

# **Plataforma para el desarrollo de servicios en línea utilizando Tarjetas Inteligentes.**

**Ander Sánchez Jardines**

Universidad de las Ciencias Informáticas , La Habana, Cuba, ajardines@uci.cu

**Vismar Fernández Santana**

Universidad de las Ciencias Informáticas, La Habana, Cuba, vsantana@uci.cu

## **ABSTRACT**

Nowadays most applications that offer services by using smartcards require the installation of certain software packages (commonly known as middleware) that allow interaction between those applications and the card. The absence of a technology that allows online communication with smartcards using different web browsers and heterogeneous operating systems has caused that the benefits that they can provide for services over the Internet haven't been properly exploited. The Center for Identification and Digital Security (CISED) belonging to the University of the Informatic Sciences (UCI), due to the experience obtained in the field of smart devices for identification, has identified the need to develop a middleware platform on the server side in order to prevent the installation and updating of certain software on the final user's PC, and also to provide greater security in communication with the cards on the web.

This article sets a relation between smartcards and the provision of services on the web, mentioning basic concepts associated with that topic. As a result of analysis, it is developed a multi-platform application for the interaction with smartcards from web browsers. The article presents the results of the investigation, in addition to the architecture and design of the proposed system. It also describes the tools and technologies that were used, and the main artifacts that were generated in the development process.

**Keywords:** Smart cards, Middleware, Online Services, Web Browsers, multiplatform application, Security.

## **RESUMEN**

La mayoría de las aplicaciones que ofrecen servicios mediante el uso de tarjetas inteligentes, requieren la instalación en su PC (personal computer) de determinados paquetes de software, comúnmente denominados middleware, que posibilitan la interacción entre dichas aplicaciones y la tarjeta. La falta de una tecnología que permita la comunicación en línea con las tarjetas inteligentes y que además, sea suportada por diferentes tipos de navegadores web y sistemas operativos, ha imposibilitado un aprovechamiento óptimo de las ventajas de estos dispositivos inteligentes de identificación en la prestación de múltiples servicios a través de Internet. El Centro de Identificación y Seguridad Digital (CISED) de la UCI, a partir de la experiencia adquirida en el área de las tarjetas inteligentes, ha identificado la necesidad del desarrollo de una plataforma que posibilite la ubicación de middleware en el lado del servidor, evitando al usuario la instalación de determinado software en su PC y la actualización periódica del mismo, además de brindar una mayor seguridad en la comunicación con las tarjetas en la web.

El presente artículo relaciona las tarjetas inteligentes con la prestación de servicios en Internet, mencionando conceptos asociados a dicho vínculo. Como resultado del análisis, se desarrolla una aplicación multiplataforma para la interacción con tarjetas a través de navegadores web. El artículo recoge los resultados de la investigación realizada así como la arquitectura y el diseño del sistema propuesto. Se describen las herramientas y tecnologías utilizadas además de los principales artefactos generados en el proceso de desarrollo.

**Palabras claves:** Tarjetas inteligentes, Middleware, Servicios en línea, Navegadores web, Aplicación multiplataforma, Seguridad.

## 1. INTRODUCCIÓN

El surgimiento de Internet en la segunda mitad del pasado siglo ha revolucionado la manera de vivir y pensar del ser humano. En la actualidad se brindan un sinnúmero de servicios en línea por instituciones, gobiernos o grandes empresas; a los cuales el usuario o beneficiario accede desde computadores. Entre estos se pueden destacar el gobierno en línea, las pasarelas de pago, los servicios de correo electrónico, la reservación de servicios hoteleros o de viaje, la certificación de documentos electrónicos, la emisión de declaraciones de impuestos o la tramitación de documentos oficiales de una nación determinada.

La prestación de estos servicios implica determinados riesgos cuando se hace a través de Internet, ya que sin la presencia física del solicitante es una preocupación lógica verificar la identidad del usuario, descartando que sea alguien que intenta suplantarla. Actualmente se registra una tendencia al uso de dispositivos inteligentes que facilitan y aseguran el proceso de identificación. Se han desarrollado tecnologías como las tarjetas inteligentes para garantizar una verificación más exacta y confiable de la identidad del usuario que solicita determinado servicio. A pesar que estas son notablemente populares en sistemas de control de acceso, licencias de conducción, cajeros automáticos y otras variantes, los múltiples servicios que se prestan a través de la web continúan sustentándose en la típica, antigua y vulnerable autenticación mediante el uso de usuarios y contraseñas u otros medios poco robustos.

Lo anterior conduce al siguiente cuestionamiento: ¿podrían mejorarse o beneficiarse los servicios en línea con la utilización de tarjetas inteligentes? Evidentemente la posibilidad de reemplazar los tradicionales modos de autenticación en la web, por uno que utilice esta novedosa tecnología, dotaría a los sistemas actuales de una mayor seguridad y calidad de servicio, además se incrementaría la confiabilidad de las transacciones a través de Internet mediante la verificación de la identidad, no solo del usuario sino también el proveedor del servicio. Actualmente es muy común en la web, no solo la suplantación de determinado usuario para recibir beneficios inmerecidos de alguna institución, sino también la suplantación de instituciones para obtener datos personales de clientes que posteriormente serán mal usados; tanto una cosa como la otra serían inconcebibles con una interacción sustentada en la tecnología de las tarjetas inteligentes. El uso de las mismas en las aplicaciones web potencia además la implementación de la infraestructura de clave pública (PKI, por las siglas en inglés de Public Key Infrastructure) para la certificación de documentos y la firma digital. Por otra parte no debe olvidarse que estos dispositivos son denominados “inteligentes” por la presencia de un microprocesador y el almacenamiento de aplicaciones que se ejecutan en su interior; gracias a estos factores podrían jugar un papel protagónico en diversas funcionalidades de una página web, procesándolas del lado del cliente y aliviando la carga del servidor, lo cual se traduciría en un servicio más eficiente para el usuario y en una mayor seguridad ya que dichas operaciones no tendrían lugar en el computador sino en el interior de la tarjeta.

En la actualidad el modo más usual de interactuar con las tarjetas inteligentes es a través de capas o librerías de software, técnicamente conocidas como middleware, que normalmente corren en la estación cliente y hacen función de intermediarios entre las diversas aplicaciones que prestan un servicio y aquellas que lo complementan desde el interior de la tarjeta. El manejo de esta comunicación mediante el uso de middleware que se ejecuten en el cliente trae consigo algunas desventajas o riesgos considerables; por ejemplo para las actualizaciones del software o la incorporación de nuevas funcionalidades a estas librerías habría que distribuirlas por todos los clientes de un sistema determinado o publicarlas en un sitio web para que sean descargados a través de la red. Esto, además de ser incómodo para el cliente, implica que los usuarios que pretenden interactuar con sus tarjetas posean ciertos conocimientos para efectuar las actualizaciones y una serie de permisos en el manejo de los recursos de la computadora para poder instalarlas. Otra de las principales desventajas es la que está relacionada con la seguridad, pues las llaves simétricas necesarias para establecer la comunicación con la tarjeta, permanecen en el código a riesgo de ser descifradas. A todo eso se añade el hecho de que se limita grandemente la utilidad de las tarjetas al no poder ser usadas en la web.

El principal objetivo de la presente investigación es implementar una plataforma que permita la prestación de servicios en línea mediante el uso de tarjetas inteligentes, posibilitando el acoplamiento de middleware de

distintos propósitos y proveedores en el lado del servidor y garantizando la comunicación de los mismos con las tarjetas de los usuarios. En este trabajo se introducen un grupo de conceptos asociados a las tarjetas inteligentes y la web, definiendo las tecnologías, las metodologías y las herramientas para el cumplimiento del objetivo de la investigación.

## **2. MATERIALES Y METODOS**

Teniendo en cuenta el objetivo definido (la implementación de una plataforma que permita la prestación de servicios en Internet haciendo uso de las tarjetas inteligentes) e identificando como principales tareas la creación de un componente de extensión para los principales navegadores web y el desarrollo de una arquitectura flexible del lado del servidor que permita el acoplamiento de middleware de distintos proveedores y propósitos; se hace el análisis de los principales materiales y métodos que serán vitales para la investigación.

### **2.1 CONCEPTOS ASOCIADOS**

#### **2.1.1 TARJETAS INTELIGENTES (SMARTCARDS)**

Son dispositivos que contienen un chip en su interior, por lo que también se les conoce como tarjetas con circuito integrado (TCI). La presencia de dicho circuito posibilita la ejecución de cierta lógica programada, de ahí que se utilice el adjetivo “inteligente” en su denominación.

#### **2.1.2 ADD-ONS**

Son módulos que proporcionan nuevas funcionalidades, mejorando las prestaciones de un programa. También se hacen llamar plug-in. Entre las aplicaciones más comunes que suelen incluirlos están los navegadores web ya que amplían las funciones de las páginas web para ver contenidos interactivos, videos y cosas similares. Son especialmente populares para Mozilla Firefox, cuyos componentes de extensión son desarrollados bajo el marco de desarrollo XPCOM( por las siglas en inglés de Cross Platform Component Object Model). Internet Explorer es también extensible mediante la incorporación de controles ActiveX, basada en el COM (por las siglas en inglés de Component Object Model).

#### **2.1.3 MIDDLEWARE**

Tomando algunas ideas de la revista (Toward a Comprehensive Framework for Evaluating the Core Integration Features of Enterprise Integration Middleware Technologies, 2013) un middleware es un software que funciona como una capa de abstracción, situada entre la de aplicación y las inferiores haciendo posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. En general son usados para relacionar sistemas que necesitan intercambio de información, permitiendo realizar la conexión a través de interfaces de alto nivel. (Rodríguez-Molina, et al., 2013)

### **2.2 METODOLOGÍA EXTREME PROGRAMMING (XP)**

XP, forma parte del conjunto de métodos ágiles que centran sus prioridades en las personas, no en los procesos. En la actualidad XP se proyecta a ser un modelo de desarrollo común, sencillo y adaptable a las características cambiantes y exigentes de empresas y clientes.

Los elementos determinantes para su elección fueron: la composición por solo dos integrantes del grupo de desarrollo, el hecho de el producto del desarrollo no esté orientado a un cliente específico sino que esté impulsado por propia iniciativa de los desarrolladores, evitar la cantidad excesiva de documentación que exigen las metodologías robustas.

### **2.3 ESTÁNDARES ASOCIADOS A LAS TARJETAS INTELIGENTES**

Dado la existencia de múltiples fabricantes de tarjetas inteligentes, lectores, programadores de aplicaciones tanto internas (*applets*) como externas que complementan las anteriores (*middleware*) se ha definido internacionalmente una serie de estándares de obligatoria consulta para toda aquella persona o entidad que desee participar en el

crecimiento de esta tecnología o sustentar determinado servicio en el uso de la misma. Para la realización de esta investigación los siguientes estándares fueron un material de vital importancia.

### 2.3.1 ISO/IEC 7816

El objetivo de estos estándares es lograr la interoperabilidad entre distintos fabricantes de tarjetas inteligentes y lectores, en lo que respecta a características físicas, comunicación de datos y seguridad. Estos estándares son basados en los ISO 7810 y 7811, los cuales definen características físicas de tarjetas. (ISO, 2011)

La implementación de la presente solución se centra en el estándar ISO/IEC 7816-4 el cual define el contenido de los pares comando - respuesta que se intercambian a nivel de interfaz entre las aplicaciones de una tarjeta inteligente y el middleware, así como estructuras para aplicaciones y datos en la tarjeta.

### 2.3.2 GLOBAL PLATFORM

Global Platform es una organización independiente enfocada en gestionar una infraestructura estandarizada para el desarrollo y despliegue de tarjetas inteligentes. Proporciona un conjunto de especificaciones universalmente reconocidas e implementadas, junto con configuraciones de mercado, aplicación de esas especificaciones y documentos de apoyo. Cubriendo toda la infraestructura de tarjetas inteligentes estos documentos técnicos ofrecen una plataforma tecnológica dinámica y completa para el desarrollo de programas para esta tecnología.

## 2.4 HERRAMIENTAS UTILIZADAS

### 2.4.1 ALTOVA UMODEL

Es una herramienta de modelado que permite el diseño visual en UML 2.0 de flujos de negocios, sistemas y aplicaciones y genera código en diferentes lenguajes tales como Java, C#, o Visual Basic .NET, así como documentación del proyecto. "UModel es la herramienta UML que hace el diseño visual de software práctico para cualquier proyecto. Es una manera simple y asequible de dibujar en UML". (businessweek, 2010) (altova, 2010)

### 2.4.2 VISUAL STUDIO 2010

Visual Studio es un entorno de desarrollo integrado (IDE, por las siglas en inglés de Integrated Development Environment) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio 2010 sirvió como entorno de desarrollo para los múltiples componentes de la solución, tanto para los objetos COM que servirán de complemento a los diferentes navegadores, los componentes genéricos del lado del servidor, así como para la creación de un sitio web de prueba donde se comprobó la funcionalidad de la solución creada.

## 2.5 PLATAFORMA

### 2.5.1 MICROSOFT .NET FRAMEWORK

El Framework .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. (netFramework, 2010)

## 2.6 TECNOLOGÍAS Y LENGUAJES UTILIZADOS EN LA PROGRAMACIÓN DE LOS COMPONENTES

### 2.6.1 C SHARP

C Sharp es un lenguaje de programación orientado a objetos estandarizado por Microsoft como lenguaje bandera de la plataforma .NET. Su sintaxis básica deriva de sus predecesores C y C++ y utiliza el modelo de objetos de la plataforma en la cual se sustenta.

Se utilizó este lenguaje en el desarrollo de los componentes del lado del servidor que serán explicados más adelante en este mismo artículo.

### 2.6.2 ASP.NET

ASP.NET es una tecnología gratuita que permite a los programadores crear páginas web dinámicas, desde sitios web personales hasta aplicaciones web empresariales.

ASP.NET con lenguaje C Sharp posibilitó el desarrollo de una aplicación web que permitiera probar la compatibilidad y funcionamiento de los componentes generados como parte de la solución.

### 2.6.3 JAVASCRIPT EN EL CLIENTE

Se usó JavaScript en su versión del lado del cliente para la interacción con el componente (ActiveX para Internet Explorer o Add-on para Mozilla Firefox) que garantizará la comunicación entre los navegadores y las tarjetas.

### 2.6.4 C++

C++ es un lenguaje de programación orientado a objetos, aunque algunas bibliografías lo clasifican como híbrido. Es particularmente útil en el manejo de librerías nativas de diferentes sistemas operativos y además una de sus características fundamentales es la de generar programas de un alto nivel de independencia. Teniendo en cuenta esto último se decidió el uso de este lenguaje para la codificación del componente del lado del cliente (ActiveX para Internet Explorer o Add-on para Mozilla Firefox) para de este modo liberar al usuario de la instalación de cualquier otro software adicional, salvo el manejador (driver) de su lector de tarjetas.

### 2.6.5 ASYNCHRONOUS JAVASCRIPT AND XML (AJAX)

Ajax es una técnica de desarrollo web para crear aplicaciones interactivas o aplicaciones de internet enriquecidas. “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes” (Jesse James, 2005). Estas aplicaciones se ejecutan en el cliente, o sea, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Así es posible realizar cambios sobre las páginas sin necesidad de recargarlas completamente.

El uso de AJAX en la implementación de la solución es de vital importancia, para lograr que el proceso continuo de intercambio de información entre las tarjetas inteligentes del lado del cliente y los middleware acoplados en el lado del servidor se produzca de una manera asíncrona, sin afectar al usuario.

## 2.7 ANÁLISIS DE OTRAS SOLUCIONES

**Coesys eGov 2.0** producida por la empresa Gemalto, es una plataforma que permite un servicio de identificación electrónica mediante tarjetas inteligentes basado en la web, en vez de un software basado en un cliente de autenticación de instalación local. Esta solución evita la administración de un middleware en el cliente, toda la funcionalidad requerida se centraliza en un servidor.

*Coesys eGov 2.0* funciona con cualquier navegador en cualquier sistema operativo permitiendo una conexión “*plug and play*” desde todos los ordenadores. Su tecnología principal se sustenta en el producto **SConnect** de la misma compañía, que permite una conexión neutral con tarjetas inteligentes. Representa un cambio en el paradigma de las tarjetas inteligentes y los servicios web. (Gemalto, 2007)

---

<sup>1</sup> Es un sistema que permite conectar cualquier dispositivo de *hardware* al ordenador, sin tener que incorporar o instalar ningún controlador, pues la configuración se realiza de forma automática. Esto supone un aumento en la facilidad de instalación y configuración de nuevos periféricos.

**SConnect** es la principal novedad que compone a **Coesys eGov 2.0**, es una extensión para los navegadores más importantes, es compatible con los sistemas operativos Windows, Mac OSX y Linux. Su objetivo principal es el de proporcionar un puente de conexión entre el JavaScript, que corre en la página web de un navegador y la tarjeta inteligente, permitiendo la conectividad entre estas últimas aplicaciones y los servicios web. **SConnect** consiste en dos partes:

- Una extensión del navegador web que conecta con la capa PC/SC estándar del ordenador, conectando una página web con una tarjeta inteligente.
- Una librería JavaScript que permite a los desarrolladores de aplicaciones web tener acceso a tarjetas inteligentes mediante SConnect.

SConnect incluye un sistema de medidas de seguridad para mitigar ciertos riesgos y proteger a los usuarios y sitios web conectados. Estas medidas incluyen una extensión con firma digital de SConnect, llave de conexión, validación de servidor, alarmas al usuario y la implementación de un protocolo HTTPS (por las siglas en inglés de Hypertext Transfer Protocol Secure) reforzado por el establecimiento de un canal seguro en ambas direcciones, validado por las llaves criptográficas compartidas entre determinado middleware y las aplicaciones que lo complementan desde el interior de la tarjeta.

A pesar de la excelencia de **SConnect**, que es una de las fuentes de inspiración para esta investigación, las limitaciones de **Coesys 2.0** consisten en que del lado del servidor es una plataforma no adaptable para el acoplamiento de middleware de distintos proveedores y prestadores de servicios en la web. Es una solución implementada a la medida de lo que Gemalto necesita exponer al mundo para promover el uso de sus tarjetas inteligentes, no un producto genérico.

### 3. RESULTADOS Y DISCUSIÓN

Los principales artefactos generados como producto de este proceso de investigación y desarrollo fueron los componentes de la plataforma: el componente de extensión para los diferentes navegadores web, las librerías clientes para el manejo de dicho componente y las librerías que posibilitan el acoplamiento de distintos middleware en el servidor. Adicionalmente se desarrolló una especie de sitio web que integrara todos estos factores en la prestación de tres servicios ficticios: un monedero electrónico para el pago de servicios hoteleros y de viaje, un servicio de identificación o credencial electrónica y un consultor de especificaciones técnicas de la tarjeta inteligente del portador. Durante todo el proceso de análisis, diseño y desarrollo de la plataforma se generaron además varios documentos, modelos y diagramas que son presentados a continuación.

#### 3.1 MODELO DE DOMINIO

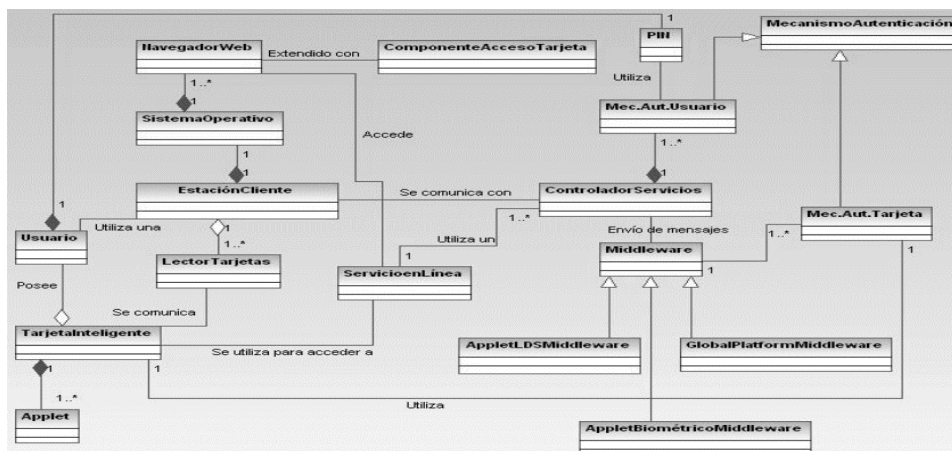


Ilustración 1: Diagrama de clases del modelo de dominio

Teniendo en cuenta que el problema científico que la plataforma pretende resolver no está determinado partir de procesos bien definidos que permitan modelar el funcionamiento de la misma, se procedió a crear un modelo de dominio. Los elementos más significativos de la figura anterior lo constituyen el “**Componente Acceso Tarjeta**” y el “**Controlador Servicios**”, el primero de ellos es una extensión para los diferentes navegadores web que posibilitará la comunicación con las aplicaciones de la tarjeta a través de lectores conectados en la estación cliente. El otro constituye un controlador que permitirá de una manera sencilla el acoplamiento de middleware de distintos fabricantes y propósitos, que usarán la plataforma como mecanismo de comunicación con la tarjeta del usuario con el fin de brindar determinado servicio. Tal como expresa la figura, del lado del servidor son admitidos tantos middleware como el dueño del sitio web desee acoplar.

De manera resumida, el modelo de dominio se traduce de la siguiente manera: para la prestación de determinado servicio web será necesario establecer una comunicación entre determinada aplicación en el interior de la tarjeta que está conectada a la PC del usuario y el correspondiente middleware acoplado del lado del servidor, estableciéndose un puente donde los principales pilares son el “**Componente Acceso Tarjeta**” encargado de extender la funcionalidad de los exploradores web para permitir el acceso a las tarjetas inteligentes y “**Controlador Servicios**” que funciona como capa de traducción de los mensajes que llegan desde el cliente web y los diferentes middleware residentes en el servidor.

### 3.2 HISTORIAS DE USUARIO ARQUITECTONICAMENTE SIGNIFICATIVAS

Las historias de usuario son utilizadas en XP para especificar los requisitos del *software*. Para este producto se han definido como significativos los siguientes.

- Obtener lectores disponibles conectados a la estación cliente.
- Gestionar conexión con la tarjeta inteligente.
- Enviar y recibir comandos desde la tarjeta.
- Gestionar transmisión asincrónica de comandos entre el cliente web y el servidor y viceversa.
- Ejecutar operaciones de un middleware en el servidor.
- Notificar a la aplicación web sobre culminación de la operación requerida por el usuario.

### 3.3 ARQUITECTURA DEL SISTEMA

La plataforma para el desarrollo de servicios en línea utilizando tarjetas inteligentes basa su implementación en la conocida arquitectura cliente/servidor. Teniendo en cuenta las características de la plataforma, cuyo principal papel es mediar entre los sitios web que presten servicios en línea y los lectores de tarjetas establecidos en la estación cliente, se identifica una capa cliente que contiene los componentes para comunicarse con el lector de tarjetas y canalizar las peticiones hacia el controlador del lado del servidor. En segundo lugar se tiene la capa servidor, donde se encuentra el resto de los componentes del sistema interconectándose entre sí para resolver las solicitudes que vienen desde el cliente, además de gestionar los middleware que se encuentran asociados a la plataforma y las funcionalidades contenidas en ellos accediendo a sus librerías.

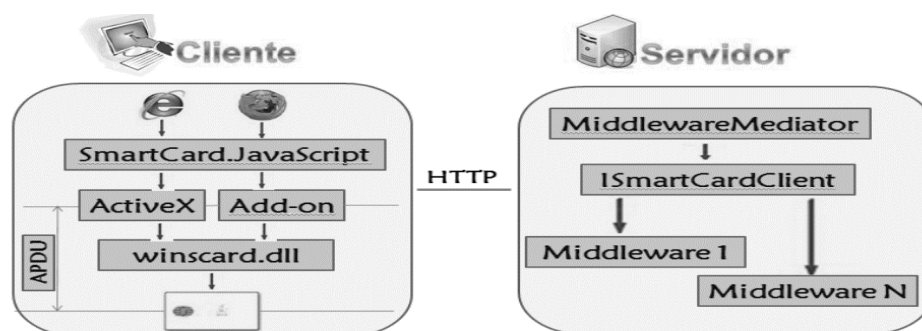


Ilustración 2: Arquitectura del sistema

En la figura previa se representan los principales componentes que garantizan el funcionamiento de la plataforma. Del lado del cliente se observa la existencia de una extensión para el navegador preferido por el usuario (**ActiveX** o **Add-on**), que será manejada a partir de un componente desarrollado con el uso de AJAX (**SmartCard.JavaScript**) que garantizará de manera asíncronica el envío y recepción de comandos desde la tarjeta y su respectivo enrutamiento hacia el servidor. En el servidor se representa el componente **MiddlewareMediator**, el principal controlador que recibirá las peticiones y comandos desde el cliente y las entregará al middleware específico que pueda dar respuesta a los mismos. La interfaz **ISmartCardClient** expondrá las funcionalidades básicas que deberán ser implementadas por todos aquellos proveedores de middleware que deseen prestar determinado servicio haciendo uso de la plataforma.

### 3.4 DESCRIPCION DEL DISEÑO

La solución está estructurada en dos paquetes, se divide en la capa cliente y la capa servidor, respondiendo a una arquitectura de dos capas. En la primera se encuentran los componentes necesarios para establecer y manejar la comunicación con la tarjeta inteligente y el servidor. En la segunda se encuentran las clases que manejarán los diferentes middleware anexados en la plataforma y los servicios ofrecidos por la aplicación, permitiendo la centralización de los mismos y su prestación a varios clientes en forma concurrente. El uso de esta arquitectura permite cierto grado de independencia entre las partes de la aplicación, característica que la convierten en una solución escalable ya que se pueden quitar o agregar middleware sin afectar su funcionamiento.

A continuación se muestran dos figuras que ejemplifican de manera general los componentes de la plataforma y otros complementos que aunque no son componentes de la misma participan en el proceso de prestación de un servicio para el usuario.

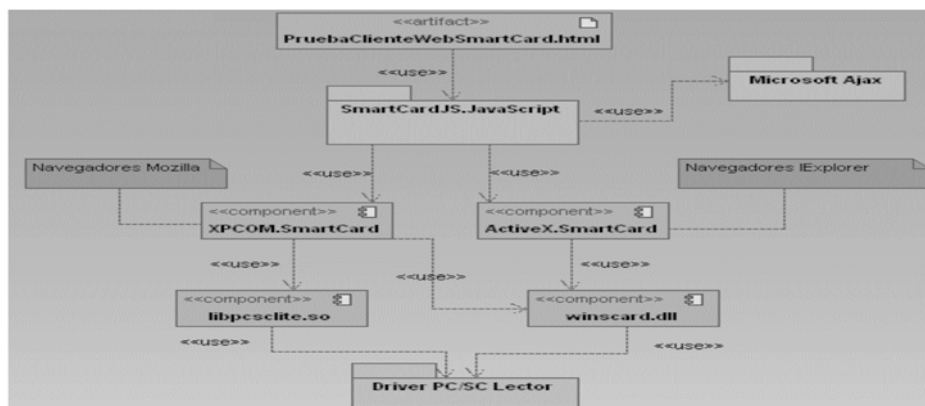


Ilustración 3: Diagrama de componentes del cliente

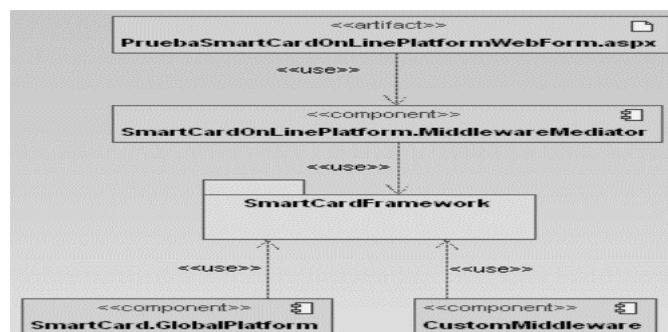


Ilustración 4: diagrama de componentes del servidor

En la ilustración 3 el componente **PruebaClienteWebSmartCard.html** representa una simple página web con la que interactúa el usuario para solicitar determinado servicio. La misma usa referencias al paquete



**SmartCardJS.JavaScript** con el objetivo de interactuar con la tarjeta, a través de las componentes **XPCOM.Samrtcard** o **ActiveX.Smartcard** que extienden las funcionalidades de Firefox e Internet Explorer respectivamente. Ambos componentes hacen uso de librerías nativas de los sistemas operativos de Windows y Linux para interactuar los lectores de tarjeta.

La ilustración 4 representa los componentes del lado del servidor es apreciable la presencia de la librería **SmartCardOnLinePlatform.MiddlewareMediator**, componente mediador entre el **SmartCardJS.JavaScript** del lado del cliente y los middleware acoplados en el servidor. Sus principales funciones son invocar las operaciones sobre una solicitud del usuario y dirigir las transmisiones de comando hacia y desde el cliente, garantizando su enrutamiento hacia el middleware específico que garantiza la respuesta a la solicitud del cliente. Los componentes **SmartCard.GlobalPlatform** y **CustomMiddleware** son dos ejemplos ficticios de middleware que pueden estar acoplados a la plataforma y su dependencia del componente **SmartCardFramework** expresa la necesidad de implementar una interfaz común para posibilitar su uso en la plataforma.

### 3.4.1 DESCRIPCIÓN DEL FLUJO DE PROCESO: SOLICITUD DE UN SERVICIO.

El proceso fundamental que ejecuta la plataforma es atender la solicitud de un servicio hecha por un usuario. A continuación una breve descripción:

1. El usuario hace click en un botón de la página web solicitando determinado servicio, como consecuencia de este evento la página experimenta una recarga parcial (asincrónica).
2. En cada recarga se obtiene una instancia de la clase **MiddlewareMediator** (controlador en el servidor). La instancia es única para una misma sesión web.
3. Como resultado al click del usuario la página intenta, a través de la instancia de **MiddlewareMediator** invocar la operación específica de middleware que da respuesta a la solicitud; para ello envía el nombre de la operación, los parámetros necesarios para la ejecución de la misma y el manejador de eventos que se encargará de procesar la respuesta una vez concluido el proceso. Inmediatamente después de la invocación se le ordena a la clase **SmartCardClientMediator** (controlador del lado del cliente) que comience el chequeo constante en espera de nuevos comandos enviados por el middleware específico desde el servidor hacia la tarjeta.
4. En cada chequeo si la instancia de **MiddlewareMediator** se encuentra en estado “procesando” se envían al cliente los comandos (uno en cada petición o recarga parcial de la página) que llegan como resultado de la operación previamente ordenada a un middleware específico.
5. Una vez que **SmartCardClientMediator** ha obtenido el comando enviado desde **MiddlewareMediator**, utiliza una instancia de **SmartCardReaderActiveXManager** (componente de extensión para los navegadores) para enviar dicho comando a la tarjeta. La respuesta de la tarjeta retorna por el mismo canal hacia el **SmartCardClientMediator**.
6. La respuesta será enviada a **MiddlewareMediator**, quién lo enviará al middleware específico que está aún procesando la operación que fue invocada. Si existe algún nuevo comando (en la comunicación con tarjetas inteligentes la mayoría de las operaciones requieren el intercambio de varios comandos antes de concluir) se enviará hasta la tarjeta por el mismo procedimiento explicado anteriormente, repitiendo este proceso hasta que el middleware haya obtenido los datos necesarios para elaborar una respuesta entendible para el usuario.
7. Una vez que el middleware le notifica a **MiddlewareMediator** la culminación de la operación, llamará al manejador que fue especificado en el momento de la invocación, para que procese la respuesta y la envíe al **SmartCardClientMediator**.
8. Este último gestiona la interfaz para mostrarle el resultado al usuario.

## 4. CONCLUSIONES

Con el desarrollo de este trabajo se ha podido demostrar cómo se han cumplido los objetivos principales definidos para la creación de una plataforma que permita la comunicación entre aplicaciones web y tarjetas inteligentes,

impulsando de esta forma el desarrollo de los servicios en línea que la utilicen. Las principales conclusiones tras el desarrollo de esta primera versión del producto son las siguientes:

- La investigación de las tecnologías relacionadas con el objeto de estudio permitió una correcta elección de la metodología y herramientas para el desarrollo de la solución.
- La utilización de una tecnología nativa para el desarrollo de un componente de extensión para el navegador, facilitará el uso de la plataforma por parte del cliente, evitándole tener que instalar software adicional en su PC.
- Los componentes que forman parte del sistema final, son también soluciones de un alto nivel de independencia, que, con mínimos cambios en su configuración, pueden ser usadas por sí solas de acuerdo a los requerimientos de otros sistemas.
- El desarrollo del sistema permite la comunicación entre una aplicación web y las tarjetas, con el middleware del lado del servidor, ofreciendo a los usuarios una mayor seguridad y confianza en este tipo de aplicaciones y facilitando la actualización de las aplicaciones anexadas al servidor, así como la incorporación de nuevas funcionalidades, todo de forma transparente al usuario final.

## 5. REFERENCIAS

- altova. 2010.** manual.altova.com. *manual.altova.com*. [Online] 2 21, 2010. [Cited: 1 30, 2013.] [manual.altova.com/umodel/umodelbasic/index.html?umtutorial.htm](http://manual.altova.com/umodel/umodelbasic/index.html?umtutorial.htm).
- businessweek, bloomer. 2010.** investing.businessweek.com. *investing.businessweek.com*. [Online] 2 12, 2010. [Cited: 1 21, 2013.] <http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=51794805>.
- Cristian, Toma. 2009.** Sample Development on Java Smart-Card Electronic Wallet Application. *Doaj*. [Online] 3 21, 2009. [Cited: 1 12, 2013.] <http://www.doaj.org/doaj?func=abstract&id=1213314&q1=7816&f1=all&b1=and&q2=&f2=all&recNo=2&uiLanguage=en.20674074>.
- Effing, Wolfgang Rankl and Wolfgang. 2003.** *Smart Card HandBook*. New York : John Wiley & Sons Ltd, Baffins Lane, Chichester, 2003.
- Global Platform. 2010.** Global Platform, Inc. *Global Platform, Inc*. [Online] 12 20, 2010. [Cited: 3 21, 2012.] [www.globalplatform.org/specifications.asp](http://www.globalplatform.org/specifications.asp).
- IBM. 2006.** developerWorks. *developerWorks*. [Online] 2 10, 2006. [Cited: 1 21, 2013.] <http://www.ibm.com/developerworks/webservices/library/co-xpcom/index.html>.
- ISO. 2011.** Anexo B Estandar ISO 7816. *ISO*. [Online] 2 17, 2011. [Cited: 3 21, 2012.] [www.tataware.com/tesis\\_maestria/13%20-%20Anexo%20B%20-%20Est%20EIndares%20ISO\\_IEC%207816.pdf](http://www.tataware.com/tesis_maestria/13%20-%20Anexo%20B%20-%20Est%20EIndares%20ISO_IEC%207816.pdf).
- Marquéz, Perez. 2010.** Pangea.org. *Pangea.org*. [Online] Pangea, 30 12, 2010. [Cited: 1 21, 2013.] <http://peremarques.pangea.org/glosinfo.htm>.
- Microsoft. 2010.** Asp.net. *Asp.net*. [Online] 2 10, 2010. [Cited: 2 1, 2013.] <http://msdn.microsoft.com/es-es/asp.net/centrum-asp-net.aspx>.
- netFramework. 2010.** desarroloweb.com. *desarroloweb*. [Online] 2 5, 2010. [Cited: 1 21, 2013.] <http://www.desarroloweb.com/articulos/1328.php>.
- oocities. 2010.** www.oocities.org. *www.oocities.org*. [Online] 3 12, 2010. [Cited: 1 12, 2013.] <http://www.oocities.org/espanol/licruthsilva/ActiveX1112.htm>.
- Pérez, Javier Eguíluz. 2009.** *Introducción a javascript*. 2009.
- Rodríguez-Molina, Jesús, et al. 2013.** Combining Wireless Sensor Networks and Semantic Middleware for an Internet of Things-Based Sportsman/Woman Monitoring Application. *http://www.mdpi.com*. [Online] 1 24, 2013. [Cited: 2 16, 2013.] <http://www.mdpi.com/1424-8220/13/2/1787>. 14248220.
- Software Metrics to Estimate Software Quality using Software Component Reusability. Trivedi, Prakriti and Kumar, Rajeev. 2012.* 2, 2012, Vol. 9. 16940784.
- Toward a Comprehensive Framework for Evaluating the Core Integration Features of Enterprise Integration Middleware Technologies. Moradi, Hossein and Bahreininejad, Ardeshir. 2013.* 1, s.l. : Journal of Systems Integration, 2013, Vol. 4. 18042724.

***Authorization and Disclaimer***

Autores autorizan a LACCEI para publicar el artículo en las actas de congresos.