

Cugnot Server: servidor de imágenes de vehículo

Cugnot server: vehicle image server

Yordanis Márquez Denis

Universidad de las Ciencias Informáticas.

ymarquez@estudiantes.uci.cu .

Resumen:

En el artículo se explica el papel del sistema **Cugnot Server** en la solución “**Cugnot: Suite para la identificación de los vehículos**”. Se muestran los requisitos del sistema, la plataforma de desarrollo, el lenguaje de programación y las tecnologías utilizadas. Se presentan además los tipos de mayor impacto en el diseño del sistema y por último se aborda el impacto económico y social de la solución Cugnot.

Palabras claves: .NET, ANPR, C#, Cugnot Suite, sistemas de identificación vehicular.

Abstract:

Cugnot Server is a component of “Cugnot: The automated vehicle identification Suite”. The behavior of Cugnot Server is explained in this paper. Moreover, functional and non-functional requirements, used development platform, used programming language and used technologies are described. The paper also presents the most critical system’s design types and special attention is dedicated to the contribution of Cugnot Suite to our society.

Key words: .NET, ANPR, Automated vehicle identification systems, C#, Cugnot Suite.

Introducción:

La identificación automática de los vehículos es un problema relevante en la actualidad. Esto se debe a la utilización masiva de los vehículos en los procesos de la sociedad y a los efectos de la urbanización, entre otros factores (Martín y Álvarez 2008).

Un sistema de identificación automática de los vehículos es el resultado de la convergencia entre la Electrónica, la Informática y la Matemática. Los medios electrónicos (e.g. detectores de presencia, sensores) permiten capturar la imagen del vehículo en la escena real (Anpr-tutorial 2006). Luego, la imagen se transmite hacia un computador para ser procesada utilizando técnicas de reconocimiento automático del número de la matrícula (Martín y Álvarez 2008). Conocida la matrícula, esta se inserta junto a otros datos (e.g. fecha, hora) en una base de datos que es consultada para ofrecer servicios a la medida. Algunos de estos servicios son el control de acceso y el monitoreo de las autopistas (Shapiro, y otros 2003).

El Polo productivo “**Imágenes**”, de la Universidad de las Ciencias Informáticas, está desarrollando la solución “**Cugnot: Suite para la identificación de los vehículos**”. La misma se compone de los sistemas **Cugnot Scheduler**, **Cugnot Engine**, **Cugnot Reviewer**, **Cugnot Server**, **Cugnot Repository** y **Cugnot Explorer**. Todos ellos han sido diseñados de acuerdo al estado del arte de sistemas afines a nivel internacional. Además, se dedica especial atención a la facilidad de uso y a la robustez.

En el artículo se detalla el funcionamiento del sistema **Cugnot Server**. Lo que resta del mismo se encuentra organizado de la siguiente forma: en la **sección 2.1** se explica el papel del servidor en la solución **Cugnot**. En la **sección 2.2** se hace referencia a los requisitos funcionales y no funcionales. La plataforma de desarrollo, lenguaje de programación y las tecnologías usadas son expuestas en la **sección 2.3**. Se muestra el diagrama de componentes en la **sección 2.4** y los principales métodos y sus responsabilidades en la **sección 2.5**. En la **sección 2.6** se describen algunas de las características de las herramientas usadas. La **sección 3** fue dedicada a impacto económico y social y la **sección 4** fue dedicada a las conclusiones.

Materiales y Métodos:

2.1 Papel del sistema en la Suite Cugnot.

El sistema **Cugnot Server** recibe imágenes vía **FTP** (www.faq.s.org 1985) . Actúa como un repositorio y tiene la tarea de registrar los eventos que sucedan en la partición del disco. Si se encuentra llena la partición del disco se procederá a un borrado o copia parcial hacia otro servidor. En cualquiera de los dos casos se hace comenzando por las imágenes que lleven mayor tiempo en el Servidor. Al ocurrir cualquiera de estas acciones es actualizada la Base de datos (Fig. 2.1).

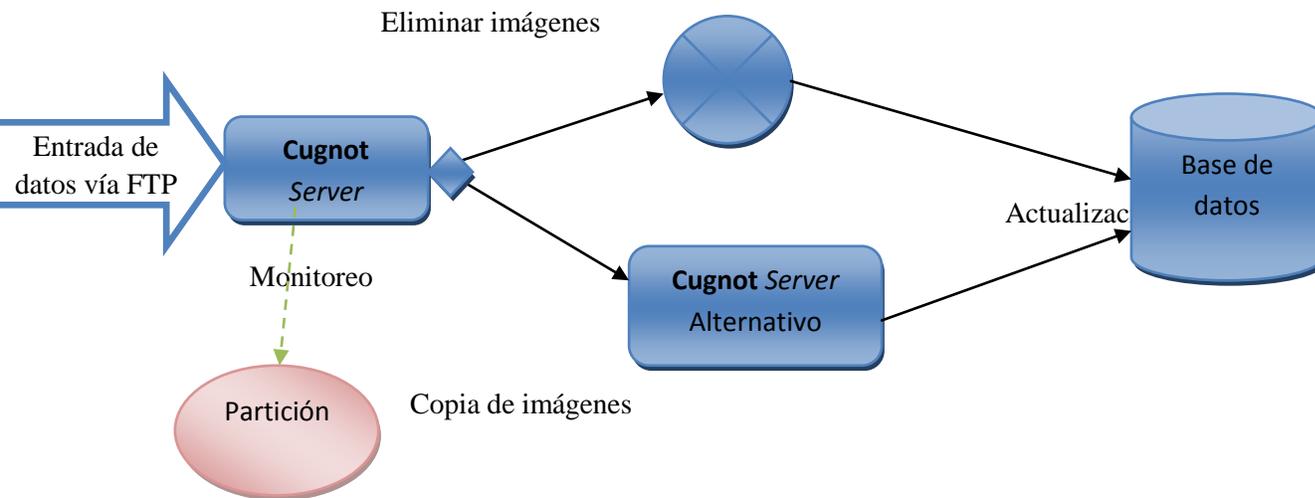


Fig. 2.1 Interacción del componente **Cugnot Server** con otros componentes del sistema **Cugnot**

2.2 Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, mientras que los requisitos no funcionales son propiedades o cualidades que el sistema debe tener (Pressman 2002). La **tabla 2.1** muestra los requisitos funcionales y la **tabla 2.2** muestra los requisitos no funcionales.

Tabla 2.1 Requisitos funcionales.

Identificador	Descripción
RF1	El sistema debe permitir configurar un tipo de reacción predeterminada ante una crisis de espacio libre en la partición.
RF2	El sistema debe ejecutar automáticamente la reacción configurada ante la crisis de espacio libre en la partición.
RF3	El sistema debe reportar hacia el registro de sucesos los hitos de su funcionamiento.
RF4	El sistema debe mostrar el estado del servidor.

Tabla 2.2 Requisitos no funcionales.

Identificador	Descripción
RNF1	El sistema debe ser compatible con la plataforma libre Mono.
RNF2	El sistema debe ejecutarse sin necesidad de que exista algún usuario autenticado en el equipo.

2.3 Plataforma de desarrollo, lenguaje de programación y tecnologías usadas.

Cugnot Server es desarrollado en el lenguaje C# de la plataforma .NET. C# facilita la práctica de buenas prácticas de programación (e.g. validación de datos, chequeo de tipos, reutilización de código). El lenguaje presenta características que lo hacen robusto, moderno y sencillo. Funciona bajo un entorno de recolección automática de memoria (Butow and Ryan 2002). Además permite utilizar tipos genéricos, indexadores, eventos, estructuras, interfaces y clases (Microsoft 2006).

2.4 Diagrama de componentes del sistema.

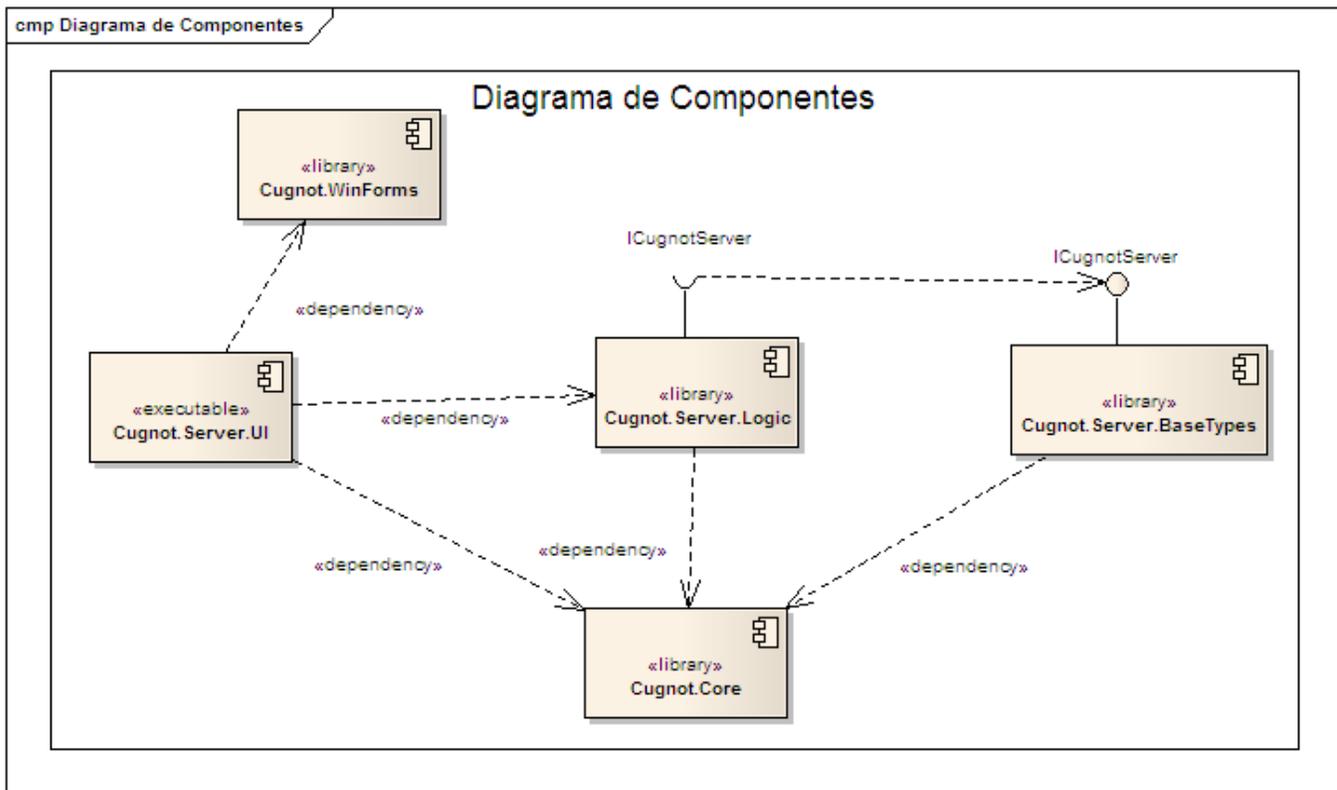


Fig. 2.2 Diagrama de componentes del sistema **Cugnot Server**.

2.5 Principales tipos y sus responsabilidades.

Las interfaces **ICugnotServer**, **IServerLoadProvider** e **IServerUrlUpdater** (Fig. 2.3) así como las clases **MyCugnotServer** y **CugnotServerController** (Fig. 2.4) son los tipos de mayor impacto en el sistema.

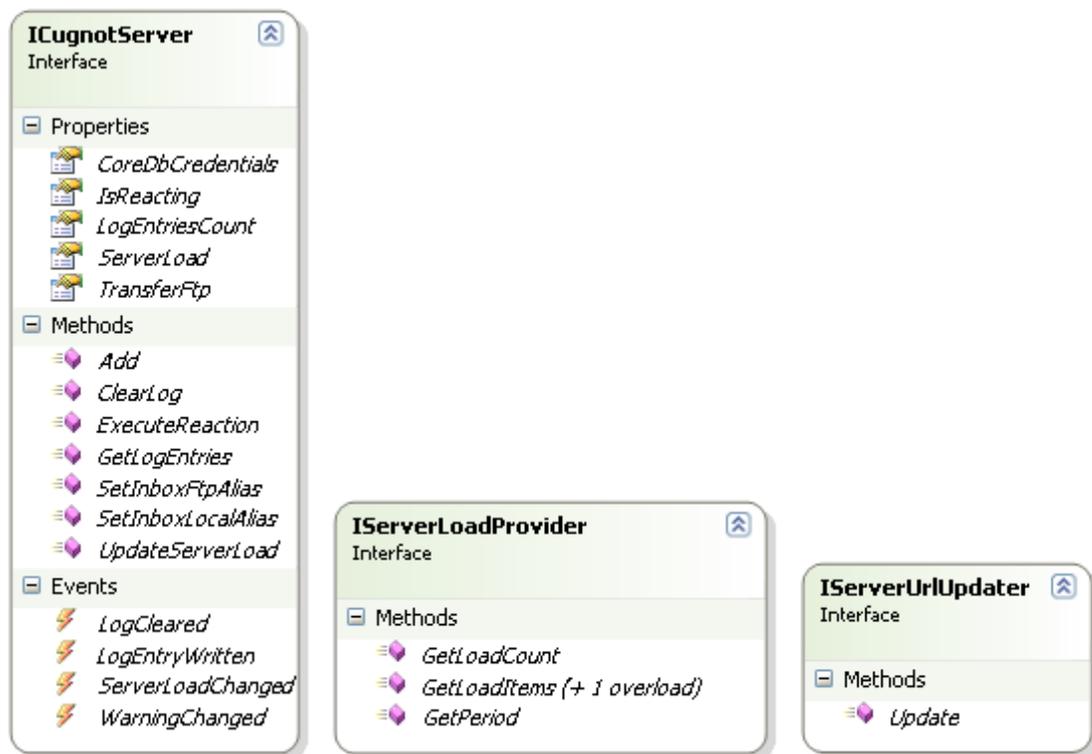


Fig. 2.3 Principales interfaces de **Cugnot Server** y sus miembros.

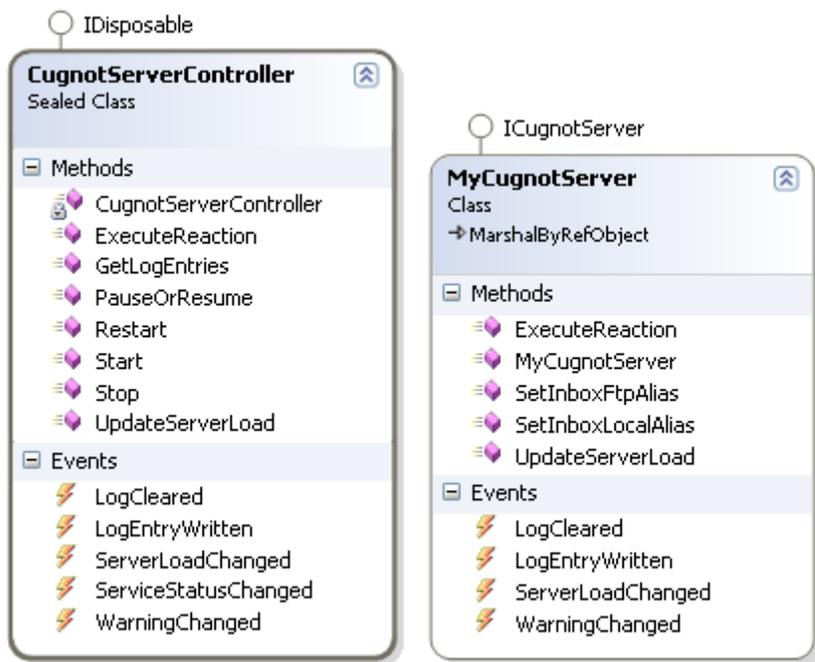


Fig. 2.4 Principales clases de **Cugnot Server** y sus miembros.

Tabla 2.3 Descripción de los métodos CugnotServerController .

Miembro	Descripción
Void Start()	Activa todas las funcionalidades de Cugnot Server
Void Restart()	Cumple con la función de reiniciar las funcionalidades del servidor.
Void PauseOrResume()	Pausa los servicios del servidor si estos se encuentran activos. Al volver a iniciar los servicios estos continúan en el estado anterior con que fueron parados.

Void Stop()	Interrumpe toda actividad en el servidor. Los servicios no conservan su estado al volver a ser solicitados.
Void ExecuteReaction()	Lleva a cabo las acciones de reacción a cambios. Las acciones pueden ser de borrado o de copia hacia otro servidor a través de FTP.
Void UpdateServerLoad()	Actualiza el estado del servidor.

2.6 Herramientas utilizadas.

Como herramientas de desarrollo se utilizó Visual Studio Team Suite 2008 y Visual Source Safe 2005. Visual Studio ofrece facilidades para modelar la arquitectura y el diseño del sistema e incorpora facilidades para optimizar la interacción de cada miembro del equipo (Microsoft corporation 2008). La aplicación Visual Source Safe 2005 implementa un eficaz mecanismo de control de versiones. Entre sus funcionalidades se destacan el control, la compartición y la combinación de los archivos (Microsoft corporation n.d.).

Impacto Económico y Social:

La puesta en práctica de la solución “**Cugnot: Suite para la identificación de los vehículos**” permitirá resolver el problema de la identificación de los vehículos en varios sectores. Los obstáculos que enfrentan las autoridades para monitorear el tráfico de los vehículos en la ciudad, pueden ser mitigados o eliminados desplegando la solución en los puntos de control existentes. Lo anterior permite consultar la huella dejada por un vehículo al circular por la ciudad y es vital para solucionar eventos de diversa índole (e.g. localización de autos robados o en fuga). Además, la solución puede ser utilizada para detectar violaciones de las leyes del tránsito, controlar el peaje y verificar el acceso a las instalaciones. En cuanto al aporte de divisas, una solución de este tipo está valorada en varios millones de dólares.

Conclusiones:

En el artículo fue abordado el papel del sistema **Cugnot Server** en la solución **Cugnot Suite**. Fueron plasmados los requisitos del sistema, plataforma de desarrollo, lenguaje de programación y las técnicas usadas. Fueron abordados también los tipos de mayor impacto en el diseño y desarrollo del sistema así como el impacto económico y social de la solución en su conjunto.

Referencias Bibliográficas

1. Anpr-tutorial. "Anpr tutorial." Anpr tutorial. 8 15, 2006. <http://www.anpr-tutorial.com/> (accessed 9 1, 2008).
2. Butow, Eric, and Tommy Ryan. C#: Your visual blueprint for building .NET applications. New York: Hungry Minds, Inc, 2002.
3. Martín, Jesmar Ernesto Fajardo, and Rafael Leodan Cardero Álvarez. "Localización y segmentación de matrículas de vehículos en imágenes digitales." Tesis de grado, La Habana, 2008.
4. Microsoft. C# language specification. Redmond, 2006.
5. Microsoft corporation. "Visual SourceSafe." Visual SourceSafe. [http://msdn.microsoft.com/en-us/library/ms181038\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ms181038(VS.80).aspx) (accessed 9 1, 2008).

6. —. "Visual Studio 2008 Team Suite." MSDN. 2008. <http://msdn.microsoft.com/en-us/vsts2008/default.aspx> (accessed 9 1, 2008).
7. Pressman, Roger S. Ingeniería de software: Un enfoque práctico. Addison-Wesley, 2002.
8. Shapiro, Vladimir, Dimo Dimov, Stefan Bonchev, Veselin Velichkov, and Giorgi Gluhchev. "Adaptive license plate image extraction." Conference Report, 2003.
9. www.faqs.org. "Network Working Group." Request for Comments-FILE TRANSFER PROTOCOL (FTP). 10 1985. <http://www.faqs.org/ftp/rfc/pdf/rfc959.txt.pdf> (accessed 9 14, 2008).