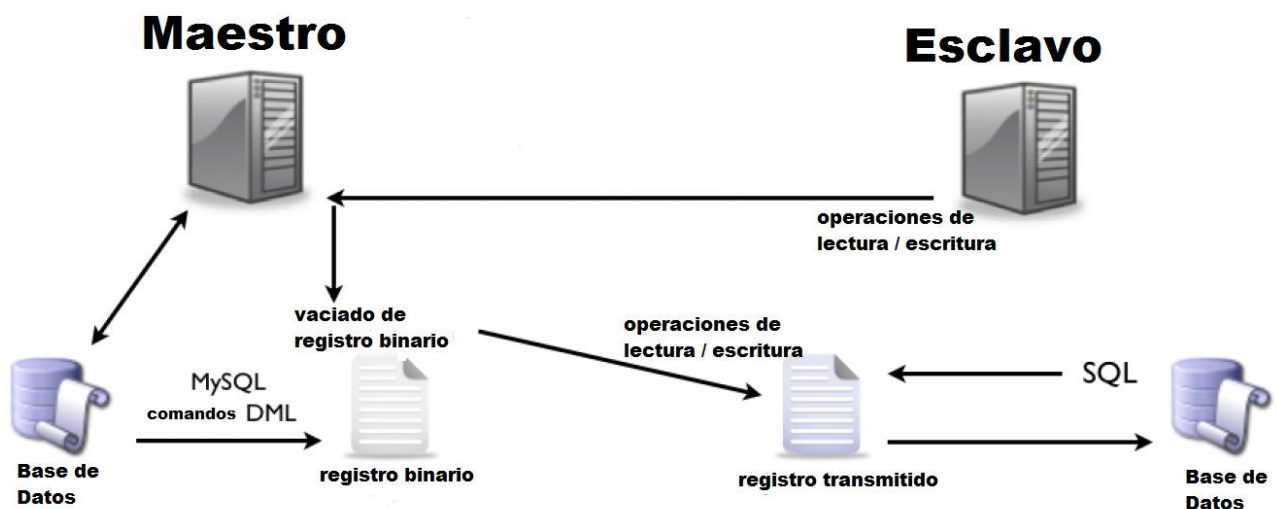
## **3. Patrones de Escenario**

En el mundo contemporáneo el tamaño de las bases de datos crece exponencialmente y se hace inevitable la necesidad de una mejora en el desempeño de la misma (Myalapalli & Savarapu, 2014).

Cuando hablamos de sistemas de recuperación de información es muy importante seguir buenas prácticas al utilizar las sentencias SQL, debido a que el proceso de construcción del índice requiere de una gran interacción con la base de datos.



Por otro lado la replicación de base de datos (Fig. 2) es ampliamente utilizada para la tolerancia a fallos, escalabilidad y desempeño. El fallo en una base de datos con réplica no detiene el sistema. La escalabilidad puede ser alcanzada distribuyendo la carga a través de las réplicas o añadiendo nuevas réplicas si la carga aumenta. Finalmente la replicación de base de datos provee un acceso local rápido incluso si los clientes están distribuidos geográficamente, si las copias están localizadas cerca de los clientes (Kemme, Jimenez-Peris, & Patino-Martinez, 2010).



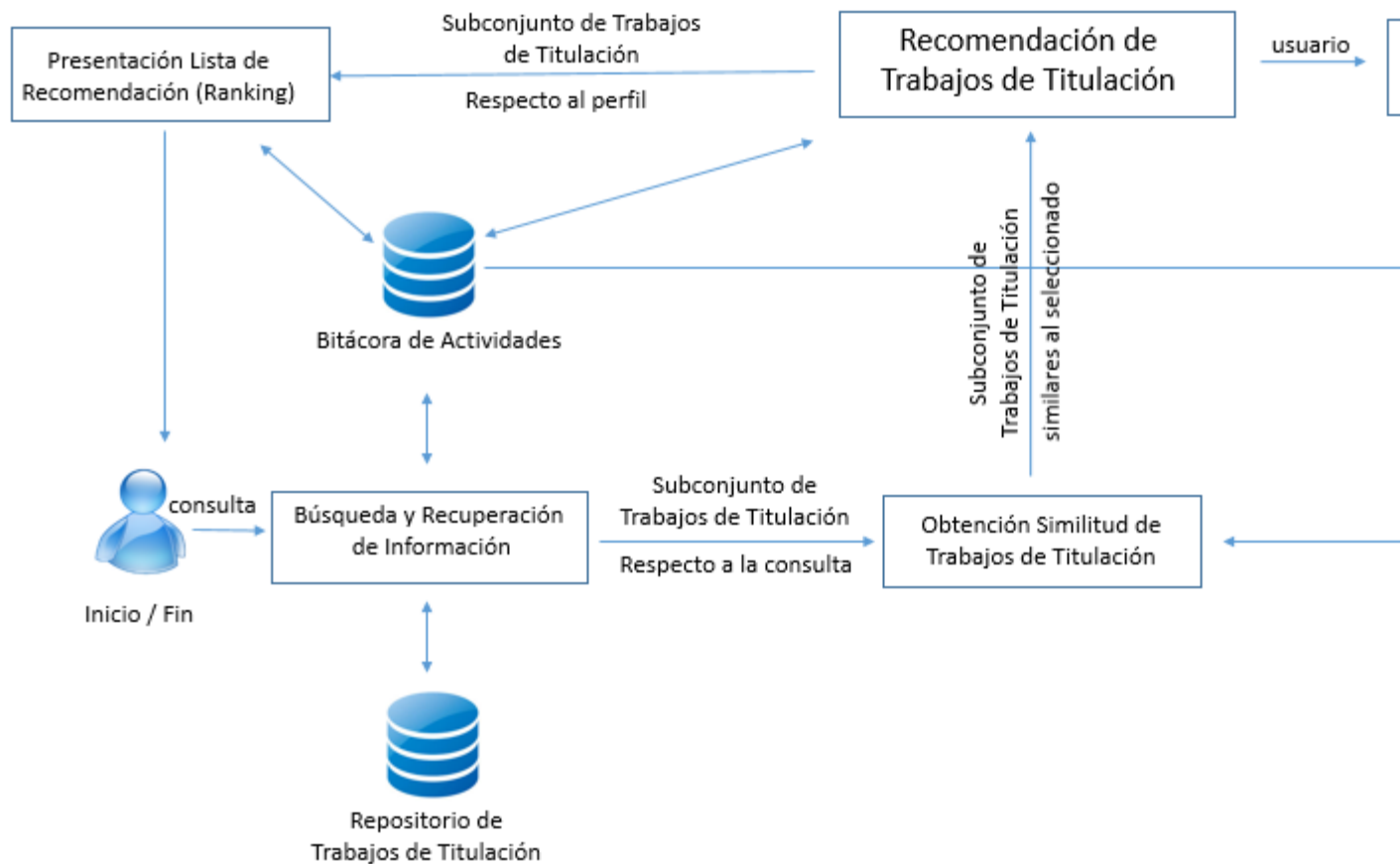
*Figura 2. Esquema de replicación maestro – esclavo de MySql.*

### 3. Caso de Estudio

Esta propuesta considera como caso de estudio el Sistema de Recomendación de Trabajos de Titulación de la Facultad de Ciencias Antropológicas de la Universidad Autónoma de Yucatán (Vences et al., 2015), que se desarrolló sobre el servidor Web Apache y el manejador de base de datos MySQL.

Este sistema presenta una solución para la búsqueda, recuperación y recomendación de trabajos de titulación en dicha institución educativa, la cual cuenta con una matrícula de más de 700 alumnos y cerca de 100 profesores, quienes son beneficiados por el uso del sistema y por lo tanto cualquier

optimización en el desempeño del mismo repercute directamente en la percepción del usuario en cuanto a la rapidez de despliegue de la información y por tanto en su usabilidad.



**Figura 3.** Diagrama de funcionamiento del Sistema de Recomendación.

Como puede observarse, en la Fig. 3 el usuario realiza una búsqueda al sistema, el cual le devuelve una lista de trabajos de titulación relevantes a su consulta y además le proporciona recomendaciones de trabajos similares a aquellos que ha consultado en el pasado o aquellos trabajos que han sido los más consultados.

Referente a las configuraciones de los servidores, para el caso de Apache nos centraremos en las configuraciones relativas al manejo de caché, para lo cual existen módulos que se activarán y adecuarán a nuestras necesidades. Un módulo de Apache es un componente que extiende las funcionalidades del servidor Web

(Apache HTTP Server Reference Manual – for Apache version 2.2.17, 2010). Los módulos `mod_cache_disk`, `mod_expires`, `mod_deflate` son las implementaciones de estas características. Para el caso de MySQL nos referiremos a la replicación del servidor (“MySQL 5.7 Reference Manual,” 2015).

A continuación mencionaremos algunas recomendaciones al momento de ejecutar sentencias SQL que impactan en el tiempo de ejecución, las cuales se encuentran agrupadas en las siguientes categorías: control de caché, compresión de datos y base de datos.

### 3.1 Control de caché

La gestión de los encabezados, especialmente aquellos relacionados con el control de caché del cliente son realizados mediante el módulo Apache `mod_expires`.

Su propósito es controlar la fecha de expiración de los documentos solicitados en las peticiones HTTP, con lo que se puede indicar cuándo un documento fue modificado o su fecha de acceso por parte del cliente. Estos encabezados permiten controlar la persistencia y validez de un documento en el caché del navegador Web.

El módulo es selectivo en la inclusión de encabezados de expiración: si un documento solicitado al servidor tiene un encabezado de expiración entonces no modifica el encabezado.

Se ha establecido que todos los documentos PDF enviados al cliente tengan una fecha de expiración de un año ya que se tratan de trabajos de titulación que no van a cambiar su contenido. Cuando se trate de imágenes, hojas de estilo, scripts, etc., se estableció su caducidad a un mes, debido a que es el tiempo recomendado en la literatura, sin embargo no es una regla, ya que depende de cada propósito en particular. Para activar el control de caché del cliente, se han incorporado las siguientes directivas en el archivo de configuración del servidor Apache:

```
<ifmodule mod_expires.c>
ExpiresActive on
```

```
AddType image/x-icon .ico
<Filesmatch "\.(jpg|jpeg|png|gif|js|css|swf|ico)$">
ExpiresDefault "access plus 1 month"
</Filesmatch>
<Filesmatch "\.(pdf)$">
ExpiresDefault "access plus 1 year"
</Filesmatch>
</ifmodule>
```

Una posibilidad que se tiene con el servidor Apache es configurar su servicio de caché, de manera que todos los documentos requeridos sean almacenados para una rápida recuperación posterior. Esto si bien tiene importancia cuando se habilita un servidor Proxy, tiene una relevancia mayor cuando se utiliza la compresión de datos junto con la generación dinámica de contenidos.

Con un caché de servidor se optimiza la carga de este pues la compresión de los datos se hace una sola vez, evitando realizar la misma operación cada vez que se requiera el mismo archivo.

El servicio tiene la posibilidad de optimizar tanto la frecuencia de actualización del caché y considerar las directivas de control de este en los encabezados, así como la configuración relevante cuando tratamos con contenidos generados dinámicamente con una alta frecuencia de actualización.

Por omisión, todos los documentos almacenados en el caché del servidor tienen una fecha de expiración de una hora. Para activar su control, se han incorporado las siguientes directivas de configuración del servidor Apache que controlan la ubicación del caché y la distribución de los archivos temporales dentro de la estructura del disco:

**Table 1.**Resumen de configuraciones de control de caché y compresión de datos.

Parámetro	Tipo de documento	Valor propuesto	Descripción
mod_expires	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;">PDF</div> <div style="border: 1px solid gray; padding: 5px;">JPG, JPEG, PNG, GIF, JS, CSS, ICO</div>	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;">Un año, ya que los documentos son trabajos de titulación y su contenido no cambia.</div> <div style="border: 1px solid gray; padding: 5px;">Un mes, siguiendo lo recomendado por la literatura.</div>	Controla la fecha de expiración de los documentos solicitados en las peticiones HTTP.
mod_cache	Cualquiera	Una hora es el tiempo establecido por omisión y es el recomendado por la literatura.	Configura el servicio de caché en el servidor. Todos los documentos requeridos son almacenados para una rápida recuperación posterior.
mod_deflate	Cualquiera excepto formatos de archivo ya comprimidos, imágenes y PDF.	Habilitado en el servidor.	Comprime documentos antes de su envío al cliente.

```

<IfModule mod_cache_disk.c>
CacheRoot c:\xampp\tmp
CacheEnable disk /
CacheDirLevels 5

```

```
CacheDirLength
```

3

```
</IfModule>
```

```
</IfModule>
```

## 3.2 Compresión de datos

El servidor web Apache cuenta con un módulo que puede comprimir documentos antes de su envío al cliente. En cuanto al módulo Apache mod\_deflate permite controlar el grado de compresión y su velocidad para no afectar el desempeño del servidor. Sin embargo, no todos los navegadores reconocen la compresión de datos en el intercambio de mensajes HTTP.

Para el caso de nuestro sistema, los documentos que se manejan son PDF y la compresión está dada por el mismo formato de archivo. Sin embargo con la siguiente configuración comprimimos todo excepto imágenes, archivos ya comprimidos y archivos PDF:

```
<IfModule                                mod_deflate.c>
SetOutputFilter                            DEFLATE
SetEnvIfNoCase    Request_URI    \.(?:gif|jpe?g|png)$    no-gzip    dont-vary
SetEnvIfNoCase    Request_URI    \.(?:exe|t?gz|zip|bz2|sit|rar)$    no-gzip    dont-vary
SetEnvIfNoCase    Request_URI    \.pdf$                no-gzip    dont-vary
</IfModule>
```

Los parámetros antes mencionados de control de caché y compresión de datos se resumen en la Tabla 1

## 3.3 Base de datos

Con respecto a la replicación de la base de datos se parte de la premisa de que se tiene dos servidores con MySQL instalados. La dinámica posee la siguiente secuencia:

### 3.3.1 Servidor maestro

Primero en el servidor maestro se edita el archivo de configuración con las siguientes directivas y después se reinicia:

```
server-id=1
log-bin=mysql-bin
innodb_flush_log_at_trx_commit=1
sync_binlog=1
```

Posteriormente se crea un usuario en el servidor maestro que servirá para la replicación con el servidor esclavo:

```
GRANT REPLICATION SLAVE ON *.* TO 'slave_user_name'@'slave_ip'
IDENTIFIED BY 's3cret';
```

Con el objetivo de prevenir la escritura a la base de datos en el servidor maestro se ejecuta el siguiente comando:

```
FLUSH TABLES WITH READ LOCK;
```

También, se considera realizar un respaldo de la base de datos del servidor maestro, para luego restaurarlo en el servidor esclavo. Después se ejecuta el siguiente comando y se anota lo que muestre 'file' y 'position' que será usado en el servidor esclavo más tarde. Adicionalmente, en el servidor maestro se ejecuta el siguiente comando para desbloquear las tablas:

```
SHOW MASTER STATUS;
UNLOCK TABLES;
```

### 3.3.2 Servidor esclavo

Ahora se configura el servidor esclavo, editando el archivo de configuración:

```
server-id=2
```

Se ejecuta el siguiente comando en el servidor esclavo:

```
CHANGE MASTER TO
MASTER_HOST='<direccion_ip_master>',
MASTER_USER='<usuario_esclavo>',
MASTER_PASSWORD='<password>',
MASTER_PORT=3306,
MASTER_LOG_FILE='<file_obtenido_del_master>',
```

```
MASTER_LOG_POS=<position_obtenido_del_master>,  
MASTER_CONNECT_RETRY=10;
```

En el servidor esclavo se restaura el respaldo de la base de datos original y se ejecuta el siguiente comando para iniciar el servicio y con esto finaliza la replicación de base de datos:

```
START SLAVE;
```

Por parte de las recomendaciones para la ejecución de sentencias SQL se pudo identificar que inicialmente, el sistema al momento de crear su índice de palabras generaba una gran cantidad de accesos a la base de datos debido a las sentencias INSERT utilizadas. Se obtiene una reducción significativa del tiempo si se genera la cadena de la sentencia INSERT por documento y se invoca su ejecución una sola vez.

En vez de generar una consulta INSERT por cada palabra por documento:

```
foreach ($palabrasClave as $key => $val){  
mysql_query("insert into palabras (iddoc,palabra,frec) values('".$key."', '".$val."' ");  
}
```

Se genera la cadena de la consulta completa por documento y ejecuta solo una vez:

```
$sqlInsert = "insert into palabras (iddoc,palabra,frec) values ";  
foreach($palabrasClave as $key => $val){  
$sqlInsert.= "(".$sarch."','".$key."','".$val/$maxFrec.""),";  
}  
$sqlInsert = substr($sqlInsert,0,-1);  
mysql_query($sqlInsert);
```

Cabe destacar que antes se realizaban 1,242,953 accesos a la base de datos y ahora se ejecutan 258, por lo tanto hay una reducción del 99.98% de accesos a la base de datos.

Algo similar sucede con las sentencias UPDATE. Cuando queremos optimizar el acceso a la base de datos tenemos que generar la cadena con la consulta y luego



acceder a la base de datos sólo una vez, en lugar de generar un acceso a la base de datos por cada sentencia generada:

```
foreach ($idf as $key => $val){  
mysql_query("update palabras set tfidf = ".$ndoctos/$val." where palabra =  
"$key."");  
}
```

Se genera la cadena de la consulta y se ejecuta solo una vez:

```
$sqlUpdate = "update palabras set tfidf = CASE palabra ";  
foreach ($idf as $key => $val){  
$sqlUpdate.="when ".$key." then ".$ndoctos/$val." ";  
}  
$sqlUpdate.= " end";  
mysql_query($sqlUpdate);
```

## 4. Resultados Experimentales

Dentro del marco de los resultados experimentales se utilizó la herramienta YSlow (“Yahoo! YSlow,” 2015), la cual analiza una página Web y le otorga un grado según una colección de reglas definidas por la empresa Yahoo! para mejorar su funcionamiento. Adicionalmente, YSlow otorga un grado de “A” hasta “F” a una página Web, siendo el mejor valor “A”, según el promedio del grado de cumplimiento de cada una de las reglas del perfil de validación. Estudios han demostrado que el tiempo de respuesta de las páginas web puede ser mejorado de 25% a 50% siguiendo estas reglas.

Yahoo! establece 34 reglas agrupadas en 7 categorías que constituyen una guía para agilizar la descarga y ejecución de una página Web. YSlow implementa las 22 reglas que pueden automatizarse para validar una página.

En la figura 4 (lado izquierdo) se muestra el análisis de la herramienta antes de aplicar las directivas al archivo de configuración de Apache con respecto al manejo

de caché y la compresión de datos. Como se puede observar la herramienta nos ofrece unas recomendaciones a seguir para mejorar el desempeño. En la misma figura 4 (lado derecho) se presenta el resultado del análisis de Yslow después de haber modificado el archivo de configuración de Apache con las directivas de manejo de caché y compresión de datos descritas en la secciones 3.1 y 3.2 respectivamente.

# Grade **A**

Overall performance score 94 Ruleset applied: YSlow(V2) URL: http://localhost/proyecto/

**ALL (23)** FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

<b>A</b> Make fewer HTTP requests	<h3>Grade A on Make fewer HTTP requests</h3> <hr/> <p>Decreasing the number of components on a page reduce reduce the number of components include: combine files, and image maps.</p> <p><a href="#">»Read More</a></p> <p>Copyright © 2015 Yahoo! Inc. All rights reserved.</p>
<b>C</b> Use a Content Delivery Network (CDN)	
<b>A</b> Avoid empty src or href	
<b>E</b> Add Expires headers	
<b>B</b> Compress components with gzip	
<b>A</b> Put CSS at top	
<b>A</b> Put JavaScript at bottom	
<b>A</b> Avoid CSS expressions	
<b>n/a</b> Make JavaScript and CSS external	
<b>A</b> Reduce DNS lookups	
<b>A</b> Minify JavaScript and CSS	
<b>A</b> Avoid URL redirects	
<b>A</b> Remove duplicate JavaScript and CSS	
<b>A</b> Configure entity tags (ETags)	
<b>A</b> Make AJAX cacheable	
<b>A</b> Use GET for AJAX requests	
<b>A</b> Reduce the number of DOM elements	
<b>A</b> Avoid HTTP 404 (Not Found) error	
<b>A</b> Reduce cookie size	
<b>A</b> Use cookie-free domains	

ALL (23) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

<b>A</b> Make fewer HTTP requests	<b>Grade A on Make fewer HTTP requests</b>  Decreasing the number of components on a page reduces the number of components include: combine files, combin maps.  <a href="#">»Read More</a>  Copyright © 2015 Yahoo! Inc. All rights reserved.
<b>C</b> Use a Content Delivery Network (CDN)	
<b>A</b> Avoid empty src or href	
<b>A</b> Add Expires headers	
<b>A</b> Compress components with gzip	
<b>A</b> Put CSS at top	
<b>A</b> Put JavaScript at bottom	
<b>A</b> Avoid CSS expressions	
<b>n/a</b> Make JavaScript and CSS external	
<b>A</b> Reduce DNS lookups	
<b>A</b> Minify JavaScript and CSS	
<b>A</b> Avoid URL redirects	
<b>A</b> Remove duplicate JavaScript and CSS	
<b>A</b> Configure entity tags (ETags)	
<b>A</b> Make AJAX cacheable	
<b>A</b> Use GET for AJAX requests	
<b>A</b> Reduce the number of DOM elements	
<b>A</b> Avoid HTTP 404 (Not Found) error	
<b>A</b> Reduce cookie size	
<b>A</b> Use cookie-free domains	
<b>A</b> Avoid AlphaImageLoader filter	

*Figura 4. Comparativa de resultados de la herramienta YSlow.*

Se puede observar en la figura 4 cómo las dos reglas que nos interesan, Add Expires headers (E) y Compress components with gzip (B), mejoraron su resultado (A) al aplicar las configuraciones mencionadas de compresión y control de caché; Al mismo tiempo el desempeño general pasó de 94 a 98.

El desempeño general es un porcentaje ponderado de las calificaciones individuales de cada regla. Las calificaciones de cada regla individual son calculadas de manera diferente dependiendo de la regla. Por ejemplo, para la regla 1, tres scripts externos son permitidos. Por cada script extra que se tenga se restan cuatro puntos de la calificación de esa regla, que tiene un peso por defecto de ocho.

La herramienta Yslow! permite además generar información con respecto al comportamiento de una página en términos de su tamaño y tiempos de transferencia. La tabla 2 presenta una comparativa de tiempos de despliegue de información y de volumen de datos transferidos. En la primera columna (Normal) se puede observar el tiempo que tarda en cargar la página web con la configuración por defecto de Apache, mientras que en la segunda columna (Optimizado) muestra el tiempo que tarda en carga la misma página pero habiendo aplicado las directivas de control de caché y compresión de datos en el archivo de configuración de Apache.

**Table 2.** Resultado de las prueba de validación.

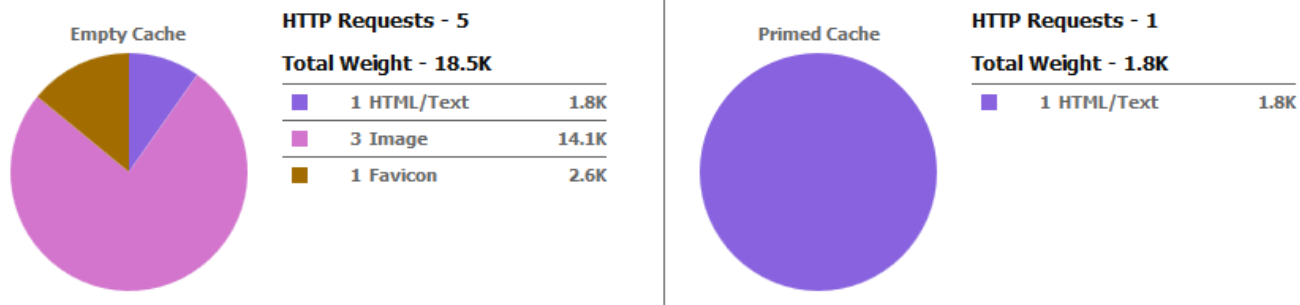
	<b>Normal</b>	<b>Optimizado</b>
Optimizado	173 ms	87 ms
Datos transferidos inicialmente	18.6 Kb	15.5 Kb
Datos transferidos posteriormente	4.8 Kb	1.7 Kb

A continuación se presentan gráficos de pastel que resaltan el beneficio de utilizar el caché y la compresión, en relación al volumen de datos que es transferido la primera vez que se accede a la página (18.6 Kb) sin compresión contra una visualización posterior de la misma página utilizando la configuración optimizada (Figura 5). En este último caso, el volumen de transferencia se reduce hasta ser de

1.7 Kb (reducción del 90.8%), que es una diferencia muy significativa en términos de percepción del usuario (la página se presentará más rápido).

**Statistics** The page has a total of 5 HTTP requests and a total weight of 18.5K bytes with empty cache

**WEIGHT GRAPHS**

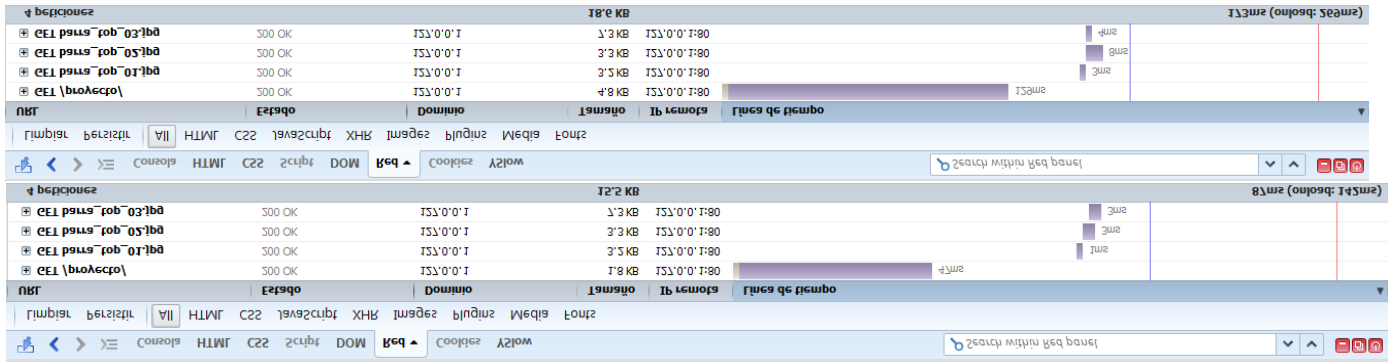


**Figura 5.** Estadísticas de descarga de la página inicial aplicando las configuraciones (gráfico generado por Yslow).

Para analizar el tiempo de descarga de la página se utilizó la herramienta FireBug que incorpora nuevas funcionalidades en un navegador Web, las cuales están orientadas al desarrollo y depuración de aplicaciones Web.

Se ha generado un reporte de duración (Figura 6) que brinda información sobre el tiempo que tarda el navegador Web en transferir y presentar los distintos recursos que conforman a la página Web. En total, el navegador ha tardado 87 ms (parte inferior) en desplegar la página, contra los 173 ms (parte superior) que si se visualizara la misma página sin considerar la compactación. Una mejora de poco más del 50%. Como se puede observar en la figura, las imágenes que ya son un formato comprimido no sufren modificación, sin embargo la página principal se reduce con la compresión de datos de 4.8 Kb a 1.8 Kb.

Cabe señalar que estos tiempos no son fijos, tienen variaciones que dependen del ancho de banda de la conexión y el grado de saturación de paquetes en la misma (no es la misma velocidad de acceso a una página en distinto horario de un mismo día). Sin embargo, la estadística generada da buena cuenta de los beneficios que aporta utilizar un esquema de optimización de aplicaciones Web.



**Figura 6.** Comparativa del tiempo de descarga, al utilizar compresión de datos, de la página principal analizada con Firebug.

## 5. Conclusiones

En este trabajo se demuestra como combinando distintas configuraciones al servidor web Apache, como son el manejo de caché y la compresión de datos y siguiendo buenas prácticas al utilizar las sentencias SQL en nuestro manejador de base de datos, se obtiene una reducción significativa (poco más del 50%) en los tiempos de respuesta de nuestro sistema, así como también una reducción en el volumen de datos transferidos (reducción del 90.8%) entre el servidor y el cliente, lo cual tiene una repercusión inmediata en la percepción del usuario ya que la página se despliega más rápido.

Por otro lado, se demuestra que para los casos donde la interacción del sistema con la base datos es primordial y requiere un gran uso de ella, como lo es la creación de índices para un sistema de recuperación de información, y siguiendo las recomendaciones en la construcción y ejecución de sentencias INSERT, se logra una reducción significativa (reducción del 99.98%) en los accesos a la base de datos lo que repercute directamente en una reducción en el tiempo de procesamiento.

Si bien se analizaron los resultados para nuestro caso de estudio, se puede afirmar que los mismos beneficios serán obtenidos siempre y cuando la plataforma bajo la que funcione cualquier otro sistema sea la misma, es decir, Apache y MySql, ya que las directivas de control de caché y compresión de datos se pueden

configurar para otros tipos de archivos, sin embargo, lo determinante es seleccionar la configuración correcta de los parámetros del servidor web Apache.

## Referencias

Ahuja, S., Wu, T. W. T., & Dixit, S. (2003). On the effects of content compression on Web cache performance. *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*. <http://doi.org/10.1109/ITCC.2003.1197573>

Ahuja, S., Wu, T. W. T., & Dixit, S. (2003). On the effects of content compression on Web cache performance. *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*. <http://doi.org/10.1109/ITCC.2003.1197573>

Apache HTTP Server Reference Manual – for Apache version 2.2.17. (2010). EUA: Network Theory Ltd.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. England: Addison-Wesley Longman Publishing Co., Inc.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132. <http://doi.org/10.1016/j.knosys.2013.03.012>

Gandhi, N., Tilbury, D. M., Diao, Y., Hellerstein, J., & Parekh, S. (2002). MIMO control of an apache web server: Modeling and controller design. *Proceedings of the American Control Conference*, 6, 4922–4927. <http://doi.org/10.1109/ACC.2002.1025440>

Jun Li, & Menghan Lu. (2013). The performance optimization and modeling analysis based on the Apache Web Server, 1712–1716.

Karin Verspoor, K. B. C. (2013). Natural Language Processing. In *Encyclopedia of Systems Biology* (pp. 1495–1498). Springer New York. [http://doi.org/10.1007/978-1-4419-9863-7\\_158](http://doi.org/10.1007/978-1-4419-9863-7_158)



Kemme, B., Jimenez-Peris, R., & Patino-Martinez, M. (2010). Database Replication. Morgan & Claypool. <http://doi.org/10.2200/S00296ED1V01Y201008DTM007>

Manning, C., Raghavan, P., & Schütze, H. (2009). An Introduction to Information Retrieval. Cambridge, England: Cambridge University Press.

Manning, C., & Schutze, H. (1999). Chapter 11. Probabilistic Context Free Grammars. Foundations of Statistical Natural Language Processing, 381–404.

Menéndez, V. H., Castellanos, E., Aguilar, R. A., & Gómez, S. (2013). Optimización del Desempeño de una Plataforma e-Learning mediante Técnicas no Invasivas. Congreso Internacional de Investigación Academia Journals, 5, 2188 –2193.

Myalapalli, V. K., & Savarapu, P. R. (2014). High Performance SQL Finesse for Lucrative Programming. Annual IEEE India Conference (INDICON).

MySQL 5.7 Reference Manual. (2015). Retrieved from <http://dev.mysql.com/doc/refman/5.7/en/>

Smith, P. (2013). Professional Website Performance: Optimizing the End and the Back End. Indianapolis, Indiana: John Wiley & Sons, Inc.

Souders, S. (2007). High Performance Web Sites: Essential Knowledge for Frontend Engineers. EUA: O'Reilly Media, Inc.

Vences, R., Menéndez, V. H., & Zapata, A. (2015). Sistema de Recomendación de Trabajos de Titulación de la Facultad de Ciencias Antropológicas de la Universidad Autónoma de Yucatán. In R. Juárez, G. César, H. Jadwiga, G. Ibargüengoitia, C. Fernández, G. Licea, & S. Vázquez (Eds.), Tendencias en Investigación y Aplicaciones Prácticas de la Ingeniería de Software (pp. 3–8). San Luis Potosí, México.

W3C. (2010). Web Accessibility Initiative (WAI). Retrieved from <http://www.w3.org/WAI/>

Yahoo! YSlow. (2015). Retrieved from <http://developer.yahoo.com/yslow/>

## Notas biográficas:



**Rodrigo Vences Nava** es Licenciado en Ciencias de la Computación por la Universidad Autónoma de Yucatán, México. Actualmente se encuentra cursando la Maestría en Ciencias de la Computación de la Universidad Autónoma de Yucatán, México. Es Administrador de Tecnologías de Información de la Facultad de Ciencias Antropológicas de la Universidad Autónoma de Yucatán, México. Su trabajo de investigación se centra en temas relacionados con repositorios de tesis electrónicas, sistemas de recomendación y minería de datos.



**Víctor Hugo Menéndez Domínguez** es Doctor en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha, España. Es Profesor Titular en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán, México. Su trabajo de investigación se centra en temas relacionados con repositorios de documentos digitales, la representación del conocimiento y la gestión de Objetos de Aprendizaje.



**Alfredo Zapata González** es Doctor en Tecnologías Informáticas Avanzadas por la Universidad de Castilla-La Mancha, España. Es Profesor Titular en la Facultad de Educación de la Universidad Autónoma de Yucatán, México. Su trabajo de investigación se centra en temas relacionados con Minería de datos y Sistemas de Recomendación para la Educación a distancia.



Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 2.5 México.

COMPUTACIÓN E INFORMÁTICA

*ReCIBE, Año 5 No. 2, Mayo 2016*

## **Building an ontological modeling mechanism for virtual cities**

Luis Casillas

Computer Science Department, University of Guadalajara,  
Mexico

[luis.casillas@cucei.udg.mx](mailto:luis.casillas@cucei.udg.mx)

Gilberto Vargas

Computer Science Department, University of Guadalajara,  
Mexico

[tachoguitar@gmail.com](mailto:tachoguitar@gmail.com)

Miguel Lares

Computer Science Department, University of Guadalajara,  
Mexico

[c.code.jp@gmail.com](mailto:c.code.jp@gmail.com)

**Abstract:** Modeling of cities is certainly a challenge. Most of this complexity is caused by the significant number of variables, as well as the vast manifestation for their ranks. Diverse efforts have been conducted by different researchers, in order

to achieve powerful representations. Some of these efforts are aimed at graphical representation, some others are aimed at functionality. Present effort is aimed at the scaffolding support for the representation and, eventually, functionality of virtual cities. Such infrastructure would support current and future efforts devoted to populating virtual cities with agents showing different behaviors and roles. These agents would be active and concrete elements in cities, such as pedestrians, cars or drivers, and some others would be dynamic synthetic-elements, such as traffic lights or drawbridges. These agents will populate the city model and manifest in future efforts from this research; current effort is mainly focused in the production of cities' infrastructure. The generation and representation of a virtual city is, per se, a significant challenge. This paper focuses in such challenge.

**key words:** Ontology, Knowledge Management, Virtual City, Simulation, Voronoi-diagram, Delaunay-triangulation.

## **Construcción de un mecanismo de modelado ontológico para ciudades virtuales**

**Resumen:** El modelado de ciudades es ciertamente desafiante. La mayor causa de esta complejidad es el significativo número de variables, así como la vastedad en la manifestación de sus rangos. Diversos esfuerzos han sido realizados por diferentes investigadores con el fin de lograr representaciones contundentes. Algunos de estos esfuerzos se han orientado a la representación gráfica y algunos otros a la funcionalidad. El presente esfuerzo está orientado al andamiaje que soporta la representación y, eventualmente, la funcionalidad de ciudades virtuales. Tal infraestructura soportaría los esfuerzos actuales y futuros dedicados a poblar la ciudades virtuales con agentes que muestren diferentes comportamientos y roles. Estos agentes serían elementos activos y concretos en la ciudades, tales como automóviles o conductores, y otros serían elementos sintéticos – dinámicos tales como semáforos o puentes levadizos. Estos agentes poblarán el modelo de ciudad y se manifestarán en futuros emprendimientos de esta misma investigación; el trabajo actual se enfoca principalmente a la producción de la infraestructura de la ciudad.

La generación y representación de la ciudad virtual es, per se, un esfuerzo desafiante. Este artículo se enfoca en tal esfuerzo.

**Palabras clave:** Ontología, Gestión del Conocimiento, Ciudad Virtual, Simulación, Diagrama de Voronoi, Triangulación de Delaunay.

# 1. Introduction

There are diverse challenges when trying to model virtual cities, due the variety of actors in real cities. Regarding real cities, most of these actors are understood as city elements. These elements are, in most of cases, disabled to change the city structure, its topology or even its functionality. Nevertheless, cities have some active elements aimed at modifying the city: municipalities, urban-planners, real-estate developers, and even natural events such as earthquakes, hurricanes, twisters, etc. These active elements eventually would produce deep-impact effects on mobility for citizens and/or vehicles. Because of the lack of information from real cities, it is difficult to grasp options and tools to simulate the real stress and the whole phenomena in a big city.

The present effort is aimed at producing an ontological structure for representing virtual cities. The primary motivation for this effort, is the lack of updated data, and in most cases the lack of data at all. Our approach is inspired by similar efforts as (Davies, Duke, & Sure, 2004), but the proposal in this paper is based on real communities instead of web sources. Most of cities are not enabled to collect every data produced by every single element in the city. Some cities are enabled to collect data from diverse phenomena happening inside the city boundaries, nevertheless lots of signaling and corresponding data is simply lost. Hence, every analysis performed over those collected data will lack of elements that somehow produce an influence. It is rather difficult to find out patterns when required data is not enough.

The urban model of big cities need to be often changed. Unfortunately such changes will not be made as frequently as required. But even when changes were required by city's inhabitants, such modifications usually face enormous resistance from social, economic, and political groups in city. When considering upgrades to

the urban environments it is needed to know what the best ways to improve the urban planning are. Throughout an effort as the presented in this paper, there is going to be an enabled tool to produce, manage and even control diverse aspects of virtual cities; in order to have a synthetic representation of a city. The main goal of this research is to find out ways for simulating virtual cities, as well as artificial agents inside, which will learn the best practices to meet their needs. Thus, this paper presents the stage devoted to produce a description for logical definitions, as well as the geometrical algorithms to generate the physical corpus of cities.

## 2. Related Work

Due to complexity bound to cities' emulation, because the number of variables and their wide ranges, several efforts have been done along the time. Such aspect is deeply reviewed by (Gandy, 2005). Some of the tools for city emulation are focused on the graphical representation, and some others focus on inner structures devoted to support the model's functionality. The modeling mechanism presented in this paper, is somehow focused on the structural approach; according to the following review.

Due to complexity bound to cities' emulation, because the number of variables and their wide ranges, several efforts have been done along the time. Such aspect is deeply reviewed by (Gandy, 2005). Some of the tools for city emulation are focused on the graphical representation, and some others focus on inner structures devoted to support the model's functionality. The modeling mechanism presented in this paper, is somehow focused on the structural approach; according to the following review.

Based on the previous statement, despite the vast quantity of tools to help urban modeling, there is a lack of tools devoted to simulate study-cases focusing in city operation, instead of its mere visualization; since modern tools frequently focus in 3D simulation and do not have enough power to deal with issues regarding how chaotic and erratic a real city is.

Based on the previous statement, despite the vast quantity of tools to help urban modeling, there is a lack of tools devoted to simulate study-cases focusing in city

operation, instead of its mere visualization; since modern tools frequently focus in 3D simulation and do not have enough power to deal with issues regarding how chaotic and erratic a real city is.

This first stage of this city simulator is aimed at the representation for the logic information from the city and the geometrical algorithms to generate pseudo-random cities. We are working from inside to out. A problem faced by this project, is the availability of data and information coming from real cities in the world. When data and information exist: they are not publicly available, they are not enough, they are not precise, they are polluted, and etcetera. In the rest of cases there are neither data nor information available. To deal with this annoying issue, we have worked in a sophisticated algorithm that is able to produce, as an output, a complex network which represents a possible structure for a virtual city. Such algorithm possess a series of parameters aimed at affecting the structure, organization and, even, operation of the virtual representation of the city. By these controlling mechanisms over cities' construction, the virtual representation of cities keeps as chaotic as real city could be.

Based on (Fortune, 1987), our algorithm approach has an inspiration from Voronoi's diagrams, in order to produce chaotic polygons. Such mesh of polygons represents an ensemble of regions, such as diverse regions and spaces inside a city. Then, every region is populated with streets. Streets population is produced by the creation of a grid inside every polygon (region). Avenues were made with a fractal algorithm, inspired in the efforts made by Parish & Müller (2001), where they used the same algorithm; although these authors had a geographical structure as an input, and an additional algorithm for building a city using L-Systems. Another interesting aspects from Parish & Müller (2001), is that they also divided the blocks on houses and generated a 3D view of the whole city, the focused on the map view as an output, we will focus on the information representation. The effort performed in the present study is not aimed at producing a complex 3D view as (Döllner & Hagedorn, 2007), instead this work is aimed at the inner handling of structures and models representing the city. The work (Courtat, Gloaguen, & Douady, 2010) models up a novel understanding for cities' virtual representation. Throughout the

use hypergraphs, authors were able to represent each street as hyper-relationship between a set of segments. As proposed in (Ge, Meng, Cao, Qiu, & Huang, 2014), authors used a different approach for simulation, which consists in giving a treatment as a macro level where social events and statistics are simulated - instead of micro events related to population gender, age, occupation, etc. Our proposal is also based on generating artificially sets of data, which replaces the absences of real data from unsupervised cities.

### 3. Representing knowledge from cities

There is a confusion regarding what is, in fact, knowledge. This is caused by the resistance offered by the concept “knowledge” itself to be defined. A practical approach to solve the emerging complexity regarding knowledge definition, is to visualize signals from environment as data. Such data, signals from environment, are paired with descriptions stored in brains. When pairing is successfully mated with experience, stored in brains, an amount of information is release. This information is the explanation given by observer to signal retrieved. Hence, the experience is the catalyzing agent for signals processing. This experience could be understood as knowledge. As explained in (Casillas, Daradoumis, & Caballé, 2013).

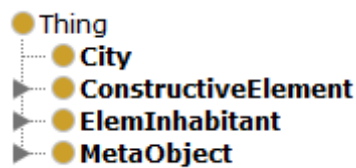
Every phenomenon, surrounding us, could be understood and dealt as an amount of knowledge. Knowledge includes a passive portion, made by data and information regarding the phenomenon; and an active portion, made by functionality and behaviors shown by actors in observed phenomenon. This passive/active composition is the most interesting quality of knowledge. Both aspects are bound and deeply mixed in knowledge. A city is actually a phenomenon susceptible of knowledge description. Cities mix passive and active elements.

In order to fulfill the challenging task of city modeling, from the approach presented in previous paragraph, our proposal has been supported by a powerful tool: Protègè . This instrument implements a series of services and machineries aimed at producing enriched models for reality. Protégé’s is a plug-in architecture which



could be adapted to build both simple and complex ontology-based applications. Developers are enabled to integrate the output of Protégé with rule systems or other problem solvers to construct a wide range of intelligent systems.

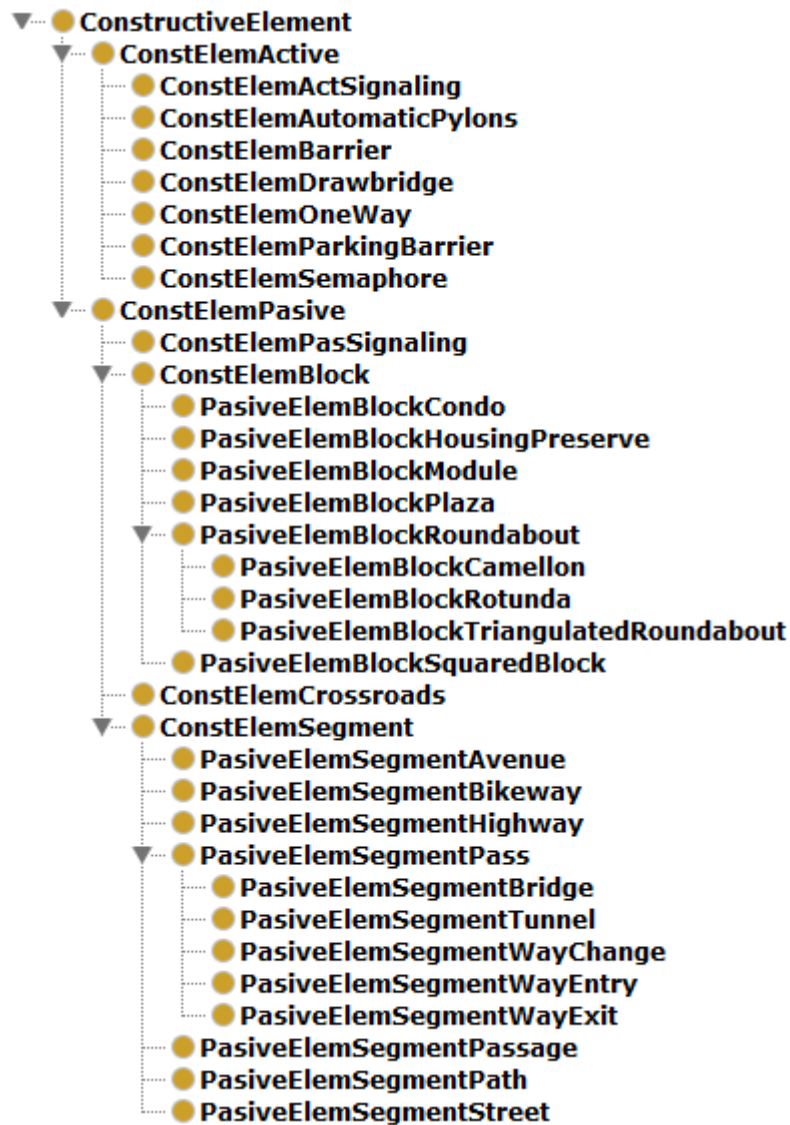
Gruber (1993) had originally defined a highly cited notion for an ontology as an “explicit specification of a conceptualization”. An ontology behaves as a bridge between knowledge representation and the corresponding information management. It is frequent to use semantic networks as a mechanism to model up ontologies. The research effort presented in this work, focuses in three kinds of aspects in order to represent a city. Figure 1 shows this preliminary decomposition



*Figura 1. Main knowledge categories for city representation*

### 3.1 Constructive elements

This knowledge category tries to represent the awareness regarding physical elements in a city, such as: blocks, cross-intersections, highways, streets, bridges, etc. by classifying them in two subclasses: active and passive elements. Figure 2 sketches the corresponding decomposition for these active and passive elements, involving all possible/regarded actors in nowadays cities.

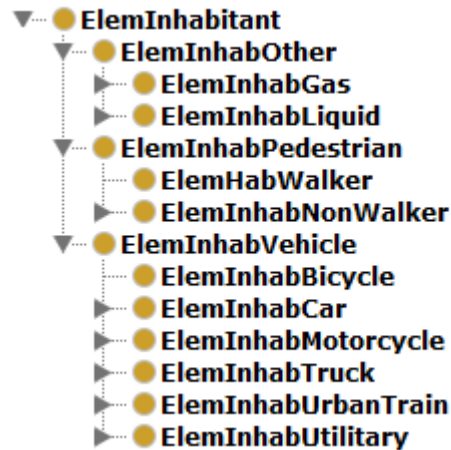


**Figura 2.** Taxonomical decomposition for the Constructive Element aspect from cities' representation proposed. Includes the corresponding decomposition for active and passive elements.

### 3.2 Inhabitant elements

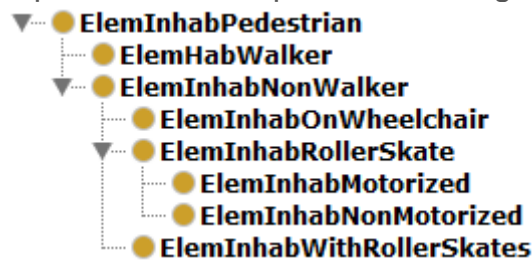
While some other researchers reduce the domain ontology about the inhabitants in a city (Zhai, Wang, & Lv, 2008), our ontology tries to be closer to the real manifestation of cities by allowing pedestrians to move around the city. Citizens could eventually move through vehicles as: car, bus, bicycle, train, and etcetera. Besides, our model for cities is enabled to include other non-regular inhabitants.

Due to complexity of these knowledge categories, the knowledge decomposition analysis will be made by sections. Figure 3 sketches a preliminary decomposition of the inhabitant aspect. Reader is invited to admit our novel approach for inhabitants, including some unalive entities.



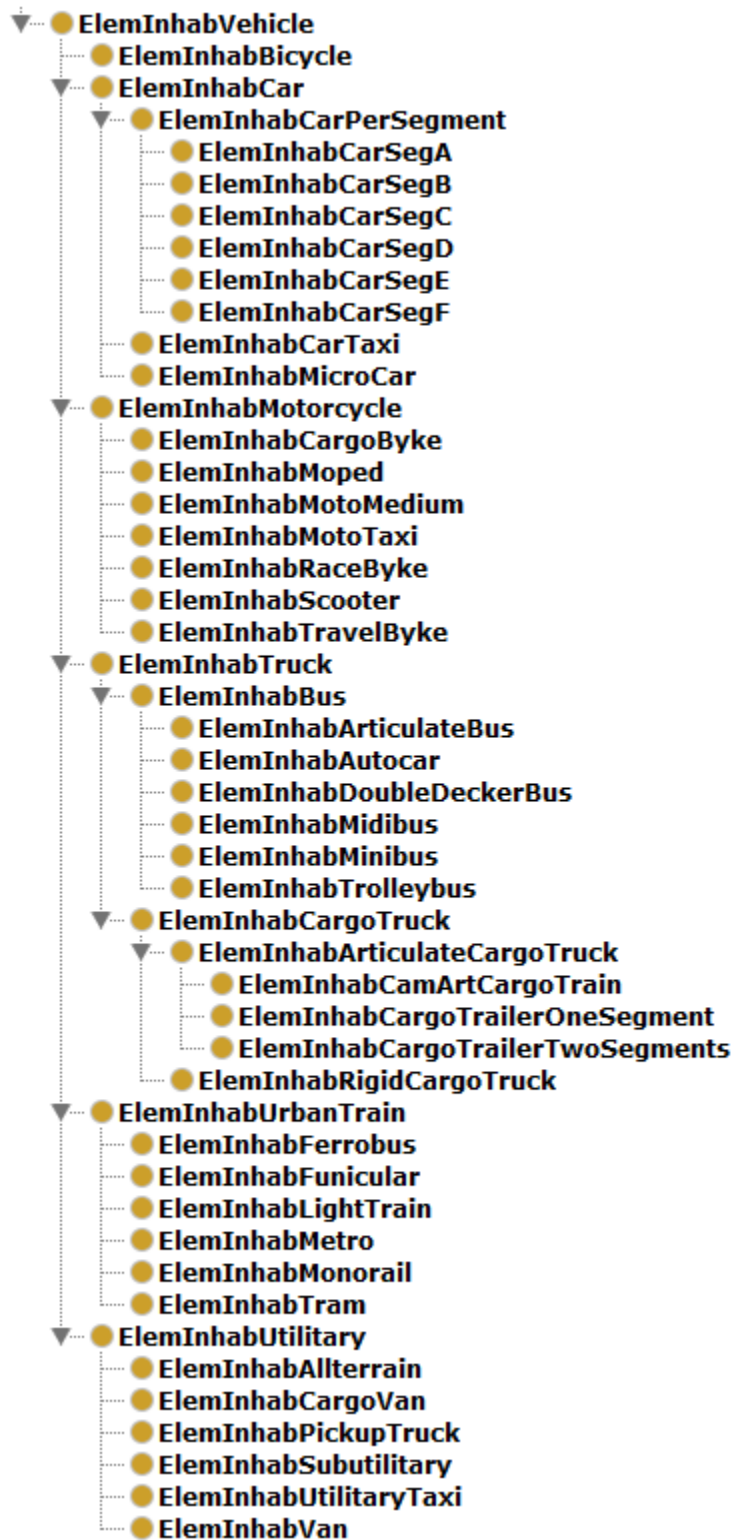
**Figura 3.** Taxonomical decomposition for the Inhabitant Element aspect from cities' representation proposed. Includes the corresponding decomposition for Pedestrian, Vehicle and Other inhabitants in the city.

Regarding pedestrian inhabitants category, there are two subcategories: walker and non-walker. The case of walker refers to humans walking or running on their own feet. The non-walker case contains different situations of humans moving on certain mechanisms or artifacts allowing slow-speed mobility among pedestrians. Figure 4 shows a decomposition of this specific subcategory.



**Figura 4.** Taxonomical decomposition for the Pedestrian Inhabitant Element aspect from cities' representation proposed. Includes the corresponding decomposition for Walker and Non-Walker inhabitants in the city.

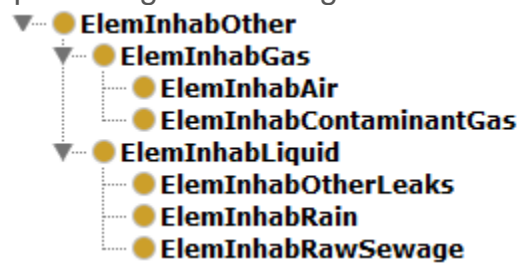
The vehicle category is actually and that could be moot. Nevertheless, vehicle al inhabitants in a city. Of course, they have no conscience or desires. Vehicles are unaware of their roles in the city, but they in fact have roles. Hence, they were considered as inhabitants of a city. Figure 5 models up a sketch of the vehicle's decomposition for our city ontology.



*Figura 5. Taxonomical decomposition for the Vehicle Inhabitant Element aspect from cities' representation proposed. Includes the corresponding decomposition for*

*Bicycle, Car, Motorcycle, Truck, Urban Train and Utilitarian cases of inhabitants in the city.*

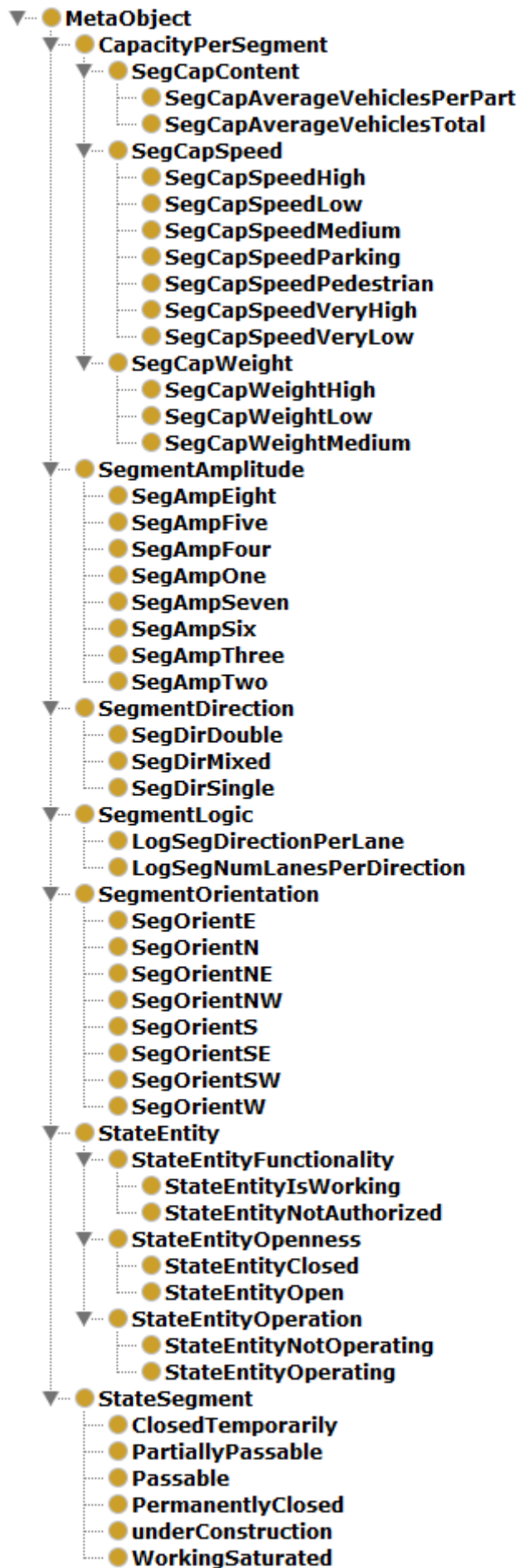
Another innovative aspect in our proposal, is the presence of the “Other” category regarding inhabitants in the city. Pedestrians and vehicles have been already considered in the ontology, but our team considered useful to have another kind of inhabitant. Of course, there could be unlimited cases of inhabitants in a city; even some cases unknown for the authors of this work. This is not the goal of the "other" category for inhabitants. The other category is composed by unanimated elements, which are present in a city and produce expected and unexpected effects. It is mainly represented by liquids and gases. This is a strong aspect of our ontology, which considers contamination, rain, raw sewage, leaks, etc. since those elements may be dangerous to society and lead to accidents in many cases they are not considered during cities synthetic representation. Figure 6 shows a decomposition of such category into liquid and gas subcategories.



**Figura 6.** *Taxonomical decomposition for the Other Inhabitant Element aspect from cities' representation proposed. Includes the corresponding decomposition for Liquid and Gas cases of inhabitants in the city.*

### 3.3 Meta Objects

Beyond the analysis presented up to the moment, we considered as important an extra branching regarding the city ontology. This branch would contain all the awareness regarding interaction, control, description, operation, etc. of city elements regarded in the ontology description. This extra branch has been named as meta-objects and refers to all information describing all the aspects and categories in our ontology for cities' representation. Figure 7 presents the elements included in such description mechanism.



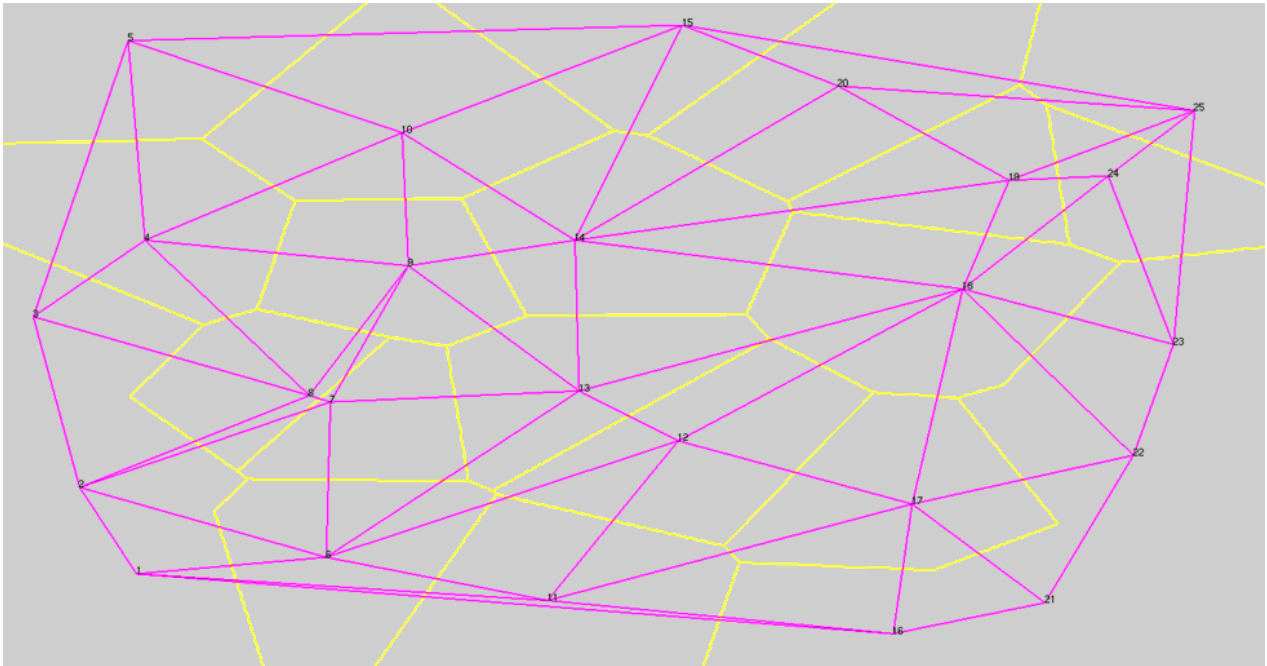
**Figura 7.** Taxonomical decomposition for the Meta Objects aspect from cities' representation proposed. Contains special classes designated to manage and/or handle diverse operation, control and interaction involving city aspects previously analyzed.

## 4. City generation

Once the ontology has been defined, we focused on the mechanisms enabling the generation of the city. As mentioned before in this work, it is not always possible to have real data or information from cities. Hence, our efforts were aimed at producing the frame of streets and avenues conforming the city's regions. By producing an algorithm enabled to create the city frame. This algorithm could reply to the question: "how can we build streets in a chaotic way but with some sort of regular patterns, which make sense as in real cities?" We found that Voronoi diagrams could help to solve problem, as long as we fix an ensemble with other algorithms.

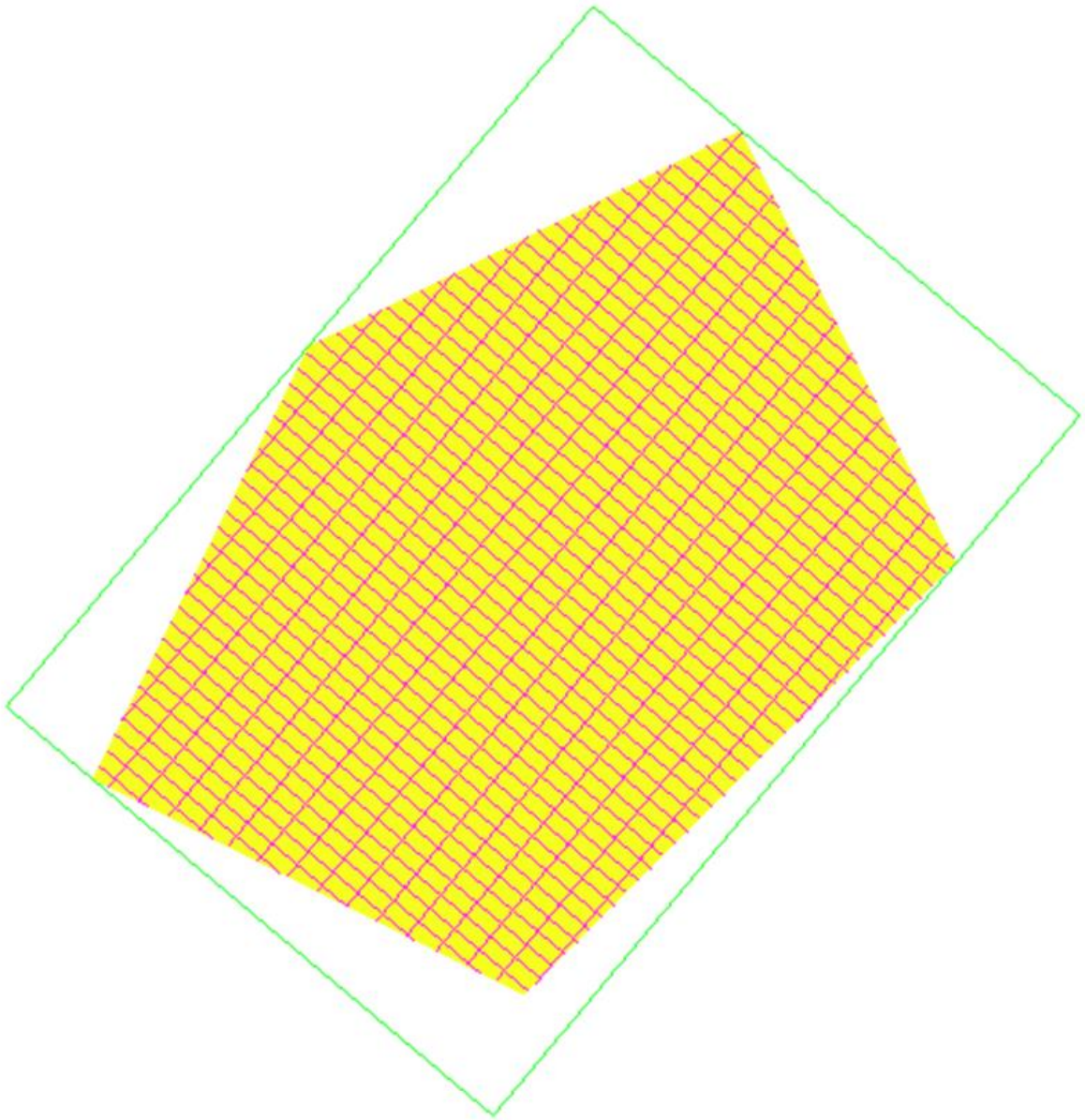
A Voronoi diagram is a partition of an infinite plane that considers a set of points  $P$  to calculate some inner polygons called "voronoi regions", each point is related to one region. Regions are defined as the set of points  $S$  that are closest to  $P_i$  for all the points in  $P$ . The most recommended strategy to produce a Voronoi diagram is the sweep line algorithm of Steve Fortune (1987), which has an  $O(n \log n)$  complexity. In order to create a Voronoi diagram, the Delaunay triangulation is calculated. The next step is getting the regions from such triangulation. These regions can be understood as city colonies. Figure 8 shows the results of running Fortune's Algorithm.





**Figura 8.** After running Fortune's Algorithm, these are the regions created. In the figure, purple lines represent the Delaunay triangulation and yellow lines contain the voronoi regions.

Once regions have been generated, they should be populated with streets. There could be used several mechanisms in order to create streets inside every region; from regular squared distribution, up to complex fractal algorithms to produce branches. At this point of the project, regions will be populated with streets using rectangles. In order to avoid a squared and predictable distribution of streets among regions, every polygon representing a region is randomly rotated and then filled up with rectangles. Figure 9 shows an example of this effort for populating regions with streets.

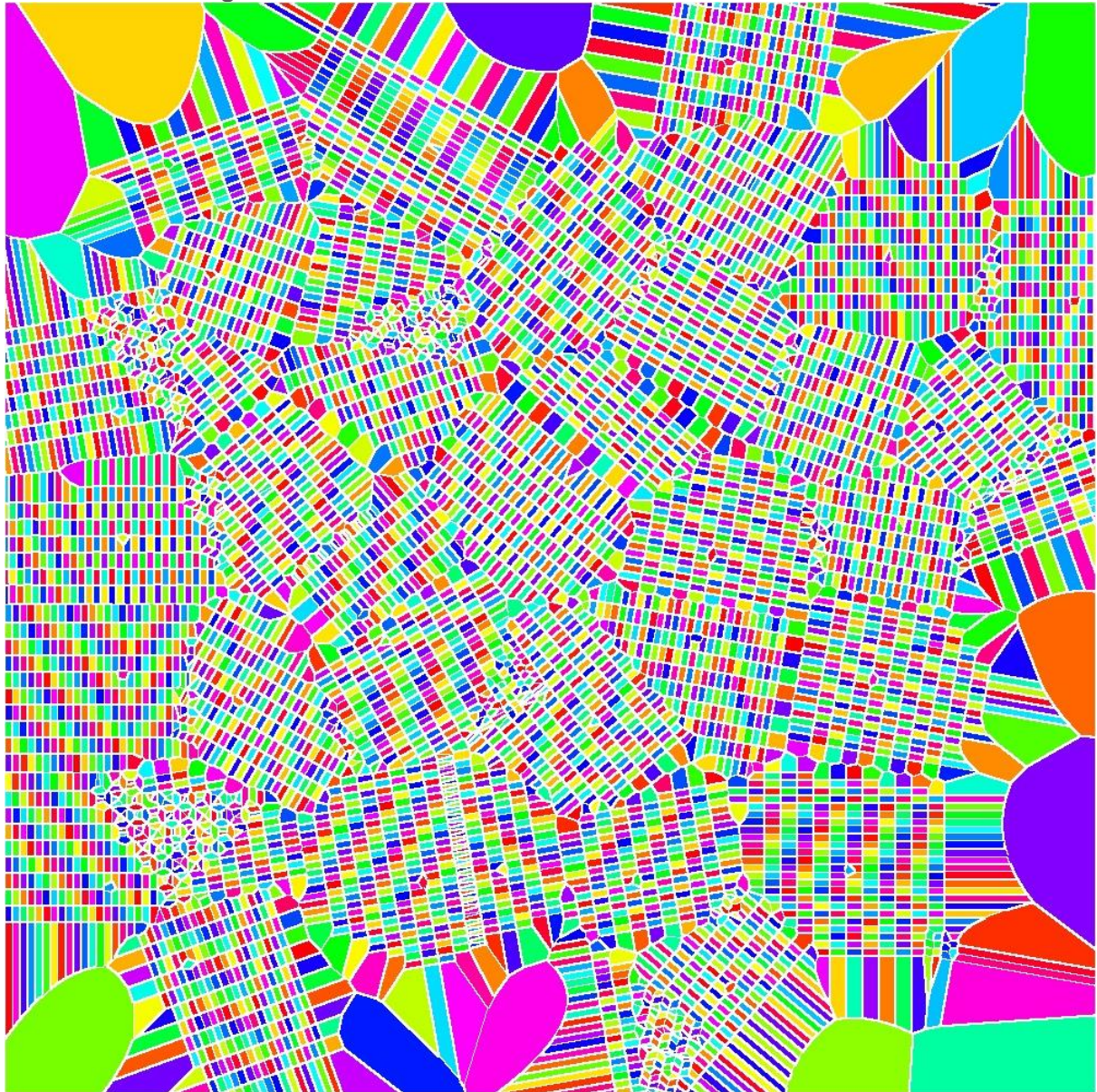


**Figura 9.** *An example of populating a polygon (city region) with rectangles, which represent the inner streets. The region was randomly rotated and populated with rectangles, producing the inner elements.*

The process is repeated over and over, involving every voronoi region created. Thus, all map regions are populated, producing an non-regular grid. Such structure clearly resembles a real city map. Figure 10 shows such results. Reader should



consider that there are different parameters for producing these elements, hence the regions' size as well as the populating strategies may imply into different city structures and organization.

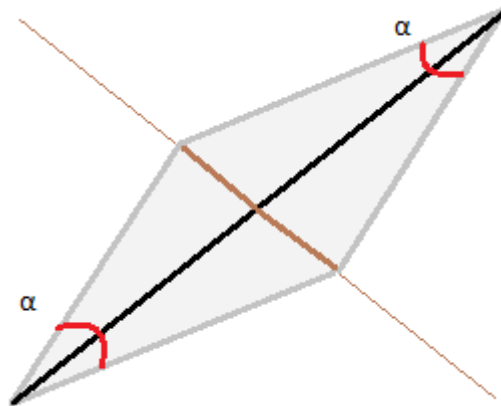


**Figura 10.** *When all regions in map are populated throughout the random rotation algorithm and rectangle fill up, suddenly results a map resembling a real city.*

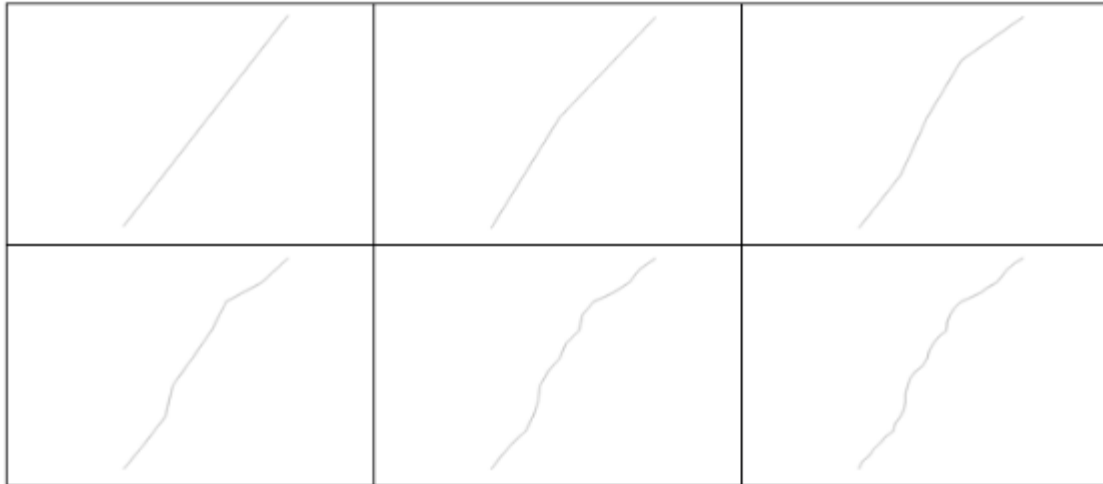
Although the maps resulting from the previous algorithm look alike cities, they are not clearly designed to not be traveled by car. These cities are similar, somehow, to old villages or cities from XVIII century. Nowadays cities require some wide streets,

long enough to divide the city and allow traveling long distances easily. Traveling by car or another long range vehicles. Hence, an additional mechanism was created, in order to generate something like avenues. These artificial avenues should not be straight lines, instead they would be rough broken lines; as they are in real world.

The first step consists in creating a straight line, and then calculate the middle point, after that we move the middle point and randomly create an angle; keeping it in the bisector. Such mechanism is repeated to every resulting line from applying middle point. Just like a fractal generation: same rule recursively. Figure 11 contains an example of this line splitting.



**Figura 11.** Line split considering angle  $\alpha$  for denoting the maximum deviation  
After applying some iterations over the original line, a virtual representation from an avenue has been created. Please, see figure 12 containing the evolution process for an artificial avenue using the described algorithm.



**Figura 12.** Evolution process for the generation of an artificial avenue, by the use of the bisector and randomly angling on lines.

Artificial avenues are integrated into the model and there is an impressive improvement regarding city configuration. The model suddenly acquires an organization which mimics real modern cities. It is possible to verify such aspect in figure 13.

The city generation algorithm has some parameters as input:

- Size of the city
- Size of the grid for the random points
- Number of vertical avenues
- Number of horizontal avenues
- Angle for the roughness of the avenues
- Size of the blocks

These elements can be manage in order to produce different city's topologies.

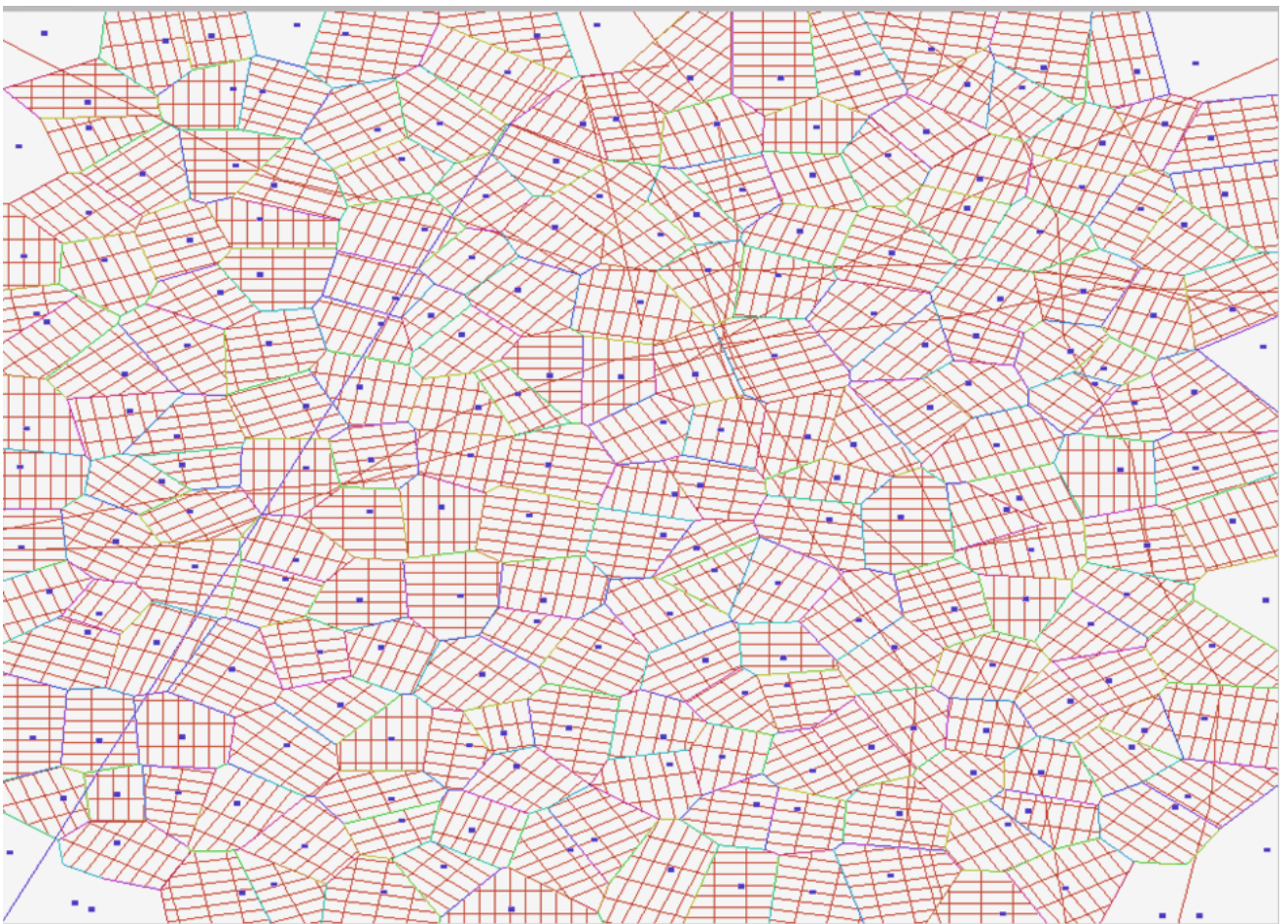
## 5. Putting all together

On the one hand there is an ontological model enabled to represent cities, on the other hand there is a mechanism enabled to produce the geometric representation for a city. The city generation mechanism has been also enabled to produce the ontological representation for the generated city. This simple action enables the



possibility to store the city and recovering it from secondary memory, giving persistence to the whole model.

The main goal of these research effort is to produce synthetic cities, which can be populated with diverse agents from different categories. Current paper focuses on the definition of the city ontological model and the very production of the city through geometrical analysis. Future efforts in this research line will focus in the agents' definition and their insertion in the synthetic city. This is certainly a complex research effort requiring different stages in order to fulfill all requirements.



**Figura 13.** *Through the insertion of avenues into de map, the whole models acquires an organization which mimics that happening in real modern cities.*

## 6. Conclusion

Even though there are massive technological resources, most of city's authorities in world do not trace and record every situation and action happening in the city. In most cases this is caused by the lack of resources, although there are another crucial aspects restricting the pervasive recording of city situations and actions: privacy of citizens, old infrastructure, size of city, number of actors, etc.

Even though there are massive technological resources, most of city's authorities in world do not trace and record every situation and action happening in the city. In most cases this is caused by the lack of resources, although there are another crucial aspects restricting the pervasive recording of city situations and actions: privacy of citizens, old infrastructure, size of city, number of actors, etc.

Current paper presents a crucial stage in our research effort, because it focuses in the creation and representation of synthetic cities, which can be populated with all kinds agents representing citizens, vehicles, semaphores, pollution, rain, etc. and verify their interaction along time. Due to the academic essence of this research effort, we are developing all the tools an elements bottom-up.

According to the reviewed references, our proposal is one of a kind. This last utterance is far from vanity, it only claims that our approach and goals are novel and untreated previously. The ontological approaching is frequently used to match descriptions with patterns. Hence, we are looking for the production of patterns through an ontology. City patterns will be used as molds to produce virtual cities in computers' memory. When virtual cities are produced, they will be populated with diverse agents. Because of the aspects described in this paragraph, we have some constraints to present some testing results.

The Protégé did not manifest any oxymoron regarding the ontological model we produced as a general constructor for virtual cities. The ontology presented in this paper has the potential to model up diverse kinds of cities. Nevertheless, regarding any unconsidered aspect in current ontology; it would be very easy to include additional categories or relations in the ontology.

Future efforts in our research are aiming at the creation of the cyber-agents, as well as their inclusion to the city representation, the latter is focus of this paper.

# References

Casillas, L., Daradoumis, T. & Caballé, S. (2013). A Network Analysis Method for Selecting Personalized Content in e-Learning Programs. *Complex, Intelligent, and Software Intensive Systems (CISIS)*, 2013 Seventh International Conference (pp. 407 - 411). Taichung, Taiwan: IEEE.

Courtat, T., Gloaguen, C., & Douady, S. (2010). Mathematics and Morphogenesis of the City: A Geometrical Approach. arXiv: Electronic archive for physics, mathematics, computer science, nonlinear sciences, quantitative biology and statistics. Cornell University Library.

Davies, J., Duke, A. & Sure, Y. (2004). OntoShare-An Ontology-based Knowledge Sharing System for virtual Communities of Practice. *J. UCS*, 10(3), 262-283.

Döllner, J. & Hagedorn, B. (2007). Integrating urban GIS, CAD, and BIM data by servicebased virtual 3D city models. R. e. al. (Ed.), *Urban and Regional Data Management-Annual*, 157-160.

Fortune, S. (1987). A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1-4), 153-174.

Gandy, M. (2005). Cyborg urbanization: complexity and monstrosity in the contemporary city. *International journal of urban and regional research*, 29(1), 26-49.

Ge, Y., Meng, R., Cao, Z., Qiu, X., & Huang, K. (2014). Virtual city: An individual-based digital environment for human mobility and interactive behavior. *Simulation* 90(8), 917-935.

Gruber, T. (1993). A Translation Approach to Portable Ontologies. *Knowledge acquisition*, v. 5, n. 2, 199-220.

Parish, Y., & Müller, P. (2001). Procedural modeling of cities. *SIGGRAPH '01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (pp. 301-308). New York, NY, USA: ACM.



Videira-Lopes, C., & Lindstrom, C. (2012). Virtual Cities in Urban Planning: The Uppsala Case Study. *Journal of theoretical and applied electronic commerce research*, 7(3), 88-100.

Zhai, J., Wang, Q., & Lv, M. (2008). Application of XML Topic Maps to knowledge navigation and information retrieval for urban traffic information portal. *Control Conference, 2008. CCC 2008. 27th Chinese (pp. 458 - 462)*. Kunming: IEEE.

## Notas biográficas:



**Dr. Luis Casillas** holds a Ph.D. and a Master's Degree in Information and Knowledge Societies, as well as a Master's Degree in Information Systems and a B.Sc. in Informatics. He has been working as full-time professor for more than 20 years in the Computer Science Department from the University of Guadalajara (Mexico). He has published various papers and scientific chapters in diverse journals and books. He serves as member of the editorial board for a couple of journals, and reviewer for diverse journals and other scientific publications about: knowledge engineering, computer science and ICT in education. His research interests are: knowledge gathering and representation, bio-inspired systems, expert systems, complex networks analysis, and soft-computing.



**Gilberto Vargas** En el 2011 inicié mi camino como estudiante de las ciencias computacionales participando en la olimpiada estatal de informática. Logré el primer lugar a nivel estatal y calificar a la fase nacional de la olimpiada, que se llevo a cabo en la ciudad de Cuernavaca, Morelos. Ahí obtuve una medalla de plata. En el 2012 volví a la olimpiada estatal pero esta vez como organizador. Hasta la fecha, mi función en la olimpiada es proponer problemas para los exámenes y entrenar a los mejores chicos que desean ir a la fase nacional. Ese mismo año ingresé a la licenciatura en la Universidad de Guadalajara y en Diciembre del 2015 egresé. Participo en el ICPC de la ACM desde el 2012 hasta el 2015 logrando calificar a la fase regional

en todos los años. También junto a mi equipo, logramos el tricampeonato en el concurso de programación organizado por la ANIEI (2013-2015). En el 2013 realice una estancia de verano en el Cidetec de la UVM campus Guadalajara Sur, trabaje en un proyecto de tráfico urbano.



**Miguel Lares** Estudiante de Ingeniería en Computación en la Universidad de Guadalajara. A participado en varios concursos de computación, ganando más recientemente el primer lugar en CU Valles. Además a concursado en varios eventos como Codico Ciencia y GuadalajaraCON, también cooperó en la organización de Install Fest y eventos varios en CUCEI.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Electrónica

*ReCIBE, Año 5 No. 2, Mayo 2016*

## **Reconocimiento de micro partículas de polen con algoritmos de procesamiento de imágenes implementados en dispositivos reconfigurables**

Juan Manuel Ruiz Varela  
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional,  
[jmruiz@gdl.cinvestav.mx](mailto:jmruiz@gdl.cinvestav.mx)

Susana Ortega Cisneros  
Centro de Investigación y de Estudios Avanzados del Instituto  
Politécnico Nacional,  
[sortega@gdl.cinvestav.mx](mailto:sortega@gdl.cinvestav.mx)

Adrian Pedroza de la Cruz  
Centro de Investigación y de Estudios Avanzados del Instituto  
Politécnico Nacional,  
[apedroza@gdl.cinvestav.mx](mailto:apedroza@gdl.cinvestav.mx)

Juan José Raygoza Panduro  
Centro de Ciencias Exactas e Ingeniería "CUCEI",  
Universidad de Guadalajara  
[juan.raygoza@cucei.udg.mx](mailto:juan.raygoza@cucei.udg.mx)

Jorge Rivera Domínguez  
Centro de Investigación y de Estudios Avanzados del Instituto  
Politécnico Nacional  
[jrivera@gdl.cinvestav.mx](mailto:jrivera@gdl.cinvestav.mx)

**Resumen:** En este artículo se presenta una propuesta de un dispositivo capaz de identificar y enumerar las diversas familias de esporas de polen existentes a partir de una muestra obtenida. Este dispositivo cuenta con la implementación de dos algoritmos, el primero se enfoca al procesamiento de imágenes y el segundo a la extracción de firmas de las imágenes ya procesadas, de ésta forma es posible obtener la identificación y clasificación de cada una de las esporas de polen en la muestra, para así posteriormente hacer el conteo. Este proyecto se lleva a cabo utilizando una videocámara Costar SI-C400N adaptada a un microscopio, la tarjeta

de desarrollo DE2i Altera, basada en un dispositivo Cyclone IV FPGA, memorias embebidas y un Softcore NIOS II. Esto será de gran utilidad en el área de la medicina, siendo más objetivos en el área de la alergología e inspección. Para la realización del proyecto se desarrollan las siguientes etapas, la primera está basada en la extracción de la imagen, esto se desarrolla con la ayuda de una videocámara montada en un microscopio con el cuál obtendremos el video de la muestra de esporas de polen a tratar. Se utiliza una videocámara la cual proporciona a su salida una señal analógica, dicha señal es introducida en la tarjeta para desarrollar el procesamiento adecuado. Al adquirir el video, éste es almacenado en la memoria SDRAM, la siguiente etapa se basa principalmente en leer el video almacenado y procesarlo correctamente con la ayuda del Softcore NIOS II, el cual tiene los algoritmos programados con lenguaje de alto nivel para hacer el procesamiento adecuado sobre el video entrante. En la tercera etapa, se guarda el video ya procesado en la memoria SRAM, para finalmente ser mostrado mediante el puerto VGA.

**Palabras clave:** FPGA, Softcore NIOS II, memorias embebidas, Procesador, procesamiento de imágenes, sistemas embebidos, microscope, SDRAM memory, SRAM memory, VGA port.

## **Recognition of pollen micro particles with image processing algorithms implemented in reconfigurable devices**

**Abstract:** This paper presents a proposal of a device capable of identifying and listing the various families of existing pollen spores from a sample obtained is presented . This device has the implementation of two algorithms, the first focuses the image processing and the second to the extraction of signatures of images already processed , this way it is possible to obtain the identification and classification of each pollen spores in the sample , so then do the counting. This project is carried out using a camcorder Costar SI- C400N adapted to a microscope, DE2i Altera development board , based on a Cyclone IV FPGA device , embedded memories and Nios II Softcore . This will be useful in the area of medicine , being more goals in the area of allergy and inspection. For the project the following stages are developed , first is

based on the extraction of the image, it develops with a camcorder mounted on a microscope what we get the video of sample pollen spores. Using a camcorder which provides an output analog signal, this signal is introduced into the card to develop appropriate processing. By acquiring the video, it is stored in the SDRAM memory, the next stage is mainly based on reading the stored video and process it correctly with the help of Softcore NIOS II, which has programming algorithms with high-level language for processing right on the incoming video. In the third stage, video already processed is stored in the SRAM memory, to finally be shown through the VGA port.

**Keywords:** FPGA, Softcore NIOS II, embedded memories, Processor, image processing, embedded systems, microscopio, memoria SDRAM, memoria SRAM, puerto VGA.

# 1. Introducción

El estudio de clasificación y conteo de esporas de polen que afectan todas las regiones del mundo se desarrolla de forma manual, lo que resulta en una tarea muy agobiante y demorada para el investigador, por lo que con éste proyecto, el operador podrá desarrollar ésta labor de forma automática, invirtiendo menor cantidad de tiempo. Se considera que también será de impacto en el área de diagnóstico médico, debido a que con esta clasificación se podrá diagnosticar que esporas, en específico y en que concentración, son portadas por un paciente el cuál sufra de intoxicación por esporas de polen, haciendo más fácil la intervención médica, y se pueda suministrar el tratamiento adecuado de forma más rápida y exacta.

El procesamiento de imágenes ha sido muy utilizado en diversas aplicaciones; tales son el sector industrial, comercial, civil y militar durante más de un par de décadas. La detección de bordes es siempre el primer paso a seguir en muchos algoritmos de procesamiento de imágenes, ya que hace una reducción en la cantidad de datos y filtra toda información inútil. Con el avance en ésta disciplina se han propuesto gran cantidad de detectores de bordes, como el detector de Roberts, el detector de Prewitt, detectores Laplacianos, el detector de Sobel y el detector de Canny, entre otros. De los algoritmos anteriores, el de Canny es el mejor detector de bordes en casi todas las situaciones (Raman, M. et al., 2009).

Hoy en día, con el rápido desarrollo de la tecnología de la imagen, las resoluciones se han multiplicado en los últimos años, por esto se propone una plataforma de hardware novedosa en FPGA, debido a su capacidad de procesamiento paralelo y características reconfigurables, además, se ha convertido en una plataforma paralela útil para procesamiento de imágenes. En (Sowmya,S. et al.,2011), se propone una arquitectura de hardware basada en FPGA para la mejora de la imagen, en (Wenhao, H. et al.,2008), se propone el algoritmo de Canny, igualmente basado en FPGA.

Hoy en día, con el rápido desarrollo de la tecnología de la imagen, las resoluciones se han multiplicado en los últimos años, por esto se propone una plataforma de hardware novedosa en FPGA, debido a su capacidad de procesamiento paralelo y características reconfigurables, además, se ha convertido en una plataforma paralela útil para procesamiento de imágenes. En (Sowmya,S. et al.,2011), se propone una arquitectura de hardware basada en FPGA para la mejora de la imagen, en (Wenhao, H. et al.,2008), se propone el algoritmo de Canny, igualmente basado en FPGA.

## 2. Procesamiento digital de imagenes

Actualmente el procesamiento digital de imágenes es una herramienta muy específica en computación, se puede definir como las técnicas que se aplican a las imágenes digitales. Este procedimiento se utiliza en diversas áreas importantes (Osio, J. et al., 2011):

- Mejorar la calidad de la información de una imagen.
- Procesamiento de los datos de una imagen de forma autónoma.
- Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.
- Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- Realzar bordes: destacar los bordes que se localizan en una imagen.

- Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.

Para el procesamiento es necesario tratar a la imagen como una matriz de valores numéricos, cada elemento de la matriz representa un pixel. Los valores de cada pixel se encuentran entre 0 y 255, donde el 0 corresponde al negro que es el tono más oscuro y el 255 corresponde al blanco que es el tono más claro, el proceso de filtrado se puede desarrollar tanto en el dominio de la frecuencia como en el dominio del espacio (Rodriguez, R. et al., 2012).

### 3. Detección de bordes

Los bordes de una imagen son las secciones en donde ocurre un cambio brusco de intensidad, es decir, en donde existe una transición entre dos partes que tienen diferentes niveles de gris, el resultado de este proceso sirve como entrada para tareas adicionales como reconocer objetos, compresión de imágenes, registro y alineación (Rajbahadur, S. et al., 2013).

Hasta el momento se cuenta con varios algoritmos de detección de bordes, entre los más destacados están:

- Algoritmo de Sobel.
- Algoritmo de Prewitt.
- Algoritmo Laplaciano.
- Algoritmo de Roberts.
- Algoritmo de Canny.

El último de los antes mencionados se eligió para la implementación, debido a la eficiencia en el tratamiento de la imagen que mostró en (Raman, M. et al., 2009) y también por ser el adecuado para este tipo de aplicación.

### 4. Algoritmo de Canny

El algoritmo de Canny es un operador desarrollado por John F. Canny en 1986 que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de

bordes en imágenes. Lo más importante es que Canny también desarrolló una teoría computacional acerca de la detección de bordes que explica por qué la técnica funciona (Gonzalez, R. et al., 1992).

#### *A. Desarrollo del algoritmo de Canny*

El propósito de Canny era descubrir el algoritmo óptimo de detección de bordes. Para que un detector de bordes pueda ser considerado óptimo debe cumplir los siguientes puntos:

*Buena detección* - el algoritmo debe marcar el mayor número real en los bordes de la imagen como sea posible.

*Buena localización* - los bordes de marca deben estar lo más cerca posible del borde de la imagen real.

*Respuesta mínima* - El borde de una imagen sólo debe ser marcado una vez, y siempre que sea posible, el ruido de la imagen no debe crear falsos bordes.

Para satisfacer estos requisitos se utiliza el cálculo de variaciones, una técnica que encuentra la función óptima para un funcional indicado. En el algoritmo de Canny, esta función se describe por la suma de cuatro términos exponenciales, pero se puede aproximar por la primera derivada de una gaussiana.

Entre los métodos de detección de bordes desarrollado hasta el momento, el algoritmo de detección de bordes es uno de los métodos más estrictamente definido que proporciona la detección buena y confiable. Debido a su optimización para cumplir con los tres criterios para la detección de bordes y la simplicidad del proceso de implementación, se convierte en uno de los algoritmos más populares para la detección (Xiaoyang, L. et al., 2012).

#### *B. Etapas del algoritmo de Canny*

El proceso del algoritmo de detección de bordes de Canny puede ser descrito en 5 etapas diferentes:

- 1. Aplicar el filtro gaussiano para suavizar la imagen con el fin de eliminar el ruido.
- 2. Encontrar los gradientes de intensidad de la imagen.
- 3. Aplicar no supresión máxima para deshacerse de la respuesta espuria de detección de bordes.
- 4. Aplicar doble umbral para determinar los bordes potenciales.



- 5. Seguimiento de borde por histéresis: Finalizar la detección de bordes mediante la supresión de todos los otros bordes que son débiles y no están conectados a los bordes fuertes (Acharya, T. et al., 2005).

## 5. Extracción de las firmas y Clasificación

Para hacer el desarrollo de éste punto se hará uso de descriptores, los cuales representan las características elementales en un objeto, además de realizar una adecuada discriminación, ser suficientemente fiables y lo más independiente posible. Existen descriptores de contornos y de regiones para el reconocimiento e identificación de objetos en una imagen. Además de un conjunto de operadores morfológicos que permiten la manipulación de la forma de los objetos, extracción de contorno, y clasificación de objetos según su tamaño (Gonzalez, R. et al., 1992).

Los esquemas de representación de forma deben tener ciertas propiedades deseables:

*Unicidad:* Cada objeto debe tener una única representación.

*Invariancia* frente a transformaciones geométricas, como traslaciones, rotaciones, cambios de escala y reflexiones.

*Sensibilidad* a capacidad de diferenciar objetos casi iguales.

*Abstracción* del detalle o capacidad para representar los rasgos característicos básicos de los objetos y abstraer los detalles.

## 6. Diseño e implementación

Para la elaboración del proyecto, se presenta la metodología de forma gráfica (Fig. 1), la cual posteriormente se explica detalladamente (Xiaoyang, L. et al., 2012).



**Figura 1.** Diseño de algoritmo.

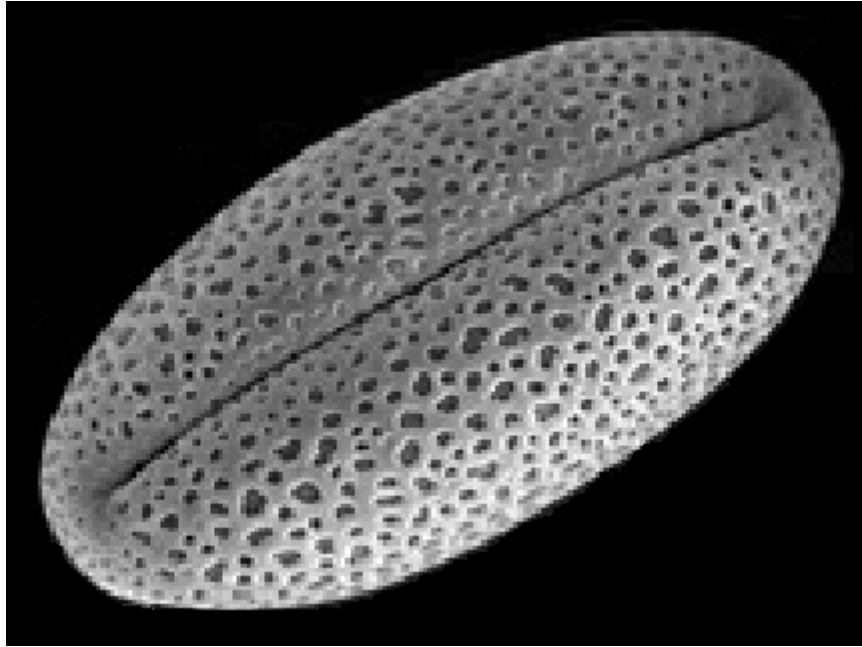
*A. Obtención de la imagen*

Para obtener la imagen a procesar, primeramente se pondrá una muestra con polen en el microscopio óptico, y mediante la videocámara se obtendrá una imagen a procesar, la cuál será enviada a la tarjeta por medio del jack de video.

*B. Almacenamiento en SDRAM*

Una vez obtenidas las imágenes son almacenadas en la memoria SDRAM, aquí son almacenadas en escala de grises, asignando un valor correspondiente a cada pixel de la matriz que esta entre 0 y 255, y con formato de imagen bmp (Archivo de mapa de bits) (Woods, R. et al., 2009).

En la Fig. 2 se muestra la espora de polen que se considera para el desarrollo de esta prueba.



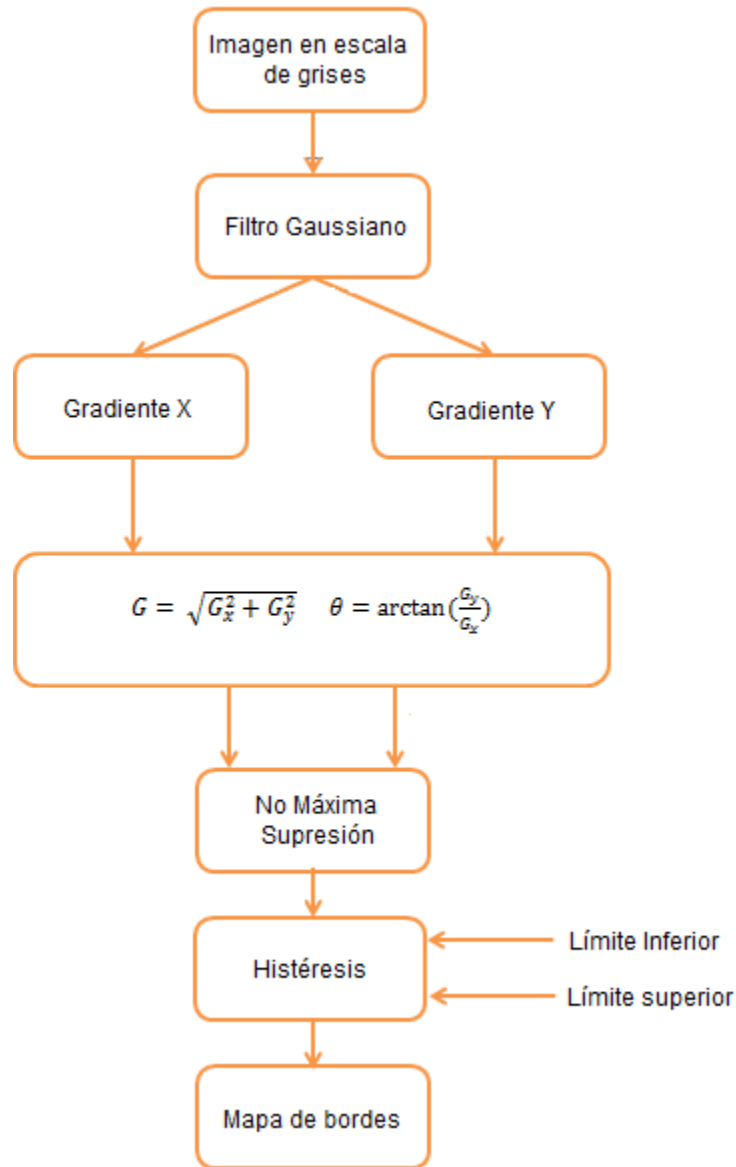
**Figura 2.** Espora de polen *Lilium auratum* a procesarse en escala de grises.

C. *Detección de bordes*

Una vez almacenadas las imágenes en la memoria SDRAM, el Softcore embebido procede con la rutina de lectura a la memoria SDRAM, para extraer la información de la imagen y aplicar el algoritmo de detección de bordes.

Mediante el detector de bordes de Canny, se aplica el filtro Gaussiano para eliminar todo el ruido y obtener un suavizado de la imagen.

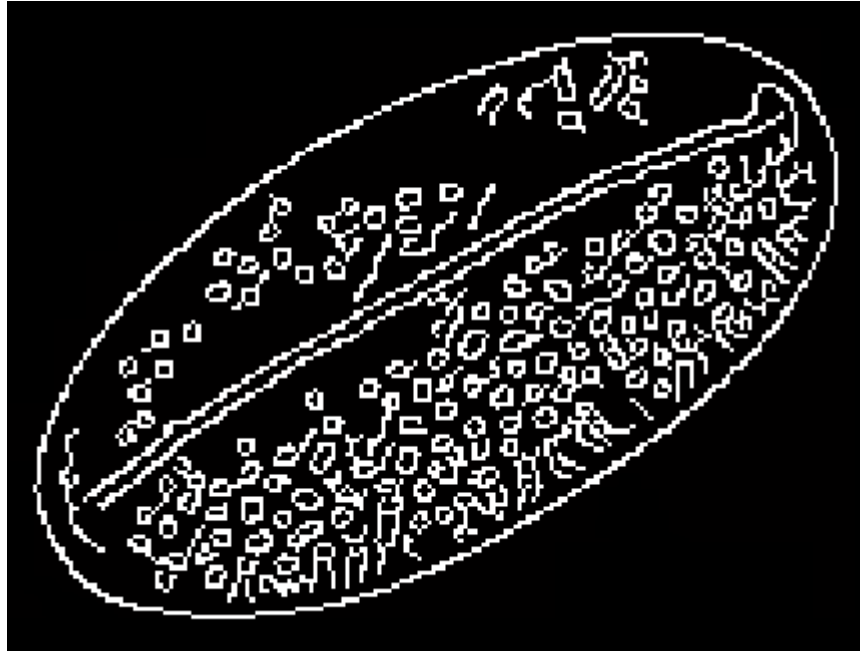
El objetivo del suavizado es el de aumentar o disminuir ciertas características que se encuentran presentes en la imagen, el filtro que hemos usado está basado en la convolución de la media aritmética que tiene como fin reducir la presencia de ruido, es decir, eliminar el componente de altas frecuencias y mantener las bajas Fig. 3.



**Figura 3.** Diagrama de flujo del detector de bordes Canny.

Diagrama de flujo del detector de bordes Canny.

Ésta imagen será la entregada por el algoritmo de detección de bordes que está implementado en el Softcore NIOS II, para posteriormente desarrollar las operaciones correspondientes a los pasos siguientes.



**Figura 4.** Imagen procesada por el detector de bordes Canny.

*D. Firma y clasificación*

Una vez obtenida la imagen procesada del detector de bordes de Canny, se procede a la extracción de la firma de la imagen y clasificación.

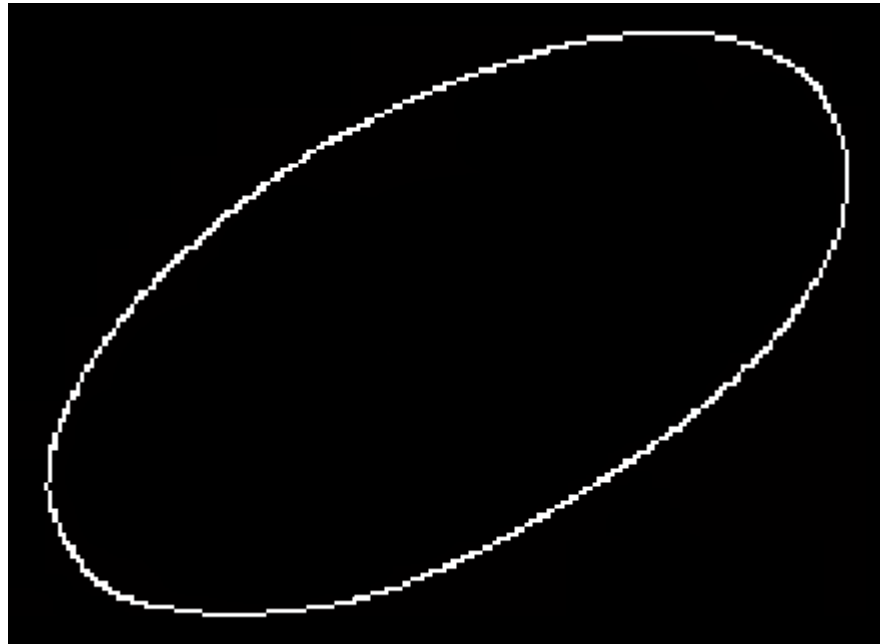
Por el tipo de objeto se seleccionaron dos clases de descriptores; morfológicos y de textura. Dentro de los descriptores morfológicos se encuentra el área, el perímetro, la circularidad y la firma. Y como descriptor de textura se consideró el histograma de cada partícula de polen y la densidad superficial que combina la distribución del nivel de gris y el tamaño mediante el cociente entre la suma de las intensidades de todos los píxeles pertenecientes al objeto y el área del mismo (Morales, J. et al., 2012).

Por el tipo de objeto se seleccionaron dos clases de descriptores; morfológicos y de textura. Dentro de los descriptores morfológicos se encuentra el área, el perímetro, la circularidad y la firma. Y como descriptor de textura se consideró el histograma de cada partícula de polen y la densidad superficial que combina la distribución del nivel de gris y el tamaño mediante el cociente entre la suma de las intensidades de todos los píxeles pertenecientes al objeto y el área del mismo (Morales, J. et al., 2012).

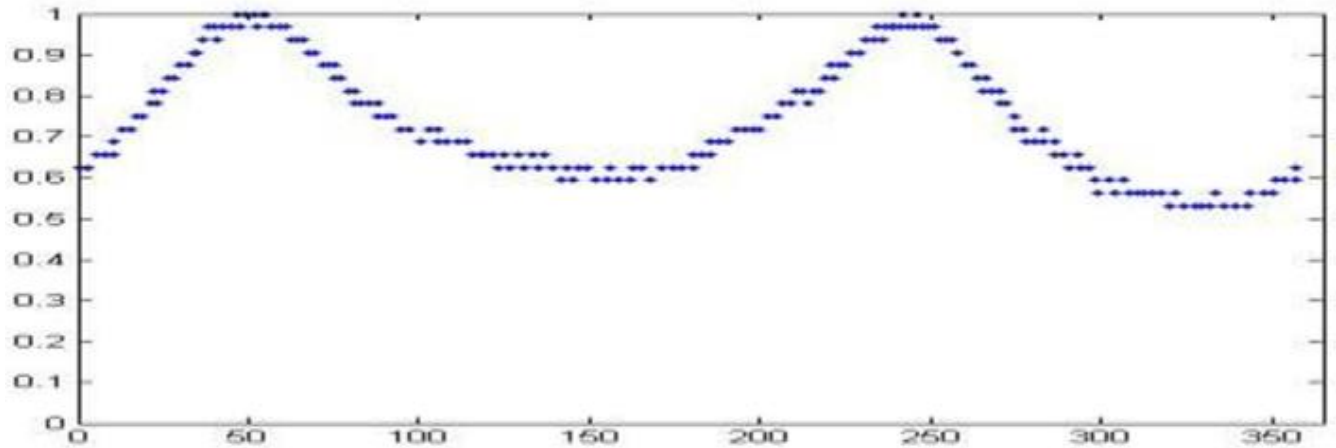
Con la imagen de la Fig. 4 procedemos a desarrollar la extracción de la forma, para así poder desarrollar después la extracción de la firma y hacer una base de datos a donde serán almacenadas las firmas en grupos de cada una de las esporas de polen, cabe aclarar que existen varias firmas de cada espora de polen para evitar errores en la clasificación y conteo.

A la imagen de la Fig. 4 se le aplica un filtrado de pixeles, el cual excluye todo pixel dentro de la espora, dando como resultado la Fig. 5, haciendo esto para reducir el procesado de datos al momento de extraer la firma y evitar también errores durante el procesado.

Con base a lo mencionado anteriormente, se procede a extraer la firma a partir de la forma obtenida, se utiliza el histograma para mostrar de forma gráfica la firma de cada una de las esporas de polen, y así albergar todas las muestras en la base de datos que se encarga de la clasificación y conteo de las partículas. Partiendo de lo antes mencionado, obtenemos una firma para la espora de muestra, la cual se puede observar en la Fig. 6 (Rocha, A. et al., 2010).



**Figura 5.** Imagen procesada después de aplicar la exclusión de pixeles.



**Figura 6.** Firma correspondiente a la espora de muestra.

*E. Almacenamiento en SRAM*

Una vez desarrolladas las etapas anteriores, el Softcore embebido es el encargado de enviar los datos a sus respectivas direcciones, como lo es el caso de la imagen ya procesada por el detector de Canny, la cual será guardada en la SRAM.

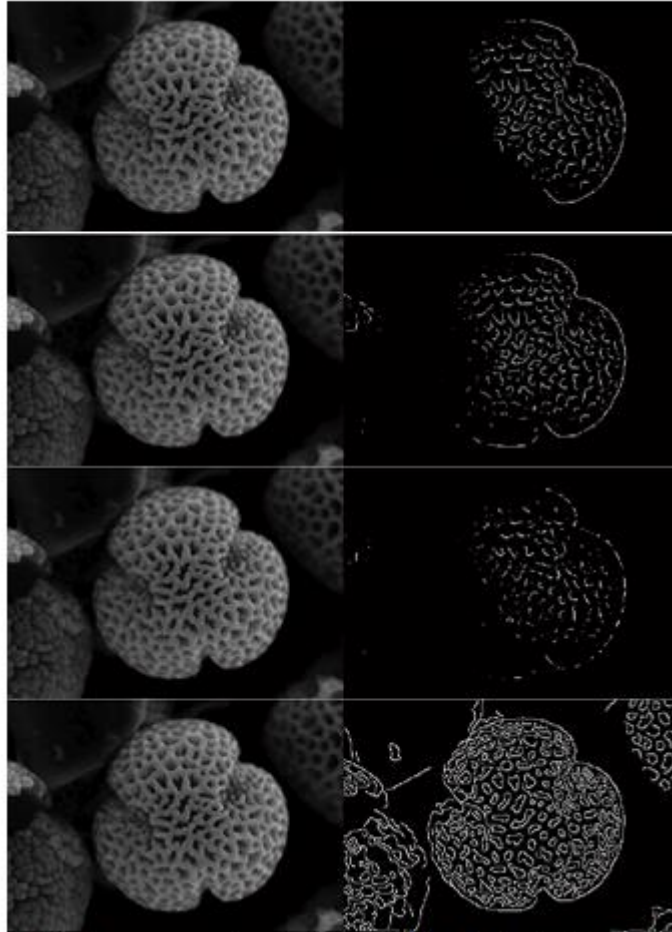
*F. Resultados en monitor VGA*

Posteriormente se leen los datos de la memoria SRAM para enviarse al Puerto VGA y de esta manera procesarse para mostrarlos en un monitor, con el fin de que el operador pueda analizar de forma gráfica el tipo de espora, con la cual un paciente está intoxicado, y de esta forma suministre el tratamiento adecuado. En el caso de estar realizando labores de investigación, podrá analizar más a detalle cada una de las esporas, también tiene visible el conteo y la clasificación de cada una de las esporas existentes en la muestra.

## 7. Resultados

Como primer paso en esta metodología de diseño, se prueban los algoritmos a implementar en el software MatLab, esto con el motivo de probar su veracidad, y obtener buenos resultados, los cuáles servirán posteriormente como base para realizar las comparaciones pertinentes con los resultados obtenidos en hardware. En la Fig. 7 se procede a mostrar los resultados de estas implementaciones

(Cuevas, E. et al., 2010). Tras haber realizado la implementación de los diferentes detectores de bordes, como se muestra en la Fig. 6, se presentan los resultados, analizando el desempeño y veracidad de cada uno.



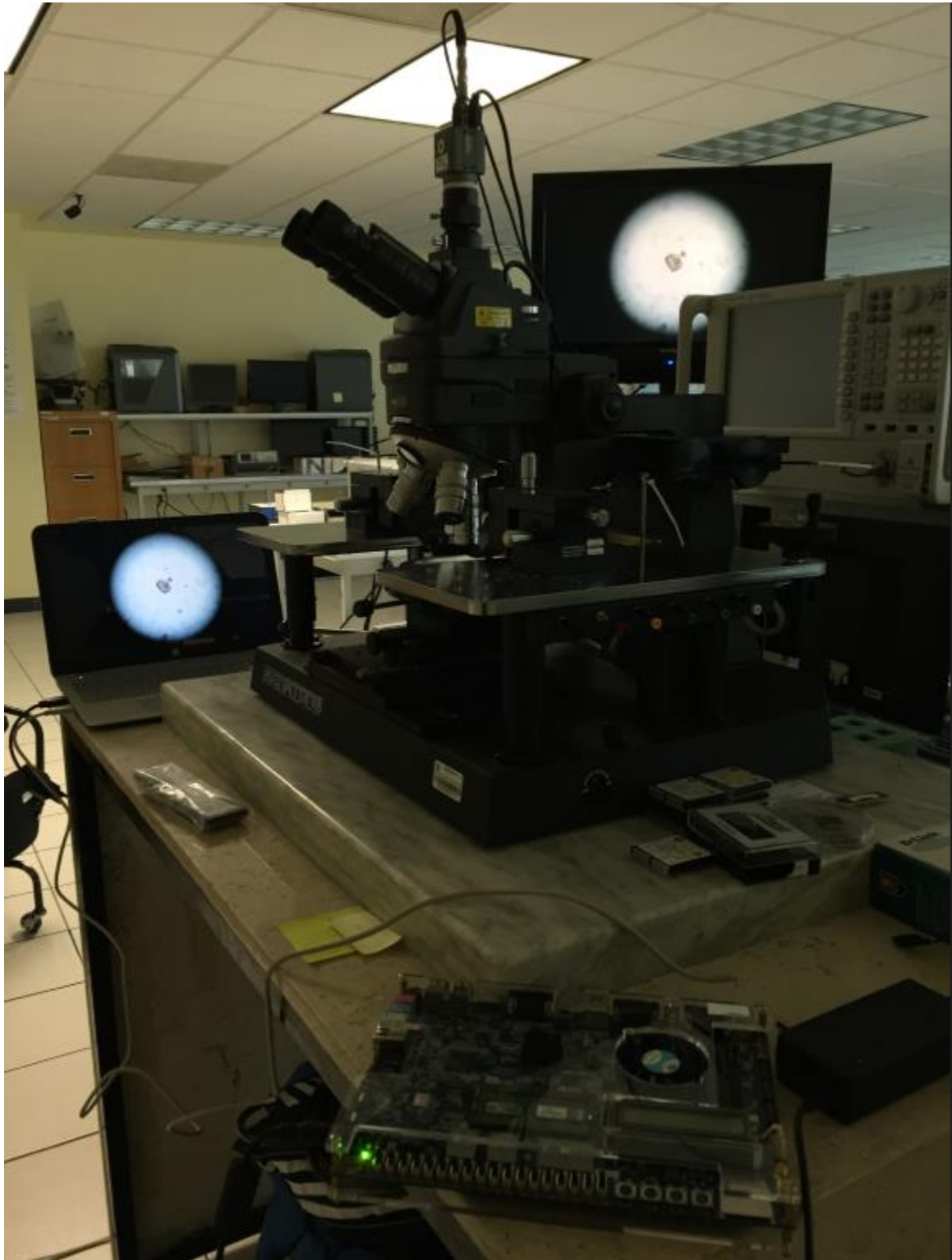
**Figura 7.** Imagen original a la izquierda, imagen procesada mediante detector de bordes a la derecha, primeramente se observa el detector de bordes de Prewitt, posteriormente el de Sobel, después el de Roberts y finalmente el de Canny.

Primeramente tenemos que para el primer detector de bordes, que es el de Prewitt, hay pérdida de información en la parte de las orillas de la imagen, ya que existen diversas esporas alrededor y no las detecta. El segundo detector de bordes, que es el de Sobel, presenta menor pérdida de información, sin embargo no alcanza a detectar totalmente las esporas que tiene en sus bordes. Posteriormente se presenta el detector e bordes de Roberts, el cual presenta pérdida de información pero menor en comparación con el de Prewitt. Finalmente



se presenta el detector de bordes de Canny, el cual presenta mayor desempeño ya que tiene menor pérdida de datos y muestra únicamente los bordes de la imagen que son de utilidad y los realza, lo demás que el filtro considera despreciable, simplemente es desechado, como se muestra, rescata la información de las esporas que tiene a su alrededor, además de brindar una buena reconstrucción de la espora central.

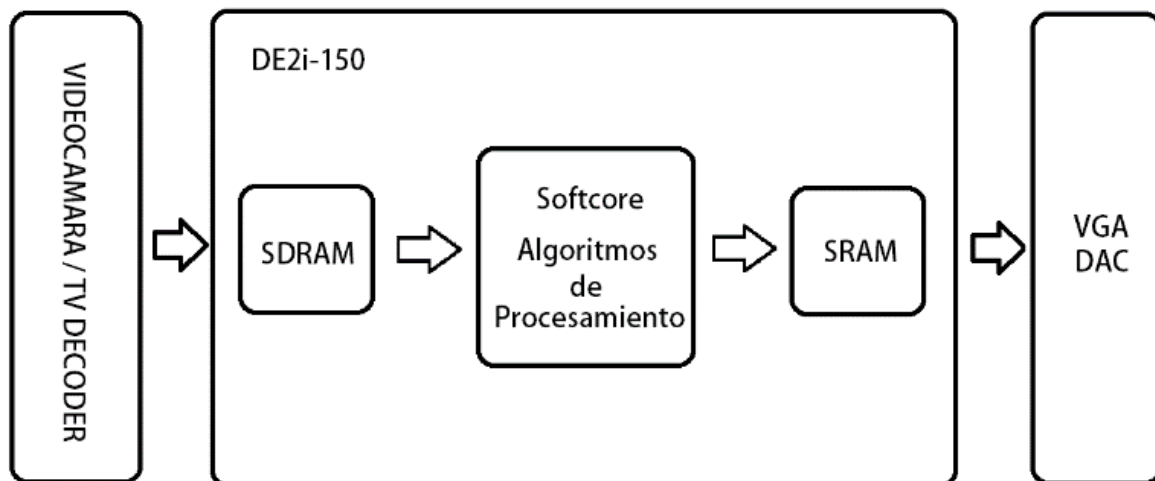
Por ello es que se implementa dicho detector de bordes, esto con el fin de alcanzar mejores resultados desde el inicio del procesado, recordando que ésta etapa de detección es la inicial, y al obtener los mejores resultados desde un inicio se optimizarán los procesos siguientes, como son: la exclusión de píxeles, firma, numeración y clasificación, obteniendo así una serie de resultados favorables, que le serán de gran utilidad al usuario. Por otro lado se desarrolla una estimación de tiempos, y se concluye que gracias al procesamiento paralelo de la FPGA, el tiempo de respuesta deberá ser menor con base al del software, ya que gracias al Softcore utilizado como bloque principal de control, nos queda un reducido bloque de hardware que a su vez tienen la función de ser bloques aceleradores de procesos, debido a esto se hacen reducciones de tiempos considerables, desarrollando de ésta manera todas las tareas que conforman el proyecto.



*Figura 8. Estación de pruebas analizando una espora de polen.*

En la Fig. 8 se muestra la estación de pruebas en donde se realizan las labores de visualización de las esporas de polen, extracción de imagen de las mismas y procesamiento de datos tanto en el dispositivo reconfigurable como en la PC. Se realizan las tareas de visualización y extracción de la imagen, siendo esta misma procesada para posteriormente almacenarse en el FPGA.

A continuación, en la Fig. 9 se muestra un diagrama de la arquitectura del sistema implementado.



**Figura 9.** Implementación del Sistema.

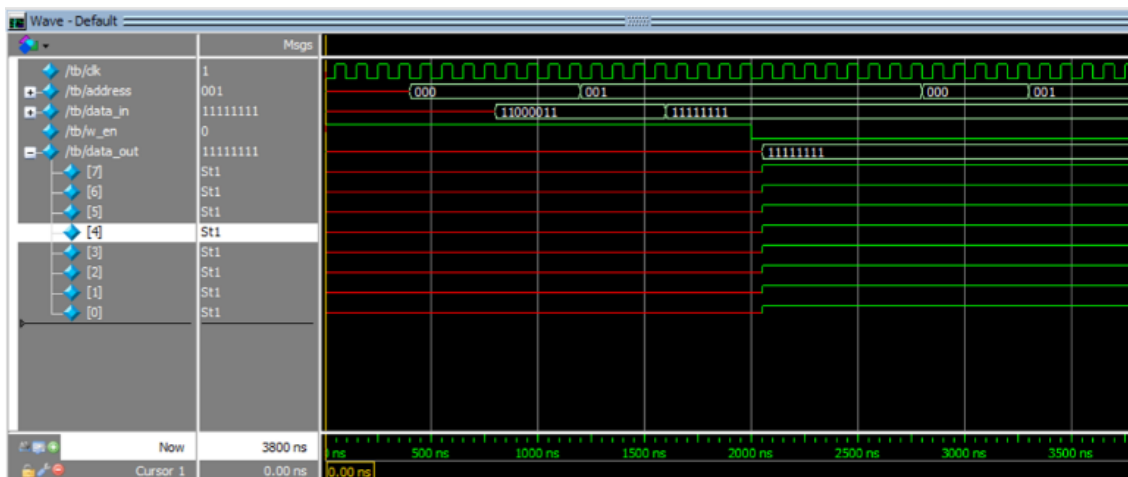
En la Fig. 9 se describe la operación del sistema de forma detallada, analizando la entrada de los datos a partir de la videocámara, posteriormente su procesamiento en la tarjeta de desarrollo y el despliegado de datos en el monitor VGA en forma de imagen.

La síntesis del sistema se realizó mediante la herramienta del software Quartus II y se obtuvo el resultado de ocupación, el cual se muestra en la Tabla 1.

<b>Elementos utilizados</b>	<b>Ocupación</b>	<b>Porcentaje</b>
Elementos lógicos	4,460 / 33,216	13 %
F. combinacionales	4,213 / 33,216	13%
Registros lógicos	2,904 / 33,216	9%
Registros totales	3000	-
Pines totales	94/475	20%
Bits de memoria	242,712 / 483,840	50%
PLLs totales	1 / 4	25%

**Tabla 1.** Resultados de ocupación de área para el sistema desarrollado.

Finalmente se muestra en la Fig. 10 el comportamiento de la primera parte del sistema, la cual consiste en la escritura de los datos adquiridos por la videocámara en la memoria SDRAM, y su salida hacia el Softcore, para posteriormente procesarse.



**Figura 10.** Comportamiento en tiempo de la escritura y lectura de datos en la memoria SDRAM incorporada en el sistema.

Se puede observar con detalle que a los 2000 ns se inicializan y sincronizan las señales de lectura y escritura de la memoria SDRAM a utilizar.

## 8. Conclusión

En este trabajo, se propone una metodología en hardware para la detección de micro partículas, que está compuesta de una arquitectura de hardware a la medida para la detección de bordes en imágenes, así como también de un diseño clasificador de micro partículas de polen, junto con un bloque que realiza las tareas de conteo de las mismas, todo esto se ha realizado con base en una FPGA. Mediante el uso del detector de Canny que es un filtro para mejorar la mediana de la imagen, el rendimiento se mejora significativamente cuando la imagen es liberada de la contaminación por ruido. La velocidad de procesamiento se mejorará y el consumo de hardware se reduce. El diseño se realiza en un Cyclone IV FPGA de Altera, los resultados experimentales y el consumo de tiempo indican que esta arquitectura de hardware es suficiente para aplicaciones que incorporen procesamiento de imagen.

## RECONOCIMIENTO

Los autores agradecen el apoyo financiero proporcionado por el CONACYT (Consejo Nacional de Ciencia y Tecnología) como una beca de maestría.

## Referencias

Acharya, Tinku., Ray, A. K. (2005). Image processing: principles and applications. John Wiley & Sons

Cuevas, Eric., Zaldívar, D., Pérez, M. (2010) "Procesamiento Digital De Imágenes Con Matlab Y Simulink" Alfaomega-Ra-Ma.

Gonzalez, Rafael., Woods, R., Eddins, S. (1992) "Digital Image Processing". Addison-Wesley.

Jorge Osio; Walter Aróztegui; Jose Rapallini; Antonio Adrián Quijano; Jesús Ocampo (2011) "Desarrollo de Algoritmos de Procesamiento de Imágenes Basados en Operadores de Ventana sobre una FPGA." Primeras Jornadas de Investigación y Transferencia, UNLP.

Morales, G.J. (2012) "Diseño de un sistema de reconocimiento y cuantificación de partículas de polen con dispositivos reconfigurables." Tesis de Doctorado Universidad de Guadalajara.

Rajbahadur Singh Raghuvanshi , Ashutosh Datar (2013) "Comparison of Gradient Operator based Pseudocolored Enhanced Medical Images." ; Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)

Raman Maini, Himanshu Aggarwal, (2009) "Study and comparison of various image edge detection techniques," Int. Journal of Image Processing.

Rocha, P.A. (2010) "Diseño de un sistema embebido basado en un microprocesador de núcleo flexible con dispositivos reconfigurables." Tesis de Doctorado Universidad de Guadalajara.

Rodriguez, Roberto., Sossa, Juan. (2012) "Procesamiento digital de imágenes" Alfaomega-Ra-Ma.

S. Sowmya, R. Paily. (2011) "FPGA implementation of image enhancement algorithms," Int. Conf. Communications and Signal processing.

Wenhao He, Kui Yuan. (2008) "An improved Canny edge detector and its realization on FPGA," World Congress on Intelligent Control and Automation.

Woods, Richard., Eddins, S., Gonzalez, R. (2009). "Digital image processing using Matlab." Pearson Education.

Xiaoyang Li, Jie Jiang, Qiaoyun Fan (2012) "An Improved Real-time Hardware Architecture for Canny Edge Detection Based FPGA", Third International Conference on Intelligent Control and Information Processing.



**Juan R. Varela** recibió el grado de Licenciatura en Ingeniería en Comunicaciones y Electrónica en el Instituto Tecnológico de Oaxaca, Oaxaca, México. Es un estudiante de Maestría en Diseño Electrónico en el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional Unidad Guadalajara, Jalisco, México. Su área de investigación es la de procesamiento digital de imagen.



**Susana Ortega Cisneros** Recibió su grado de licenciado en ciencias en Comunicaciones y Electrónica de la Universidad de Guadalajara, México en 1990, el grado de maestría del Centro de Investigación y de Estudios Avanzados del IPN, Zacatenco, México. Susana Ortega recibió su grado de doctor en Ciencias de la Computación y Telecomunicaciones de la Universidad Autónoma de Madrid, España. Se especializa en el diseño digital de arquitecturas basadas en FPGAs, DSPs y microprocesadores. Las principales líneas de investigación en las cuales trabaja son: Control Digital, Sincronización Self-Timed, Sistemas Electrónicos Aplicados a la Biomedicina, Diseño de Microprocesadores Embebidos, electrónica digital y DSPs personalizados en FPGAs.



**Juan José Raygoza** recibió el grado de Licenciado en Ingeniería en Comunicaciones y Electrónica en la Universidad de Guadalajara, México en 1989, el grado de Maestro en el Centro de Investigación y de Estudios Avanzados del IPN; Zacatenco, México. Juan José

Raygoza recibió el grado de Doctor en Ciencias de la Computación y Telecomunicaciones de la Universidad Autónoma de Madrid, España. De 1996 al 2000, trabajó en IBM, participó en la transferencia tecnológica de la fabricación de cabezales de discos duros, en la planta de manufactura de IBM en San José C.A a Guadalajara, México. Se especializa en el diseño de arquitecturas digitales basadas en FPGAs, Microprocesadores, sistemas embebidos y bioelectrónica. Las principales líneas de investigación en las que trabaja son: Sistemas Electrónicos aplicados a la Biomedicina, Diseño de Microprocesadores, Control Digital, Sistemas Embebidos.



**Jorge Rivera** nació en El Rosario, Sinaloa, México en 1975. Recibió su grado de licenciado en ciencias del Instituto Tecnológico del Mar, Mazatlán, México en 1999, y los grados de maestro en ciencias y doctorado en ingeniería eléctrica del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Guadalajara, México en el 2001 y 2005, respectivamente. Desde 2006, ha estado en la Universidad de Guadalajara, Guadalajara, México, como profesor a tiempo completo en el departamento de ingeniería electrónica. Actualmente es comisionado como profesor CONACyT para el CINVESTAV, IPN, campus Guadalajara. Sus intereses de investigación se enfocan en teoría de reguladores, control de modo deslizante, sistemas de control no lineales en tiempo discreto, y sus aplicaciones a las maquinas eléctricas. Ha publicado más de 40 artículos técnicos en revistas internacionales y conferencias, y ha servido como revisor para diferentes revistas internacionales y conferencias.



**Adrian Pedroza** se recibió de la Licenciatura en Ingeniería en Comunicaciones y Electrónica en la Universidad de Guadalajara, Jalisco, México, en el 2008. En el 2010 recibió el grado de Maestro



en Ciencias en Ingeniería Electrónica y Computación en la Universidad de Guadalajara. Del 2010 al 2011 trabajó en Intel como Ingeniero de Diseño de Componentes. En 2015 recibió el grado de Doctor en Ciencias en Ingeniería Eléctrica en el Centro de Investigación y Estudios Avanzados del IPN Unidad Guadalajara, Jalisco, México. Hoy en día su investigación se centra en el área de aceleradores con arquitecturas de memorias en paralelo para procesamiento digital de imágenes y verificación de hardware.