

# Desarrollo de aplicación y servicios web para consulta de bases de datos



*Miguel Ángel Rodríguez Leiva*

*Director: Pascual Pérez Blasco*



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



A mi familia, por su apoyo y comprensión,  
y porque de no ser por ellos lo hubiese ido  
dejando “para mañana” una y otra vez.

A Pascual Pérez, por hacerlo todo tan fácil  
y estar siempre tan cerca de sus alumnos.  
No me equivoqué al elegir el tutor de mi proyecto.

A mis jefes y compañeros de la DTV, por darme la oportunidad  
de aprender tanto en tan poco tiempo, y en especial a José y Javi,  
por sus consejos y su infinita paciencia conmigo.

Y por último a mis amigos y a todas las personas con  
las que he compartido grandes momentos durante este  
último año. De no ser por vosotros hubiese terminado  
mucho antes, sí, pero también hubiese acabado desquiciado.



# TABLA DE CONTENIDOS

Agradecimientos .....	3
Tabla de contenidos .....	5
1. Introducción .....	7
2. Estado del arte .....	8
2.1. Tecnologías utilizadas .....	9
2.1.1. SQL Server .....	9
2.1.2. SQL .....	10
2.1.3. Active Directory .....	10
2.1.4. Microsoft Visual Studio .....	11
2.1.5. C# .....	11
2.1.6. ASP.NET .....	12
2.1.7. SOAP .....	12
2.1.8. XML .....	13
3. Objetivos .....	15
3.1. Objetivos de los servicios web .....	15
3.1.1. Servicio web para la base de datos del SIP .....	15
3.1.2. Servicio web para la base de datos TRIADA .....	15
3.2. Objetivos de la aplicación web .....	16
4. Cuestiones previas .....	17
4.1. Conocimientos previos de C# .....	17
4.2. Introducción a los servicios web .....	18
4.3. Elección del entorno de programación .....	19
5. Memoria del trabajo realizado .....	20
5.1. Resumen .....	20
5.2. Servicio web BuscaSipWebservice .....	21

5.2.1. Funciones de BuscaSipWebservice .....	21
5.3. Servicio web BuscaTriadaWebservice .....	24
5.3.1. Funciones de BuscaTriadaWebservice .....	25
5.4. Aplicación web BuscaSipTriada .....	29
5.4.1. Default.aspx .....	29
5.4.2. BuscaSIP.aspx .....	31
5.4.3. BuscaTriada.aspx .....	32
5.4.4. Resultado.aspx .....	36
5.4.5. Denied.aspx .....	38
6. Estructura del trabajo realizado .....	40
6.1. Estructura de los servidores .....	40
6.2. Estructura de los directorios .....	41
6.3. Diagrama de flujo de la aplicación web .....	42
6.3.1. Algoritmo de la aplicación web .....	43
7. Conclusiones .....	45
8. Futuras líneas de trabajo .....	47
9. Bibliografía .....	49
9.1. Referencias web .....	49
10. Anexos .....	50
101. Tabla de anexos .....	50

# 1. INTRODUCCIÓN

La Dirección Territorial de Valencia de la Conselleria de Sanitat (en adelante DTV) tiene acceso a la base de datos del Sistema de Información Poblacional (en adelante SIP) para obtener cierta información de utilidad en sus aplicaciones internas. Además, almacena y/o actualiza estos datos en una base de datos local llamada TRIADA para agilizar algunas operaciones y permitir una búsqueda más amplia de pacientes. Mientras a la base de datos TRIADA se le puede hacer consultas por cualquier campo de información que contengan sus tablas, la base de datos del SIP, de carácter autonómico, sólo permite consultar los datos del paciente a partir del número del SIP (único para cada paciente y que le identifica).

Actualmente estas dos consultas se realizan manualmente o mediante aplicaciones independientes destinadas a este propósito. Es por ello que se propone el desarrollo de sendos servicios web que permitan integrar estas consultas a futuras aplicaciones de la DTV. Además, resultará muy interesante que se consiga acceder a la información de la base de datos del SIP de pacientes de los cuales se desconoce el número del SIP, pero sí se conocen otros datos que aparecen en la base de datos TRIADA.

Finalmente, por su utilidad en una futura mejora de la intranet local de la DTV y también como método eventual para comprobar la correcta funcionalidad de ambos servicios web desarrollados, se pedirá además la implementación de una aplicación web que haga uso de los servicios web anteriormente descritos.

Al tratarse de una aplicación interna de la Dirección Territorial de una Conselleria, y especialmente por el tipo de información privada con que se trabaja, será requisito indispensable la implementación de un sistema de seguridad totalmente fiable y que en cada momento registre la persona y el ordenador que están haciendo uso de la aplicación, permitiendo el acceso únicamente a ordenadores y usuarios a los que se les haya asignado los privilegios correspondientes.

## 2. ESTADO DEL ARTE

La DTV utiliza en la gran mayoría de sus aplicaciones internas información obtenida de la base de datos del SIP. Para ello, en la actualidad, a la hora de desarrollar cada programa, se recurre a un método POST que envía al servidor de la base de datos del SIP una sentencia XML con la consulta del número de SIP pertinente a la base de datos, y recibe mediante un método GET otra sentencia XML que contiene la información del paciente consultado. Estos métodos han de integrarse nuevamente en cada aplicación que se implementa, adaptándolos al lenguaje de programación utilizado y creando nuevas funciones que los ejecuten. No obstante, donde realmente radica la dificultad de estas llamadas es en la clasificación y el procesado de los datos adquiridos. La respuesta enviada por el centro de datos del SIP se recibe en una sentencia XML cuyo corpus contendrá tantos nodos como apartados de información se tengan de dicho paciente de en la base de datos. A su vez, cada apartado se dividirá en sub-apartados que contendrán la información específica de cada uno de estos. Recae sobre la aplicación el tratamiento de esta cadena y su volcado en el tipo de contenedor de datos que se elija, siendo ésta una de las fases más laboriosas y delicadas de la adquisición de los datos provenientes del SIP. Más aún si cabe si para cada aplicación distinta que se desarrolle, hay que escoger la manera de clasificar todos estos datos.

En el grupo de trabajo INTAI de la DTV se empezó a trabajar en el desarrollo de una librería que proporcionase las funciones necesarias para ejecutar este tipo de consultas, ahorrando la programación específica de estas funciones en cada futura aplicación de escritorio de la DTV. No obstante, el proyecto está ahora mismo parado y a la espera de ser reanudado en el futuro, dando mayor prioridad a otros proyectos más urgentes.

De esta situación actual que se expone, se desprende la importancia del desarrollo de un servicio web que solucione la tarea de tener que programar cada vez las funciones de consulta a la base de datos del SIP.

En el caso de TRIADA, actualmente no existe ningún método que posibilite la consulta a su base de datos desde otra aplicación. Sólo los administradores del servidor *cipres*, en el que se hospeda la base de datos, pueden acceder a ella mediante la aplicación *SQL Server* de *Windows* y ejecutando consultas en lenguaje SQL sobre las tablas de información.



Es por esto que el disponer de una herramienta que permitiese búsquedas en la base de datos TRIADA sería de gran utilidad en las futuras aplicaciones internas de la DTV. En este caso, un servicio web similar al que se desarrollaría para la base de datos del SIP, podría integrarse fácilmente en cualquiera de estas futuras aplicaciones, bien sean de escritorio o aplicaciones web (cada vez más utilizadas estas últimas). Se permitiría así un acceso a los datos de pacientes mucho más efectivo que si sólo dispusiéramos de las consultas a la base de datos del SIP, pues cabe recordar que actualmente sólo se puede acceder a los datos de un paciente en esta base de datos por su número del SIP, muchas veces desconocido.

De estas dos propuestas, surge la idea de combinar ambos servicios para disponer de una herramienta que permita: primero, utilizar la base de datos TRIADA para obtener el número del SIP de un paciente del cual se desconoce ese dato concreto pero sí se conocen otros que le identifican; en segundo lugar, ejecutar una consulta sobre la base de datos del SIP con el recién adquirido número del SIP; y finalmente, obtener todos los datos del paciente para utilizarlos según convenga.

Adicionalmente, cabe destacar que es también el grupo de trabajo INTAI el que se dedica al mantenimiento diario de la base de datos interna TRIADA. Por todo esto, el flujo de información con el responsable y los trabajadores de este grupo será muy importante, ya que el constante intercambio de datos y sobre todo de ideas será pieza clave en el desarrollo del proyecto.

Paralelamente a este proyecto, se está trabajando en la DTV en un nuevo portal en *SharePoint* de *Windows* que funcionará a modo de intranet. Es por ello que se pide el desarrollo de estas herramientas como servicios web, con vistas a incluir su funcionalidad en futuras aplicaciones web, además de aplicaciones de escritorio.

## **2.1. Tecnologías utilizadas**

En este punto se pretende describir brevemente las tecnologías empleadas a lo largo del desarrollo del proyecto, bien sean lenguajes de programación o herramientas para la gestión de bases de datos.

### **2.1.1. SQL Server**

SQL Server es el servidor profesional para la gestión de bases de datos de Microsoft y está basado en el modelo relacional. Se instala únicamente sobre sistemas Windows y permite ejecutar todas las acciones típicas de los servidores profesionales de bases

de datos, como son: transacciones, control de concurrencia, conexiones remotas, diferentes instancias, varias cuentas de usuario, etc.

Dispone de una herramienta gráfica similar a la empleada por Microsoft en su aplicación Access, que permite trabajar de una manera más rápida y sencilla evitando las complejas instrucciones mediante línea de comandos.

Los lenguajes que utiliza para las consultas sobre la base de datos son *T-SQL* y *ANSI SQL*, siendo este último el que utilizaremos nosotros.

### 2.1.2. SQL

*SQL* son las siglas en inglés de *Structured Query Language*, o lo que es lo mismo, lenguaje estructurado de consultas. Es un lenguaje formal declarativo que se utiliza para manipular información de una base de datos, y en la actualidad es el lenguaje estándar para la gran mayoría de sistemas de gestión de bases de datos.

Si bien *SQL* nos permite no sólo consultar, sino también manipular la información de nuestra base de datos, éste no será el caso de nuestra aplicación, ya que además de no contar con los permisos necesarios para llevar a cabo este tipo de operaciones sobre las bases de datos con las que trabajaremos, nuestra aplicación únicamente pretende obtener información a partir de unos datos introducidos, y no alterar los registros que se tienen almacenados en las bases de datos.

Para obtener la información requerida, es necesario construir una sentencia *select* en lenguaje *SQL* donde describimos el tipo de información que queremos obtener, y el servidor *SQL* buscará y nos devolverá todas las instancias en la base de datos que cumplan todas las condiciones.

La versión utilizada en nuestro proyecto será *SQL Server 2005*.

### 2.1.3. Active Directory

El Directorio Activo es “un componente central de la plataforma Windows que proporciona los medios para gestionar las identidades y relaciones que organizan los entornos de red” [*W<sup>3</sup>ACTDIR*]. Mediante distintos protocolos (*LDAP*, *DNS*, *DHCP*, *kerberos*, ...) jerarquiza una red distribuida como lo puede ser la de una empresa relacionando los diferentes componentes de la red: usuarios, grupos de usuarios, permisos, etc. *Active Directory* identifica los dominios y subdominios utilizando la misma notación de las zonas *DNS*, y por ello requiere uno o más servidores *DNS* que

permitan el direccionamiento de los componentes lógicos y los elementos pertenecientes a la red, como por ejemplo el listado de usuarios y el de equipos conectados a la misma. Además, diferentes dominios pueden relacionarse mediante una relación de “*trust*” (o “confianza”) permitiendo que los distintos usuarios y recursos sean visibles entre ellos.

En nuestro caso, *Active Directory* nos servirá para crear grupos de usuarios con los permisos pertinentes para el uso de cada uno de los elementos del proyecto. De esta forma, podremos elegir los usuarios que tienen acceso a cada servicio web o a cada parte de la aplicación web, comprobando sus credenciales y denegando el acceso en caso negativo.

#### **2.1.4. Microsoft Visual Studio**

Visual Studio es el IDE (Entorno de Desarrollo Integrado) desarrollado por Microsoft para sistemas operativos Windows. Por defecto soporta varios lenguajes de programación como son las versiones *visual* de C++, C#, J#, ASP y Basic; aunque existen extensiones para poder integrar muchos otros.

Esta herramienta permite a los desarrolladores crear aplicaciones tanto de escritorio como web, así como sitios y servicios web gracias a la plataforma .NET.

En nuestro caso utilizaremos Visual Studio para desarrollar la mayor parte de nuestro proyecto, trabajando con C# y ASP.NET como lenguajes de programación y de diseño respectivamente.

La versión utilizada en nuestro proyecto será Microsoft Visual Studio 2005.

#### **2.1.5. C#**

C#, o *C Sharp*, es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. [*W<sup>3</sup>CSHARP*]

Se puede argumentar pues que C# corresponde a un salto evolutivo sobre su precursor C++ (el nombre de C# proviene de dibujar otros dos signos positivos encima de los dos signos positivos de C++).

En el caso de nuestro proyecto, trabajaremos con la versión de Visual C# que incluye el paquete básico de nuestro entorno de desarrollo Visual Studio 2005.

### 2.1.6. ASP.NET

ASP.NET es la estructura de software desarrollada por Microsoft para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web XML, convirtiéndose en la sucesora de la tecnología ASP (*Active Server Pages*).

Las páginas diseñadas mediante ASP.NET, conocidas como *web-forms* o formularios web, están contenidas en archivos con extensión *.aspx* que contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo *Controles Web* que se procesan del lado del servidor y *Controles de Usuario* donde los desarrolladores colocan todo el código estático y dinámico requerido por la web. Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<% -- código dinámico -- %>`, aunque generalmente se aconseja no hacerlo de esta manera, y sí utilizar el modelo *code behind*, o código de respaldo. Este modelo *code behind* coloca el código en un archivo separado o en una etiqueta de *script* especialmente diseñada. Cuando se usa este estilo de programación, el desarrollador escribe el código correspondiente a diferentes eventos, como la carga de la página, o el clic en un control, en vez de un recorrido lineal a través del documento. En teoría, esto permite a un diseñador web, por ejemplo, enfocarse en la creación del diseño con menos posibilidades de alterar el código de programación mientras lo hace. [W<sup>3</sup>ASPNET]

En nuestro caso, utilizaremos formularios web ASP.NET para el diseño de las páginas web de las que consta nuestra aplicación, optando por la opción del *code behind* con C# como lenguaje de programación para las funciones dinámicas de los formularios web.

### 2.1.7. SOAP

El acrónimo SOAP corresponde a las siglas en inglés de *Simple Object Access Protocol*, que significa Protocolo de Acceso a Objetos Simples. Es un protocolo que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Fue creado por Microsoft, IBM y otros, y actualmente ostenta la condición de estándar auspiciado por el W3C (*World Wide Web Consortium*).

Es el protocolo más utilizado por los servicios web y se utiliza por ejemplo para el envío de correos electrónicos.

En nuestro caso, será utilizado para hacer las llamadas necesarias desde nuestra aplicación web a los dos servicios web que hemos desarrollado, siendo la vía también por la cual estos servicios web nos enviarán la información solicitada.

### 2.1.8. XML

Y finalmente llegamos al lenguaje que nos permite enlazar las distintas partes que conforman nuestro proyecto.

El acrónimo XML deriva de las siglas en inglés de *Extensive Markup Language* (Lenguaje de Marcas Extensibles), y es un metalenguaje extensible de etiquetas desarrollado por el W3C. Es una simplificación y adaptación del SGML (*Standard Generalized Markup Language* o Lenguaje de Mercado Generalizado) y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Esto lo logramos mediante las etiquetas, que consisten en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma **<nombre>**, donde *nombre* es el nombre del elemento que se está señalando. [W<sup>3</sup>XML]

En nuestro caso utilizamos XML para comunicarnos con los servicios web, pero también para estructurar la información que recibimos, como por ejemplo para organizar los datos obtenidos de un paciente en diferentes secciones y celdas.

## **3. OBJETIVOS**

En esta sección se pretende explicar los diferentes objetivos y motivaciones de las diferentes partes que conforman el proyecto, resultando la suma de estos el objetivo final del proyecto en sí.

### **3.1. Objetivos de los servicios web**

El objetivo principal de los servicios web a desarrollar es el de proporcionar a los futuros programadores de aplicaciones internas una herramienta que les permita obtener de una manera rápida y sencilla los datos que necesiten de las bases de datos con que se trabaja en la DTV. De esta manera se evitará tener que implementar funciones específicas para este cometido en cada uno de los nuevos programas, ahorrando tiempo y complicaciones en cuanto a cómo conectar a las bases de datos y consultar sus tablas, además de la manera en la que recibir y clasificar esta información.

Los servicios web se hospedarán inicialmente en el servidor destinado a pruebas DTV CERES, desde donde atenderán a las peticiones que podrán provenir. Previamente, el servicio web tendrá que cerciorarse que el usuario que consulta la base de datos está autorizado a ello y en caso negativo no proporcionará dicha información. Para ello contaremos con la herramienta *Active Directory*.

#### **3.1.1. Servicio web para la base de datos del SIP**

En el caso concreto del servicio web que consulta la base de datos del SIP, únicamente se recibirán peticiones por número de SIP. Con este dato se consultará la base de datos del SIP, y entonces se devolverá la información obtenida al usuario que ha ejecutado la llamada al servicio web.

#### **3.1.2. Servicio web para la base de datos TRIADA**

En este segundo caso, el desarrollo del servicio web que consulta la base de datos TRIADA se antoja más laborioso, ya que aquí no sólo se permitirá la búsqueda por el número de SIP (único para cada paciente), sino que además se podrán hacer consultas por diferentes campos que TRIADA, al ser una base de datos privada de la DTV, sí muestra a los usuarios autorizados. Inicialmente estos serán Nombre, Apellidos, DNI y

número de SIP, aunque en función del desarrollo del servicio web se podrán contemplar cambios con respecto a esta decisión.

### **3.2. Objetivos de la aplicación web**

Como complemento a los servicios web se pide el desarrollo de una aplicación web que funcione a modo de buscador y que haga uso de ambos servicios. Si se conoce el número SIP del paciente, la aplicación web deberá consultar directamente la base de datos del SIP y mostrar la información obtenida. No obstante, en el más que probable caso de que no se conozca el número SIP del paciente pero sí otros datos como pueda ser el DNI o su nombre y apellidos, la aplicación web ha de consultar en primera instancia la base de datos TRIADA, mostrando todas las coincidencias que encuentre (si es que hay alguna). Posteriormente será el usuario quien escoja cuál de las coincidencias mostradas se corresponde al paciente que buscaba, la aplicación web obtendrá de los datos de TRIADA el número SIP del paciente, y ahora sí podrá consultar a la base de datos del SIP para finalmente mostrar la información solicitada. Al igual que los servicios web, la aplicación web también comprobará previamente en el *Active Directory* que el usuario tiene permisos para acceder a cada sección de la aplicación y denegará el acceso de no ser así.

El diseño de la interfaz de la aplicación deberá ser lo más amigable posible, con vistas a ser integrada en la nueva intranet de la DTV en la que se está trabajando y que se estima estará acabada para cuando el desarrollo de este proyecto llegue a su fase final. Los usuarios potenciales de esta aplicación serán los trabajadores de la sección de informática, así como su grupo de trabajo paralelo INTAI, pero en un futuro podría extenderse a ciertos cargos de la DTV que carecen de una formación específica en informática, por lo que la sencillez y amigabilidad de la aplicación serán pieza clave de su éxito como aplicación interna y no sólo como elemento de prueba de los servicios web.



## 4. CUESTIONES PREVIAS

En primer lugar, nos encontramos ante la tesitura de no haber trabajado nunca con el lenguaje de programación que se exige en el proyecto, siento éste C#. Si bien sabemos que su sintaxis deriva de C y C++, el modelo de objetos de la plataforma .NET que utiliza se asemeja más al de Java, y por tanto se puede argumentar que aún en un mismo lenguaje las dos características más interesantes de cada uno de los lenguajes (además de incluir mejoras derivadas también de otros lenguajes como Delphi) [*W<sup>3</sup>CSHARP*].

Además, será también necesario buscar información sobre cómo trabajan los servicios web, y cuál será la mejor manera de utilizarlos en este caso en concreto.

Por todo esto, será necesaria una primera fase de investigación en la cual comencemos a familiarizarnos con este nuevo lenguaje y a dar nuestros primeros pasos en el desarrollo de servicios web.

### 4.1. Conocimientos previos de C#

El primer paso que daremos será el de encontrar y completar un más que necesario tutorial sobre C#, que de una manera rápida y sencilla nos muestre todos los detalles específicos del lenguaje y nos permita mediante ejemplos y ejercicios familiarizarnos con el lenguaje que vamos a trabajar durante los siguientes meses. No se pretende en ningún caso convertirse en unas semanas en un experto programador en C#. En primer lugar, conocemos los lenguajes en los cuales se basa tanto su sintaxis como su modelo de objetos, y por lo tanto deberíamos hacernos una idea de lo que nos puede ofrecer C#. Además, el devenir del propio proyecto será quien nos indique de una manera u otra en qué aspectos de la programación nos tenemos que centrar, y así qué características del lenguaje tenemos que llegar a dominar para sacar el proyecto adelante.

En este caso se escogió el **Curso de iniciación a la programación con C#**, de la web de ElGuille [*W<sup>3</sup>GUILLE*]. La decisión de elegir este pequeño curso se basó, en primer lugar, en que era un curso on-line y además gratuito, distribuido en trece entregas y con ejercicios para resolver al final de cada capítulo sobre el tema tratado y revisando lo aprendido en capítulos anteriores. Pero sobre todo, porque ya conocíamos de antemano este sitio web, habiéndonos sido útil en otras ocasiones, lo que nos aportaba una confianza extra que a la postre no nos decepcionó.

Durante la familiarización (puesto que no se puede denominar ni siquiera aprendizaje), nos congratuló observar que el lenguaje en cuestión, como prometía, no se trataba ni más ni menos que de una especie de actualización del más que utilizado C++, adaptando todas las mejoras propias de Java, que tanto facilitan y agilizan la programación orientada a objetos.

Una vez completado el tutorial y habiendo resuelto de manera notable los ejercicios propuestos, podemos afrontar con confianza el desarrollo de cualquier proyecto en C#.

## **4.2. Introducción a los servicios web**

En cuanto a los servicios web, primero que nada tendremos que informarnos sobre las posibilidades que nos ofrecen, y después elegiremos la mejor solución para nuestro caso concreto.

No existe una definición clara sobre lo que son los servicios web, pero según la oficina española del World Wide Web Consortium, *“Una posible sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios.”* [W<sup>3</sup>SERWEB]. Los servicios web permiten, mediante mecanismos de comunicación estándares, que diferentes aplicaciones interactúen entre sí. De esta manera se consigue presentar información dinámica al usuario.

Para entender el concepto mejor pensemos en el caso de la página web de una agencia de viajes que permite a sus clientes reservar sus vacaciones on-line. El usuario accede a la página web y simplemente seleccionando el destino, la fecha de llegada y el número de noches desea conocer cuánto le costará el viaje. Es tarea pues de la web de la agencia de viajes consultar a sus asociados: a la aerolínea el precio de los billetes, y al hotel el precio de la habitación por tal número de noches. Y el proceso no es otro que mediante los servicios web. En el momento en que el usuario consulta a la web de la agencia de viajes, ésta consulta la información pertinente a cada uno de los servicios web de sus asociados, procesando los datos obtenidos y mostrándoselos al cliente. De hecho, la propia web de la agencia de viajes está siendo consultada y mostrando los resultados al cliente mediante su propio servicio web. Adicionalmente, en caso de que el cliente deseara efectuar la compra del paquete vacacional, la web de la agencia de viajes utilizará otro servicio web que interactuará con su banco para validar la venta con los datos de la tarjeta de crédito introducidos por el cliente. Con este ejemplo se observa claramente la utilidad de los servicios web en la actualidad, permitiendo el

flujo de datos entre aplicaciones (e incluso tecnologías) distintas de manera automática.

### **4.3. Elección del Entorno de Programación**

Siendo C# el lenguaje de programación que vamos a emplear para el desarrollo de los servicios web, hemos decidido que también será este lenguaje el que utilizaremos para la programación de los scripts (*code behind* o código de respaldo) de la aplicación web. Para ello utilizaremos el entorno de programación *Visual Studio 2005* de *Microsoft*, ya que además de estar ya instalado en los ordenadores de la sección de Informática de la DTV, es un entorno que nos es familiar puesto que ya hemos trabajado con él anteriormente.

Por lo tanto, ya tenemos decidido que tanto los servicios web como el *code behind* de las páginas web se programarán en C#. Ahora falta elegir la manera en la que desarrollaremos las páginas de la aplicación web, y teniendo en cuenta que vamos a trabajar con *Visual Studio 2005*, la elección lógica es aprovechar el *framework* para aplicaciones web ASP.NET (sucesor del ASP clásico) que ya incluye la aplicación y que además permite la creación de sitios webs dinámicos y servicios web XML, que es justo lo que necesitamos.

Una vez decididas todas estas cuestiones previas, podemos ponernos manos a la obra con el desarrollo del proyecto en sí.

## 5. MEMORIA DEL TRABAJO REALIZADO

En esta sección describiremos uno a uno todos los pasos seguidos en el desarrollo del proyecto, profundizando en cada una de las partes que conforman tanto la aplicación como los servicios web.

### 5.1. Resumen

Se ha desarrollado un sistema totalmente independiente de cualquier otra aplicación ya existente en la DTV que permite hacer consultas a las bases de datos del SIP y TRIADA para obtener cierta información almacenada sobre pacientes del servicio sanitario de la Comunidad Valenciana. El sistema consta de sendos servicios web (uno para cada base de datos) y una aplicación web que actúa como interfaz para el uso de los mismos. Además se ha implementado un control de seguridad basado en *Active Directory* que garantiza un uso restringido tanto de la aplicación web como de los servicios web, evitando que los datos puedan llegar a manos no deseadas.

La aplicación web y los dos servicios web se han alojado de manera temporal en el servidor de pruebas de la Dirección Territorial DTV CERES, pudiendo ser trasladados a cualquier otro servidor de la DTV en el futuro sin que esto afecte a su funcionalidad.

Los servicios web se espera que sean integrados en futuras aplicaciones internas de la DTV, permitiendo estos un gran avance en cuanto a la consulta de sus respectivas bases de datos y no siendo necesario reprogramar estas consultas en cada aplicación que se desarrolle. Mientras tanto, y gracias a la aplicación web diseñada, se pueden utilizar para hacer consultas independientes de una manera rápida y sencilla, sin tener que acceder a las tablas de las bases de datos en cuestión para el caso de TRIADA, o tener que utilizar programas externos (y por tanto limitados) para conseguir cierta información de la base de datos del SIP, como se hacía anteriormente. Actualmente sólo se puede acceder a la aplicación web conociendo su ruta en el servidor DTV CERES, pero una vez pasada con éxito la fase de pruebas, es tarea del grupo de informáticos de la DTV integrarla en la intranet corporativa si procede. En este caso, la aplicación sólo será visible para el personal autorizado a consultar dichas bases de datos, ya que la intranet también hace uso de *Active Directory* para comprobar las credenciales de cada usuario que accede a la misma.

Con todo esto, y combinando consultas a ambas bases de datos, ahora se pueden consultar los datos de un paciente no sólo por su número de SIP como era la única

opción, si no que se puede encontrar al mismo gracias a otros datos como son su nombre, apellidos, DNI, y fecha de nacimiento<sup>1</sup>.

Pasamos ahora a describir cada una de las partes que conforman el proyecto, explicando para cada una cómo se han desarrollado y cómo funcionan e interactúan entre sí.

## 5.2. Servicio web *BuscaSipWebservice*

El servicio web *BuscaSIPwebservice.asmx* es el encargado de consultar la base de datos del Servicio de Información Poblacional. A esta base de datos sólo se le pueden hacer consultas por el número de SIP, recibiendo en formato XML la respuesta con la información almacenada sobre el paciente consultado. Anteriormente sólo se podía consultar esta información mediante la inclusión de un engorroso método POST en alguna función de la aplicación que se estuviera desarrollando. Gracias a este servicio web que integra la llamada al servicio web remoto de la base de datos del SIP con el número de SIP que se le proporcione, se puede añadir la consulta a esta base de datos en cualquier nueva aplicación que se programe de manera rápida y sencilla.

El servicio web diseñado necesita recibir tres atributos para poder efectuar la consulta, siendo estos:

**SIP:** Número del SIP del paciente del cual se quiere obtener la información.

**usuario:** Login del usuario que va a ejecutar la consulta.

**contraseña:** Password del usuario que va a ejecutar la consulta.

El login y el password se piden a efectos de seguridad, ya que el servicio web verificará en primer lugar que el usuario que intenta acceder a la información está autorizado a consultar esta base de datos.

### 5.2.1. Funciones de *BuscaSipWebservice*

El servicio web está programado en ASP.NET con C# como lenguaje de programación del código de soporte. Consta de una función básica y otras tres auxiliares privadas, pasamos a describirlas a continuación.

---

<sup>1</sup> Más tarde se verá que se pueden hacer búsquedas también por año de nacimiento, o incluso por mes y año de nacimiento, sin ser un requisito conocer el día exacto.

### ***IsAuthenticated***

Función del tipo `bool` que comprueba que un usuario está autenticado en el Active Directory.

Parámetros que recibe:

`string` `srvr`: Ruta del servidor que hospeda el Active Directory.

`string` `usr`: Login del usuario a autenticar.

`string` `pwd`: Password del usuario a autenticar.

La función conecta con el servidor del Active Directory e intenta autenticarse con las credenciales aportadas. En función del éxito, devuelve un valor `true` o `false` en una variable booleana.

### ***peticionXML***

Función del tipo `string` que, a partir de un número de SIP dado, completa la petición en formato XML que se enviará al servicio web remoto de la base de datos del SIP para pedir la consulta. Esta petición incluye el número de SIP que identifica al paciente, así como los bloques de datos que se solicitan.

Parámetros que recibe:

`string` `nsip`: Cadena que contiene el número SIP del paciente a consultar.

Esta función no hace otra cosa que introducir la cadena dada en el lugar oportuno del texto en formato XML que espera recibir el servicio web remoto del SIP. Esta petición XML se conoce porque es la que utilizaba la DTV en sus anteriores aplicaciones para formular el método POST. Finalmente devuelve una cadena con la nueva petición XML, ya con el número de SIP insertado en el lugar oportuno.

### ***TestPeticion***

Función del tipo `string` que conecta con el servicio web remoto de la base de datos del SIP y envía la petición que anteriormente hemos compuesto para recibir y volcar en un documento XML la información del paciente solicitada.

Parámetros que recibe:

`string` `peticion`: Petición en XML que incluye el número de SIP y los bloques de información solicitados.

La función primero codifica la petición XML recibida y ejecuta un `HttpRequest` a la dirección `http://sip.san.gva.es/pls/tarjeta/wusu.directo_web`. Si las credenciales aportadas en la petición son correctas, el servicio web remoto del SIP nos devolverá vía `HttpResponse` la información solicitada, y nuestra función volcará este contenido en un documento XML debidamente formateado. La cadena XML interna de este documento será la que se devuelva en una variable `string`.

### ***busca\_SIP***

Por fin llegamos a la función principal de este servicio web. Esta función serializa todas las demás, amén de incluir otros detalles como comprobar que el usuario está autorizado a hacer uso de esta herramienta, o también guardar en un documento de texto un log de accesos y consultas.

Parámetros que recibe:

`string` `SIP`: Número del SIP del paciente del cual se quiere obtener la información.

`string` `usuario`: Login del usuario a autenticar.

`string` `contrasena`: Password del usuario a autenticar.

En primer lugar, y por propósitos de seguridad, consulta y almacena la dirección IP desde la cual se está accediendo al servicio web. El paso siguiente es autenticar al usuario, y comprobar que éste tiene autorización para consultar la base de datos del SIP. Esto se consigue llamando a la función `IsAuthenticated()` y, en el caso de estar autenticado, se verificará también que el usuario pertenece al grupo *BuscaSip*, lo que nos indicaría que se le han concedido privilegios para poder acceder a esta base de datos. Si estas dos condiciones se cumplen, no existe motivo por el cual no se tenga que proceder a la consulta de la base de datos del SIP, primero creando la petición con la función `peticionXML()`, y luego obteniendo la respuesta XML mediante `TestPeticion()`.

A la hora de hacer la consulta, se almacenará en el fichero “.\Logs\BuscaSipWS\_Log.txt” los datos de la consulta realizada (i.e. Fecha y hora de la consulta, dirección IP, usuario, y número de SIP consultado). En caso de que no se llegue a realizar la consulta por algún tipo de error, siendo los más comunes que el usuario no esté autenticado o no tenga autorización, también se almacenarán en el log los datos de los que se disponga del intento de consulta, de manera llamativa para que los encargados del mantenimiento de la aplicación puedan darse cuenta e investigar esta situación si es menester.

### 5.3. Servicio web *BuscaTriadaWebservice*

El servicio web *BuscaTRIADAwbservice.aspx* es el encargado de consultar la base de datos TRIADA, privada de la DTV. Esta base de datos está alojada en el servidor *cipres* de la DTV, y anteriormente sólo existía una manera de obtener información de ella, que era logueándose en el servidor, accediendo la base de datos mediante el *Microsoft SQL Server* y formulando consultas en SQL utilizando la herramienta que proporciona la misma aplicación. No hace falta pues explicar lo complejo que se hacía consultar unos simples datos de un paciente, requiriendo conocimientos de SQL que ya de por sí excluyen a casi cualquier empleado de la DTV que no pertenezca al grupo de Informática. Además, no se podía confiar en que algún empleado inexperto alterase la base de datos de alguna forma, estando pues la base de datos TRIADA fuera del alcance de todos los operarios de la DTV, exceptuando los informáticos que trabajan en su mantenimiento.

Gracias al servicio web diseñado a tal efecto, ahora es posible consultar la base de datos TRIADA con una sencilla combinación de campos, siendo el propio servicio web el encargado de componer la sentencia SQL que se utilizará para efectuar la consulta. La respuesta obtenida de la base de datos, se formateará a modo de documento XML para facilitar la clasificación de la información de la manera que requiera cada futura aplicación que haga uso del servicio web, pudiéndose dar el caso que para los campos rellenados, exista más de una coincidencia en la base de datos.

Los campos de la base de datos TRIADA que se pueden consultar y que por tanto también se muestran son los siguientes<sup>2</sup>: número del SIP, DNI, nombre, apellidos y

---

<sup>2</sup> La base de datos TRIADA contiene, para cada paciente, muchos más datos de los que será necesario mostrar para la identificación de un usuario. Por el tipo de información privada que se maneja hay que tener mucha delicadeza a la hora de elegir qué datos se muestran y cuáles se mantienen ocultos. En cualquier caso, los campos que se han elegido son suficientemente significativos como para poder encontrar a un paciente sin riesgo a equivocarnos.



fecha de nacimiento. Además se mostrará, pero no se podrá consultar, el número de la seguridad social.

El servicio web diseñado podrá recibir entre tres y diez atributos para poder efectuar la búsqueda, siendo dos de ellos siempre el **usuario** y la **contraseña** del usuario que va a ejecutar la consulta. Los otros ocho atributos, y de los cuales al menos habrá que aportar uno (si no se permitiría un volcado completo de todos los pacientes), son los siguientes: **número del SIP, DNI, nombre, primer apellido, segundo apellido, día de nacimiento, mes de nacimiento y año de nacimiento.**

El login y el password, como en el caso del anterior servicio web, se piden a efectos de seguridad, ya que éste verificará en primer lugar que el usuario que intenta acceder a la información está autorizado a consultar esta base de datos.

### **5.3.1. Funciones de BuscaTriadaWebservice**

El servicio web está programado en ASP.NET con C# como lenguaje de programación del código de soporte. Consta de una función básica y otras dos auxiliares privadas, pasamos a describirlas a continuación.

#### ***IsAuthenticated***

Función del tipo `bool` que comprueba que un usuario está autenticado en el Active Directory.

Parámetros que recibe:

`string` `srvr`: Ruta del servidor que hospeda el Active Directory.

`string` `usr`: Login del usuario a autenticar.

`string` `pwd`: Password del usuario a autenticar.

Es la misma función utilizada en *BuscaSIPwebservice*. La función conecta con el servidor del Active Directory e intenta autenticarse con las credenciales aportadas. En función del éxito, devuelve un valor `true` o `false` en una variable booleana.

#### ***letra\_nif***

Función del tipo `string` que calcula para un DNI dado, la letra del NIF asociada a ese DNI.

Parámetros que recibe:

`UInt32 dni`: Dígitos del DNI en forma de entero sin signo.

La función calcula mediante un sencillo algoritmo  $[W^3ALGNIF]$  la letra del NIF que corresponde al DNI dado, y devuelve en forma de cadena el NIF completo resultante de unir los dígitos del DNI con la letra del NIF obtenida.

### ***busca\_TRIADA***

Y llegamos por fin a la función principal de este servicio web. Esta función serializa todas las demás, amén de incluir otros detalles como comprobar que el usuario está autorizado a hacer uso de esta herramienta, o también guardar en un documento de texto un log de accesos y consultas.

Parámetros que recibe:

`string sip`: Número del SIP del paciente del cual se quiere obtener la información.

`string dni`: DNI del paciente del cual se quiere obtener la información.

`string nombre`: Nombre del paciente del cual se quiere obtener la información.

`string apellido1`: Primer apellido del paciente del cual se quiere obtener la información.

`string apellido2`: Segundo apellido del SIP del paciente del cual se quiere obtener la información.

`string dia`: Día de nacimiento del paciente del cual se quiere obtener la información.

`string mes`: Mes de nacimiento del paciente del cual se quiere obtener la información.

`string anyo`: Año de nacimiento del paciente del cual se quiere obtener la información.

`string usuario`: Login del usuario a autenticar.

`string contrasena`: Password del usuario a autenticar.

En primer lugar, y por propósitos de seguridad, consulta y almacena la dirección IP desde la cual se está accediendo al servicio web. El paso siguiente es autenticar al usuario, y comprobar que éste tiene autorización para consultar la base de datos del

SIP. Esto se consigue llamando a la función `IsAuthenticated()` y, en el caso de estar autenticado, se verificará también que el usuario pertenece al grupo *BuscaTriada*, lo que nos indicaría que se le han concedido privilegios para poder acceder a esta base de datos. Si estas dos condiciones se cumplen, no existe motivo por el cual no se tenga que proceder a la consulta de la base de datos TRIADA.

Para poder conectar con la base de datos, deberá conocer los parámetros de la conexión. En un primer lugar estos formaban parte del código fuente, pero a petición de los encargados del Área de Informática se decidió almacenarlos en el fichero “.\Conexion\conexion.xml” de la carpeta *PruebasSip* que contiene el servicio web en el servidor *dtvceres*. De esta manera, si en algún momento se muda la base de datos a otra ubicación, o incluso si se cambia el nombre de usuario y la contraseña del servicio web periódicamente por motivos de seguridad, sólo habrá que modificar el fichero *conexion.xml* y el servicio web seguirá funcionando con toda normalidad. Siendo así, el servicio web abre el archivo XML y recorre los nodos para obtener estos parámetros, que son:

**datasource:** Servidor donde se encuentra hospedada la base de datos.

**initialcatalog:** Nombre de la base de datos a consultar.

**userid:** Login exclusivo para el servicio web.

**Password:** Password del servicio web.

Una vez obtenidos estos datos y compuesta la cadena de conexión, se aborda la tarea de crear la sentencia *SELECT* en SQL que represente la búsqueda por los campos introducidos en la llamada al servicio web. Una particularidad que cabe destacar sobre la sentencia *SELECT* es la manera en que se buscarán las coincidencias en la base de datos. Para los casos de la búsqueda por número de SIP y/o DNI se procederá a una búsqueda exacta; sin embargo, para el caso del nombre y los apellidos, se permite una búsqueda parcial. Esto es, que aprovechando el comparador *LIKE* del lenguaje SQL y el caracter comodín “%”, se pueden efectuar búsquedas como por ejemplo “El primer apellido empieza por Martin~” (pudiendo ser éste: Martín, Martínez, Martino, ...) o “Cuyo nombre contiene Francisco” (pudiendo ser éste: Francisco, José Francisco, Francisco Javier, ...). Para el caso de los apellidos, será necesario introducir manualmente el símbolo “%” en el lugar oportuno (e.g. “Martin%” para el caso anterior), no obstante, teniendo en cuenta el grandísimo número de nombres compuestos utilizados en nuestro país, y que además estos suelen omitirse o abreviarse en el ámbito coloquial, se ha decidido incluir ya en el código los caracteres comodín al principio y al final de cada búsqueda por nombre. De esta manera, cada vez

que busquemos por ejemplo “Javier”, el mismo programa automatizará la búsqueda parcial para mostrarnos, además de los *Javier*, también los *José Javier*, *Javier Félix*, *Carlos Javier*, etc.

El caso de la fecha de nacimiento es seguramente el más complejo. Si conocemos la fecha de nacimiento completa del paciente será muy fácil efectuar la consulta. Así pues, también podemos consultar indicando sólo el año de nacimiento, o incluso el año y el mes. No obstante, y por motivos de limitaciones a la hora de acotar las fechas, el sistema no es capaz de reconocer fechas en las cuales se indique un día y un año de nacimiento, pero no un mes. Los tipos de datos *date* en las bases de datos en SQL son muy particulares, y si bien aportan muchas ventajas a la hora de la búsqueda y ordenación de datos, no permiten búsquedas parciales de ese tipo, habiéndose tenido que diseñar una solución específica para este caso que nos permite las búsquedas anteriormente citadas. Por este motivo el propio servicio web ignorará este tipo de casos en que no se especifica un mes pero sí un día, consultando sólo por el año de nacimiento e ignorando el día en los casos que se pueda.

Con la cadena de conexión y la sentencia *SELECT* en SQL de la búsqueda ya formadas, podemos proceder a conectar con la base de datos y enviarle nuestra consulta. La respuesta la almacenaremos en un documento XML agrupando, primero por coincidencias (recordemos que más de un paciente pueden ser resultado de una búsqueda), y luego para cada coincidencia organizaremos los distintos campos en sub-nodos. Este documento XML es el que finalmente devuelve el servicio web. Hay que enfatizar el caso de que en la base de datos TRIADA no se encuentre ninguna coincidencia para los campos consultados, de ser así el documento XML contendrá un aviso en consecuencia.

A la hora de hacer la consulta, se almacenará en el fichero “.Logs\BuscaTriadaWS\_Log.txt “ los datos de la consulta realizada (i.e. Fecha y hora de la consulta, dirección IP, usuario, y la sentencia *SELECT* consultada). En caso de que no se llegue a realizar la consulta por algún tipo de error, siendo los más comunes que el usuario no esté autenticado o no tenga autorización, también se almacenarán en el log los datos de los que se disponga del intento de consulta, de manera llamativa para que los encargados del mantenimiento de la aplicación puedan darse cuenta e investigar esta situación si es menester.

## 5.4. Aplicación web BuscaSipTriada

Finalmente, pasamos a describir paso por paso cómo funciona la aplicación web desarrollada para hacer uso de ambos servicios web, combinando las ventajas de cada uno para poder obtener los datos de casi cualquier paciente dado de alta en el Servicio de Información Poblacional.

Esta aplicación web permite que, una vez logueado y autenticado en el *Active Directory* de la DTV, un usuario ejecute las consultas necesarias contra las bases de datos a las que esté autorizado a consultar. Mostrándole los resultados de una manera ordenada y organizada, de manera que el usuario pueda fácilmente encontrar la información que necesita.

En primer lugar, aparecerá una página de identificación donde el usuario introducirá su login y password de su cuenta de la DTV y seleccionará la base de datos que desea consultar, SIP o TRIADA. Sus credenciales serán verificadas y si el usuario está autorizado para esa base de datos, podrá continuar hasta la siguiente página. De no ser así, será redirigido a una página de error donde se le informará de que no tiene los privilegios requeridos para hacer uso de esa aplicación, y sólo se le permitirá volver a la página principal. Una vez en la página BuscaSIP o BuscaTRIADA, el usuario podrá rellenar los campos de búsqueda que serán los que se enviarán al servicio web, y si la consulta es satisfactoria, se le mostrará el resultado obtenido.

Todo esto se ha llevado a cabo tratando que la interfaz sea lo más amigable posible con vistas a una futura integración en la intranet corporativa, intentando también que el diseño y los colores encajen con la misma.

Pero mejor será que pasemos a describir una por una las diferentes páginas web que componen la aplicación, explicando para cada una de ellas los diferentes casos que se pueden dar y cómo se ha programado el *code behind* que les da soporte.

### 5.4.1. Default.aspx

En la **figura 4.1** podemos ver la primera página con la que se encontrará el usuario que tenga pensado hacer uso de la aplicación web.

Como se puede apreciar, al usuario se le solicita su login y su password, para que después seleccione alguno de los dos motores de búsqueda disponibles.



**BUSCA**

GENERALITAT VALENCIANA  
CONSELLERIA DE SANITAT

**TRIADA**

INTRODUZCA SU LOGIN Y SU PASSWORD PARA AUTENTICARSE

Login:

Password:

SELECCIONE EL MOTOR DE BÚSQUEDA

GENERALITAT VALENCIANA  
CONSELLERIA DE SANITAT

**TRIADA**

Figura 5.1: "Default.aspx"

El código de respaldo de la página consta de tres sencillas funciones tan básicas que ni siquiera será necesario entrar en demasiados detalles. Existen dos funciones, una para cada imagen-botón, que no hacen otra cosa si no, en primer lugar asegurarse de que tanto el cuadro de texto del login como el del password han sido rellenados, emitiendo un mensaje de aviso en caso contrario; y en segundo lugar redirigiendo el control de la aplicación web a la siguiente página, *BuscaSip.aspx* o *BuscaTriada.aspx*, según sea el caso, enviando vía contexto de la web las credenciales introducidas. Puede sorprender que se avance a la siguiente página sin haber verificado antes en el *Active Directory* que se tengan los permisos necesarios, no obstante, ahora se verá en la explicación de ambos buscadores que esto no es más que un truco para evitar repetir estas consultas posteriormente agilizando así el continuo tráfico de datos que ya de por sí soporta el servidor *baladre*.

Cabe mencionar también la función `Page_Load()` que es la encargada de dar título a la página web, además de situar el foco del cursor de texto en la casilla del login, típicamente la primera en ser teclada.

### 5.4.2. BuscaSIP.aspx

Lo primero que ha de hacer esta página<sup>3</sup> es comprobar que el usuario logueado tiene permiso para acceder a la misma. Para ello, se conectará al *Active Directory* con las credenciales introducidas en la página *Default.aspx* y de no ser así, antes de que la página cargue ninguno de sus componentes, se redirigirá al usuario a la página de error *Denied.aspx*.

En el caso de que el usuario sí tenga permisos para consultar la base de datos del SIP, la página web que se le mostrará será la de la **figura 4.2**.



**Figura 5.2: “BuscaSip.aspx”**

Recordemos que para el caso de la base de datos del SIP, sólo se pueden hacer consultas por número de SIP, con lo cual será éste el único campo que podremos rellenar antes de hacer clic en el botón “*Buscar*”.

<sup>3</sup> Siempre que se habla de las funciones de una página hay que entender que estas funciones pertenecen al *code behind* de la página escrito en C#, y no a la página web en sí que no es más que una serie de componentes en ASP.NET (texto, imágenes, botones, cuadros de texto, etc) dispuestos según un formato elegido.

En cuanto se refiere al *code behind* de la página web, contiene cuatro funciones, una de ellas siendo la misma `IsAuthenticated(string srvr, string usr, string pwd)` que ya conocemos de los servicios web.

Al cargar la página, se ejecuta la función `Page_Load()`, que en primer lugar consulta mediante `IsAuthenticated()` que las credenciales que recuperamos del contexto se corresponden con un usuario autenticado en el *Active Directory* y que además pertenece al grupo *BuscaSip*. De no ser así, y antes de cargar nada más, se redirigirá al usuario a la página de error ***Denied.aspx***, como ya se avanzó anteriormente. Si en cambio el usuario sí que tiene permiso para acceder a esta base de datos, se cargarán los componentes de la página web, dando título a la misma y colocando el foco del cursor de texto en la casilla del número de SIP.

Una vez el usuario introduzca el número de SIP que desea consultar, bien pulsando el botón “*Buscar*”, o bien dándole a la tecla intro “↵” (cosa que conseguimos mediante la función `txtSip2_TextChanged()`), se ejecutará la función `Button1_Click()`, que contiene la llamada al servicio web *BuscaSIPwebservice*, además de almacenar en el fichero “`.\Logs\BuscaSip_Log.txt`” los datos de la consulta realizada (i.e. Fecha y hora de la consulta, dirección IP, usuario, y número de SIP consultado). Destacar que se ha incluido un validador que no permite la llamada al servicio web si el número de SIP introducido no se ajusta al formato estandarizado del mismo, en este caso una cadena de entre 4 y 7 dígitos. Eso se consigue en *ASP.NET* gracias al uso de validadores de expresiones regulares, que son condiciones que deben cumplirse para que la página web permita la llamada a funciones del *code behind*, siendo para este caso concreto la expresión regular **`[0-9]{4,7}`**.

La respuesta obtenida de la base de datos del SIP, que recordemos llegará en forma de documento XML, será enviada por contexto a una nueva página web ***Resultado.aspx***, donde, como veremos en el más adelante **punto 5.4.4**, se nos mostrará la información de una manera sencilla y organizada. En caso de no obtener respuesta, bien sea porque el número del SIP no se encuentra en la base de datos o porque exista algún problema temporal en la conexión, se informará de ello y quedará constancia de la búsqueda en el log de la aplicación “`.\Logs\BuscaSip_Log.txt`”.

### 5.4.3. *BuscaTriada.aspx*

Esta página será, seguramente, el eslabón principal de la gran mayoría de búsquedas mediante esta aplicación. Como ya hemos comentado anteriormente, el



caso más habitual no va a ser una búsqueda por número de SIP, normalmente desconocido (a veces incluso por el propio paciente), si no más bien una búsqueda por nombre y apellidos o DNI, que son datos mucho más comunes. El DNI tiene la ventaja de ser único e identificativo, sin embargo no es un dato que siempre se conozca. En ese caso, tendríamos que recurrir a una búsqueda más abierta basándonos en los datos no únicos que conociéramos, pudiendo ser estos el nombre, los apellidos, o la fecha de nacimiento del paciente. Por todo esto, el servicio web *BuscaTRIADAwbservice* y esta página web que hace uso del mismo conforman la parte más importante de la aplicación, permitiéndonos a base de completar campos ir acotando nuestra búsqueda hasta dar con el paciente buscado, para después, consultar su información completa gracias al servicio web *BuscaSIPwbservice*.

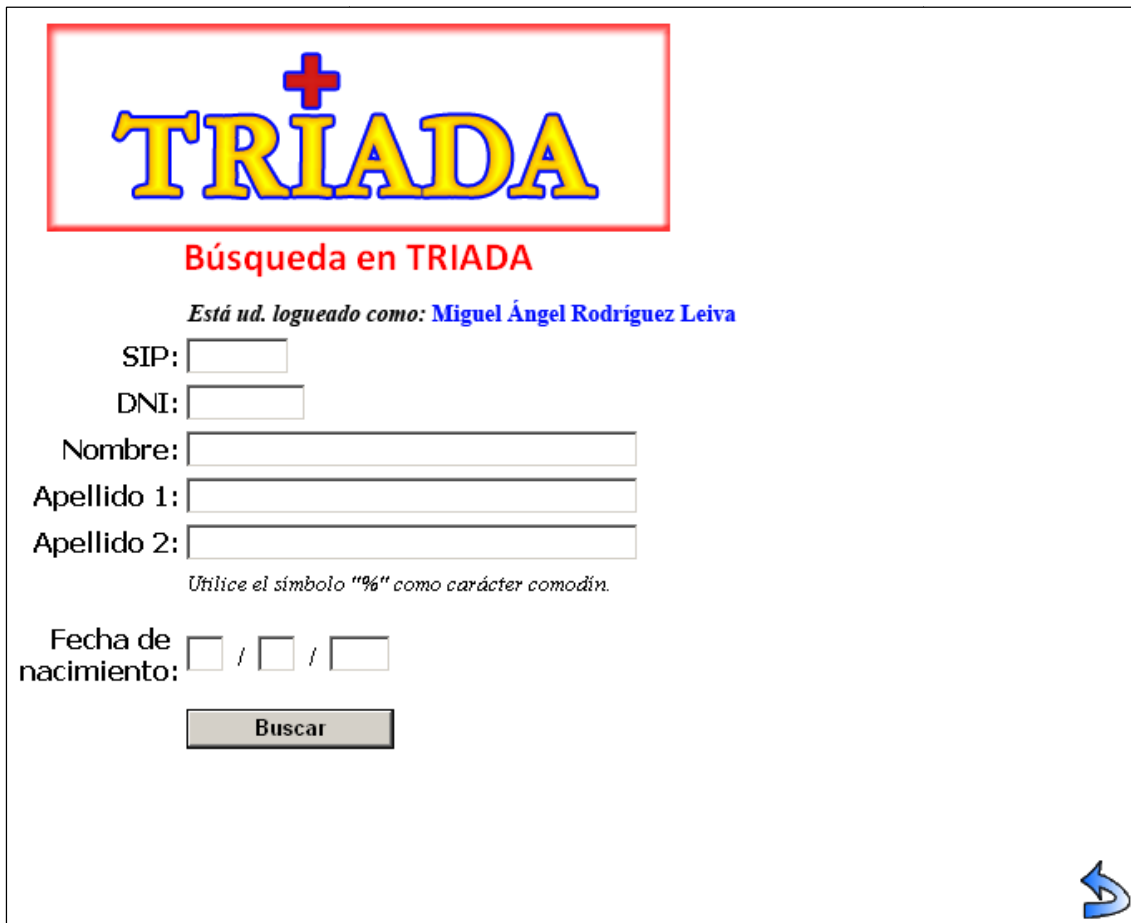


Figura 5.3: "BuscaTriada.aspx"

Como vemos en la **figura 4.3** la página se asemeja mucho a *BuscaSip.aspx*, con el logo del motor de búsqueda, la identificación del usuario logueado, los campos por los cuales se puede buscar (que en este caso son más de uno) el botón que ejecuta la llamada al servicio web.

Al igual que en **BuscaSip.aspx**, de lo primero que se encarga el *code behind* de la página es de comprobar que el usuario cuyas credenciales recuperamos del contexto está autenticado en el *Active Directory* y también que pertenezca al grupo *BuscaTriada*. Esto se consigue una vez más mediante la función `IsAuthenticated(string srvr, string usr, string pwd)`. Si el usuario no está autorizado a acceder a esta base de datos, inmediatamente se le redirigirá a la página de error **Denied.aspx** donde se le advierte que no puede hacer uso de esta aplicación. En el caso de que las credenciales sean las correctas, la función `Page_Load()` continúa su ejecución de manera normal mostrando todos los elementos que hemos descrito en el párrafo anterior.

Ahora ya podemos hacer uso de la búsqueda en Triada. Para ello, sólo tendremos que rellenar los campos que conozcamos del paciente y hacer click sobre el botón “*Buscar*” (o bien dar a la tecla intro “↵”). Esto lanza la ejecución de la función `Button1_Click()`, pero para que esta llamada se efectúe, y como en el caso de **BuscaSip.aspx**, primeramente tendrán que cumplirse las condiciones especificadas en los validadores de expresiones regulares asociados a cada campo de búsqueda. Son los siguientes:

**Número de SIP:** Tendrá que ser una cadena de entre 4 y 7 dígitos (`[0-9]{4,7}`).

**DNI:** Tendrá que ser una cadena de 7 dígitos, pudiendo ser el primero también una X o una Y para el caso de pacientes con DNI extranjero (`[X|x|Y|y|0-9][0-9]{7}`).

**Nombre:** Podrá ser un texto con espacios con la única restricción de que los números no serán permitidos (`[\\D]*`).

**Apellido 1:** Mismo caso que para **Nombre**.

**Apellido 2:** Mismo caso que para **Nombre** y **Apellido 1**.

**Día de nacimiento:** 1 ó 2 números.


**Mes de nacimiento:** 1 ó 2 números.

**Año de nacimiento:** 4 números.

**Nota:** Será el *code behind* de la página quien se encargue de comprobar que la fecha de nacimiento introducida se corresponde con una fecha real, teniendo en cuenta tanto los días que tiene cada mes, como las peculiaridades de los años bisiestos.

En el caso de que todos los campos rellenados (por supuesto se pueden dejar en blanco) cumplan las condiciones indicadas para cada uno de ellos, y siempre que al menos uno contenga algún dato, se procederá a la llamada al servicio web

*BuscaTRIADAwbservice*, enviándole los campos de la consulta para que nuestro servicio web se encargue de formular la sentencia SQL que nos devolverá en formato XML la respuesta obtenida. Recordemos que esta respuesta puede contener información sobre más de un paciente, puesto que búsquedas por apellidos o fecha de nacimiento van a obtener más de una coincidencia. Toda esta información, organizada en XML por nodos y sub-nodos, se va a volcar en una tabla que aparecerá en esta misma página debajo de la zona del buscador. En la tabla se muestra, organizada por páginas de como mucho 50 resultados, la información de todos estos pacientes: número de SIP, NIF, nombre, apellidos, fecha de nacimiento y número de la seguridad social. Esta tabla *GridView* es un componente propio de *ASP.NET* que nos es muy útil en este caso para poder mostrar toda esta información, ya que permite además de como ya hemos comentado paginar por un número máximo de filas, sobre todo nos da la opción de ordenar los resultados por la columna que indiquemos, tanto en orden ascendente como descendente, y tan solo haciendo clic en el título de la columna por la cual queremos ordenar. Esta manera tan intuitiva de ordenar la información obtenida de la base de datos TRIADA es sin duda la clave de la amigabilidad y funcionalidad de esta parte de la aplicación, y por extensión de la aplicación entera, ya que si para una búsqueda que devuelve digamos 5.000 usuarios, tuviéramos que ir buscando nuestro paciente página por página o en una interminable lista, la aplicación carecería totalmente de utilidad. Pudiendo ordenar por columnas como se puede, es mucho más fácil encontrar al paciente en cuestión bien sea porque sabemos en qué rango de edad se encuentra o que tiene un hermano mayor cuyo nombre también conocemos.

Una vez encontrado el paciente que buscábamos, y si la información que aparece en la tabla no nos es suficiente, a partir de su número de SIP y mediante nuestro servicio web *BuscaSIPwbservice* podemos consultar la base de datos del SIP. Podríamos ahora pues apuntar su número de SIP que nos muestra la tabla, volver al menú principal de la aplicación, acceder a la página de búsqueda en el SIP y consultar el número de SIP apuntado. No obstante, y teniendo en cuenta que este caso en el que una vez localizado un paciente en TRIADA vamos a necesitar consultar toda su información en la base de datos del SIP, se ha integrado esta búsqueda también en la tabla de la aplicación *BuscaTRIADA*. Para cada coincidencia en la base de datos TRIADA, o lo que es lo mismo, en cada fila de la tabla resultado, se ha integrado una pasarela al servicio web *BuscaSIPwbservice* en forma de icono incrustado en cada celda de la primera columna de cada fila de la tabla. Este icono , que representa la tarjeta del SIP que todos los usuarios del Servicio de Información Poblacional poseen, será el botón que ejecutará la consulta a la base de datos del SIP con el número de SIP del paciente en cuya fila de la tabla se encuentra el icono. Es decir, que si yo localizo al

paciente que estaba buscando, bastará con hacer clic sobre el icono de la tarjeta del SIP que se encuentra en la misma fila que todos sus datos y la función `GridView1_SelectedIndexChanging1()` se encargará de hacer la llamada al servicio web que consulta la base de datos del SIP y nos devolverá toda la información almacenada sobre este paciente, que recordemos llegará en forma de documento XML, enviada por contexto a la nueva página web **Resultado.aspx** que veremos a continuación en el **punto 5.4.4**.

Como en todos los casos, aquí también se guarda en un fichero de texto todas las consultas y/o intentos de acceso no autorizados, siendo este fichero `“.\Logs\BuscaTriada_Log.txt”`.

#### 5.4.4. Resultado.aspx

Ésta es la página web a la que redirigen tanto **BuscaSip.aspx** como **BuscaTriada.aspx** siempre que se consulta la base de datos del SIP y ésta devuelve la información de un paciente.

The screenshot displays the 'Resultado.aspx' web page. At the top, there is a header for 'GENERALITAT VALENCIANA' and 'SIP CONSELLERIA DE SANITAT'. Below the header, there are input fields for 'Número SIP:', 'Número Seguridad Social:', and 'CIP:'. The main content area is divided into three sections: 'Datos del nacimiento' (with fields for 'Fecha de nacimiento:' and 'Lugar de nacimiento:'), 'Datos de la residencia' (with fields for 'Domicilio:' and 'Teléfono(s):'), and 'Clave Médica:' (with fields for 'Hospital Asignado:', 'Centro Médico Asignado:', 'Médico de Cabecera:', and 'Especialidad:'). At the bottom right, there is a field for 'Última actualización:'. A logo for 'CÓDIGO XML' is visible at the bottom center.

Figura 5.4: “Ejemplo de Resultado.aspx”

Recordemos que la información se recibe en forma de documento XML, así que para mostrar la misma de una manera clara y ordenada hemos diseñado una presentación en forma de ficha sencilla pero vistosa, que muestra la información más

relevante del paciente organizada en diferentes secciones según el carácter de la misma. Un ejemplo de ficha que contiene la información de un paciente sería la mostrada en la **figura 4.4**.


Como se puede apreciar<sup>4</sup>, los datos de interés del paciente se agrupan en tres secciones: En primer lugar se muestra el nombre, y los números únicos que le identifican, siendo estos el NIF, el número de SIP, el número de la Seguridad Social (pudiendo ser éste compartido por padres e hijos) y el código CIP. En segundo lugar, recuadrado en azul, encontramos los datos propios de la persona, como son sus datos de nacimiento y los datos de su última residencia registrada. Tras esta información, en un recuadro rojo, se muestra la información médica del paciente, como es su clave médica, y sus centros y médico asignados. Finalmente, pero no por ello menos importante, se muestra la fecha de la última actualización de los datos de este paciente en la base de datos del SIP. Este dato puede significar bien que la información es fidedigna, puesto que se ha actualizado recientemente, como también puede significar que algunos datos (principalmente los que se refieren a la actualidad) puedan ponerse en entredicho y sea necesaria una revisión de los mismos.

Toda esta información se obtiene del documento XML, viniendo ésta organizada en nodos y sub-nodos, y para mostrarla de esta manera tan amigable es necesario que la función `Page_Load()` se encargue de *parsearla* recorriendo todos los nodos del documento XML con un `XPathNavigator` y su propiedad `SelectSingleNode()`, que nos permite dado un camino de nodos, leer la información del nodo destino y volcarla en una variable del tipo `string`. Veamos el caso por ejemplo del código postal: Una vez estudiado y comprendido el documento XML que devuelve la base de datos del SIP para cada consulta exitosa, hemos encontrado que el código postal de la residencia actual del paciente se encuentra en el nodo **msg\_respuesta** → **usu** → **loca** → **dirRes** → **copo** . Esto se traduce en que para acceder a tal información, tendremos que “navegar” hasta el nodo “**copo**” pasando por los demás que nos encontramos en su ruta, para luego leer su información y almacenarla en un `string`. De esta manera se obtienen todos los datos relevantes del paciente y se formatean para ser mostrados a modo de ficha médica en la página actual. Cabe la posibilidad, y de hecho es bastante común, que algunos nodos no contengan la información requerida, significando esto que no se conoce esta información. En ese caso, para dejar constancia de este hecho y así además no alterar el diseño de la ficha, el lugar donde debería mostrarse esta información quedará vacío.

---

<sup>4</sup> Por cuestiones de privacidad se ha aplicado un filtro de distorsión sobre los datos privados del paciente del ejemplo. No obstante, esto no es impedimento para imaginar el resultado de una búsqueda real con la aplicación web.

Se ha desarrollado también la función `string formatea_fecha(string fecha_orig)` para formatear las fechas de formato “YYYY-MM-DD” a “DD de MM de YYYY”, que se ha considerado más propio de una ficha médica.

Sin embargo, los datos extraídos del código XML y mostrados en la ficha médica no son los únicos que se reciben de la base de datos del SIP, incluyéndose además de estos muchos otros relativos a la cobertura del seguro contratado o a información de carácter todavía más privado y que escapan al desarrollo de este proyecto. Es por ello que, para el caso de que fuese necesario consultar el código XML recibido (y que recordemos es el texto con el que se estaba trabajando hasta la implantación del servicio web diseñado para este proyecto), se ha añadido al final de la ficha médica un botón “Código XML” () que permite mostrar u ocultar el documento XML que se recibe en un cuadro de texto a continuación de la información mostrada. Este botón no es otra cosa que un conmutador del estado de la visibilidad del cuadro de texto que contiene el mensaje devuelto por la base de datos del SIP, por defecto no visible. Esto se hace posible mediante la función `ImageButton1_Click()`.

#### 5.4.5. Denied.aspx

Ésta es la web a la que redirige *Default.aspx* por medio de la comprobación que se ejecuta al principio de la carga de *BuscaSip.aspx* y *BuscaTriada.aspx*.



Figura 5.5: “Denied.aspx”

Recordemos que la primera misión de las dos páginas web con motor de búsqueda es la de verificar que el usuario está autorizado a consultar esa base de datos. Al efectuar dicha comprobación, si el usuario no está autenticado en el *Active Directory*, o lo está pero no pertenece al grupo de usuarios con permisos para consultar dicha base de datos, la página web detendrá su carga y le redirigirá en cambio a una página web como la de la **figura 4.5**.

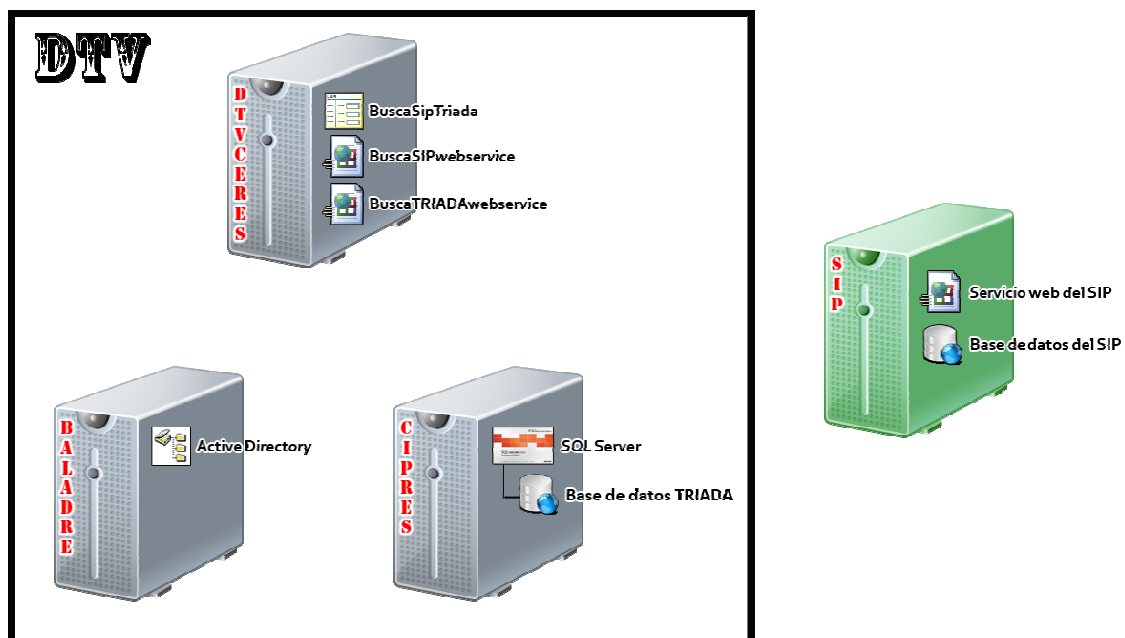
Esta página advierte de la situación y únicamente permite regresar a la página principal, frustrando cualquier intento no autorizado de hacer uso de la aplicación. Cabe destacar que no es necesario registrar en ninguna parte estos intentos ya que es la propia página web que verifica el acceso y lo deniega redirigiendo a ésta la que almacena en su propio log los datos del intento de uso de la aplicación.

## 6. ESTRUCTURA DEL TRABAJO REALIZADO

Vamos a describir de una forma más gráfica dónde se localizan cada una de las partes que conforman todo el proyecto, así como la manera en que interactúan unas con otras.

### 6.1. Estructura de los servidores

Desde el punto de vista del hospedaje, en la **figura 5.1** se muestran los diferentes servidores que participan en el desarrollo del proyecto, y qué partes del mismo contiene cada uno.



**Figura 6.1: “Diagrama de la estructura de los servidores”**

Como se puede observar, hacemos uso de cuatro servidores en total: tres de ellos privados de la Dirección Territorial, y un cuarto perteneciente al Servicio de Información Poblacional. Este último es el que contiene la base de datos del SIP y un servicio web que nos permite consultarla.

En cuanto a los tres servidores de la DTV, tenemos el servidor de pruebas *DTVCERES*, que es donde se aloja la aplicación web y los servicios web desarrollados. Además hacemos uso de los servidores *BALADRE*, que contiene el *Active Directory* con los datos de los usuarios, grupos y permisos de la DTV; y *CIPRES*, que únicamente se



utiliza para almacenar bases de datos en un Microsoft SQL Server y que nos permitirá acceder a la base de datos TRIADA.

## 6.2. Estructura de los directorios

Será importante también explicar la estructura en directorios de la aplicación web y los servicios web hospedados en *DTVCERES*, de manera que se pueda localizar fácilmente cada uno de sus elementos:

La carpeta que contiene tanto la aplicación web como el resto de archivos que conforman esta parte del proyecto tiene como nombre **PruebasSip**, y a su vez contiene tres subcarpetas más: **bin**, **Conexión** y **Logs**.

En el directorio principal **PruebasSip** hallamos los archivos propios de cada página de la aplicación web desarrollados mediante el Microsoft Visual Studio, todos ellos de extensión *.aspx*. Además, encontramos también ambos servicios web, **BuscaSIPwebservice.asmx** y **BuscaTRIADAwbservice.asmx**. Finalmente, podemos localizar aquí también las imágenes utilizadas en cada una de las páginas web del proyecto y el archivo de configuración *web.config*.

La carpeta **bin** la crea automáticamente el Microsoft Visual Studio a la hora de generar la aplicación, y contiene la copia compilada de cada una de las partes de la misma, tanto páginas como servicios web.

Dentro de **conexión** encontramos únicamente un archivo *.xml* del mismo nombre que contiene la información relativa a la conexión a la base de datos TRIADA. Recordemos que este fichero almacena en formato XML el servidor, el nombre del catálogo inicial de la base de datos y la dupla login/password para poder acceder a TRIADA, y es consultado antes de cada conexión para obtener esta información. De esta manera, si por algún motivo la ubicación de la base de datos cambia, sólo habrá que modificar los datos de este fichero para que la aplicación siga funcionando correctamente. Además, permite que la contraseña sea variada periódicamente mejorando la seguridad de la aplicación.

Finalmente encontramos la carpeta **Logs**, que como se puede deducir, almacena una bitácora por cada acceso realizado a alguna de las aplicaciones o servicios web. Toda consulta a las bases de datos es registrada, avisando también de los intentos maliciosos, y es en esta carpeta donde se almacenan estos registros. Pero esto no es todo, ya que para evitar que el tamaño de los logs aumente de manera peligrosa, se

optó por que cada primero de mes, los logs del mes anterior se compriman y se guarden en la carpeta **Backup**, creando unos nuevos para el mes que empieza. Esto lo hemos logrado con la ayuda del compresor **7-Zip**, que permite su ejecución en ventana de comandos y no necesita instalación, y mediante un script que hemos desarrollado para tal tarea y al que hemos llamado *ComprimeLogs.bat*. Este fichero está programado para ejecutarse el día 1 de cada mes, y realiza todas las tareas anteriormente explicadas.

En el **anexo B** podemos ver gráficamente cómo queda estructurado el directorio **PruebasSip** de DTVCERES.

### 6.3. Diagrama de flujo de la aplicación web

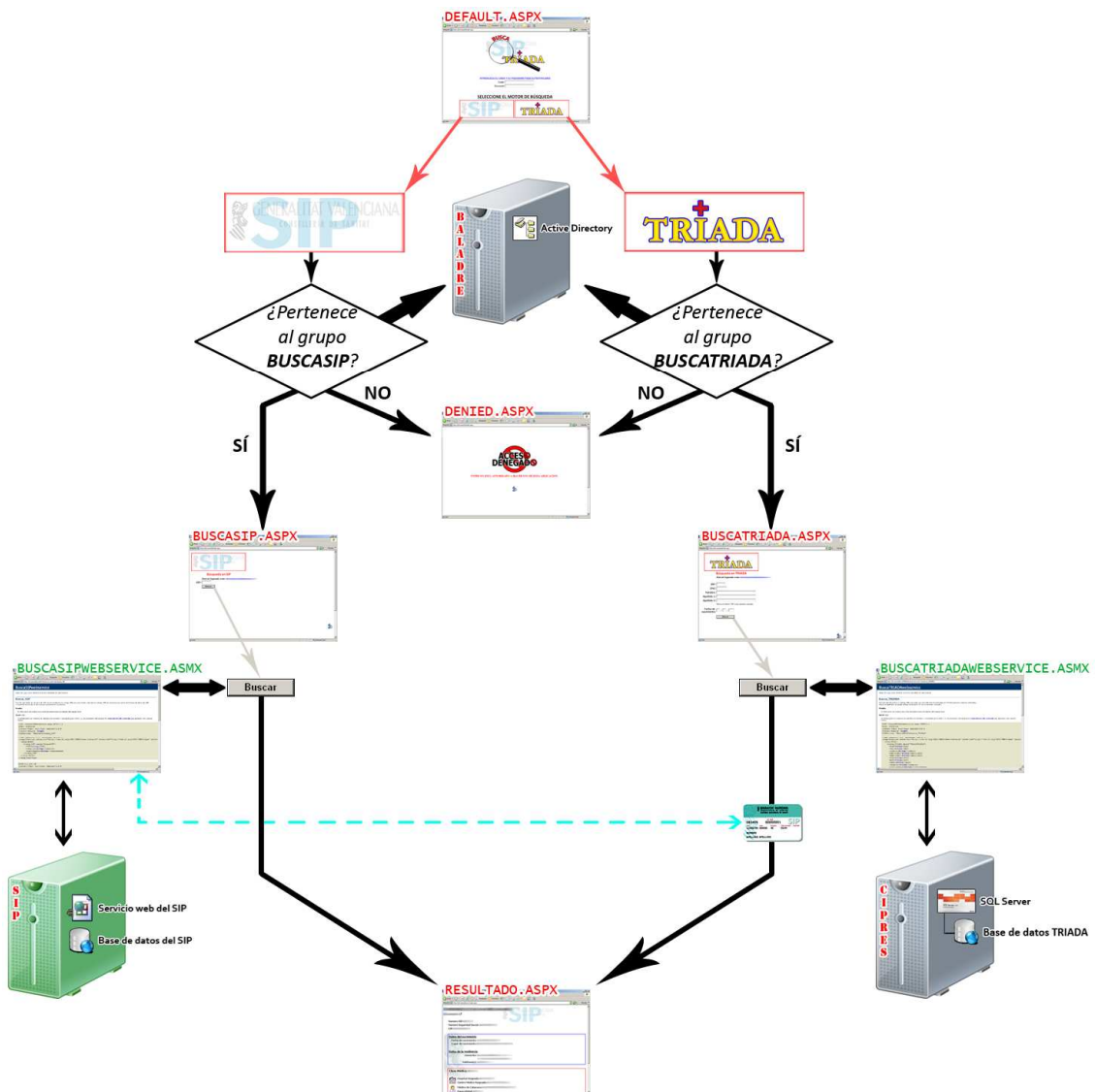


Figura 6.2: "Diagrama de flujo de la aplicación web"

Desde el punto de vista de la interacción de todas las partes que conforman el proyecto, o lo que es lo mismo, para explicar de una manera más clara los pasos que se siguen desde que un usuario accede a la página principal y hasta que obtiene el resultado final de su búsqueda (siempre que se cumplan todos los requisitos necesarios), se ha diseñado un diagrama que muestra los elementos más importantes del proyecto y los pasos que lleva a cabo la aplicación, así como el recorrido de los datos. En este caso se han obviado elementos secundarios como puedan ser retornos a la página inicial o incluso los logs que almacenan cada consulta a las diferentes aplicaciones y también servicios web.

El camino que recorrería la aplicación se puede apreciar en la **figura 5.2**. Si bien con este sencillo diagrama son ya fáciles de entender los diferentes pasos de la aplicación, seguidamente describimos el algoritmo en pseudocódigo que utiliza la aplicación:

### 6.3.1. Algoritmo de la aplicación web

- 1 Abrir la aplicación por su página principal *Default.aspx*
- 2 Seleccionar un motor de búsqueda:
  - Si seleccionamos **SIP**, ir a **3**
  - Si seleccionamos **TRIADA**, ir a **10**

---

- 3 La aplicación consulta el **Active Directory**:
  - Si no pertenece al grupo **BuscaSip**, ir a **4**
  - Si pertenece al grupo **BuscaSip**, ir a **5**
- 4 Se nos muestra la página *Denied.aspx* indicándonos que no podemos acceder a dicho servicio

---


- 5 Se nos muestra la página *BuscaSip.aspx*, introducimos el número SIP a consultar y **Buscar**
- 6 La aplicación llama al servicio web *BuscaSIPwebservice.asmx* enviándole el número SIP
- 7 El servicio web *BuscaSIPwebservice.asmx* consulta la base de datos del SIP y devuelve el resultado
- 8 *BuscaSip.aspx*, reenvía este resultado a la página *Resultado.aspx*
- 9 *Resultado.aspx* formatea el resultado obtenido y lo muestra en forma de ficha.

---

- 10 La aplicación consulta el **Active Directory**:
  - Si no pertenece al grupo **BuscaTriada**, ir a **4**
  - Si pertenece al grupo **BuscaTriada**, ir a **11**

---

- 11 Se nos muestra la página *BuscaTriada.aspx*, introducimos los campos a consultar y **Buscar**

- 12** La aplicación llama al servicio web *BuscaTRIADAwebservice.asmx* enviándole los datos introducidos
- 13** El servicio web *BuscaTRIADAwebservice.asmx* consulta la base de datos TRIADA y devuelve todas las coincidencias
- 14** *BuscaTriada.aspx* muestra en forma de tabla todas estas coincidencias
- 15** Si el usuario hace click sobre el botón del SIP (  ) la aplicación llama al servicio web *BuscaSIPwebservice.asmx* enviándole el número SIP de la coincidencia seleccionada
- 16** El servicio web *BuscaSIPwebservice.asmx* consulta la base de datos del SIP y devuelve el resultado
- 17** *BuscaSip.aspx*, reenvía este resultado a la página *Resultado.aspx*
- 18** *Resultado.aspx* formatea el resultado obtenido y lo muestra en forma de ficha.

## 7. CONCLUSIONES

En primer lugar cabe destacar que el conjunto de aplicaciones desarrollado tendrá en la Dirección Territorial de Valencia la importancia que desde el Departamento de Informática se le quiera dar. Esto viene a significar que el proyecto como tal no tendrá futuro si no se sigue una línea de trabajo que incluya estos avances en los que tanto empeño se ha puesto. Recordemos que si bien se ha culminado el proyecto con una amigable aplicación web que permite la consulta a la dos bases de datos, el objetivo principal del proyecto era el desarrollo de sendos servicios web que permitiesen de una forma segura y fiable este tipo de consultas, dado que era un escollo importante a la hora de la programación de distintas aplicaciones internas que tenían que hacer uso de estas bases de datos. Antes de que se desarrollasen estos dos servicios web, era necesaria una tediosa reprogramación de funciones similares que ya existían en diferentes aplicaciones internas para poder consultar la base de datos del SIP, mientras que no se disponía de manera alguna para acceder a los extremadamente útiles datos que contiene la base de datos TRIADA, por ser ésta de ámbito privado de la DTV y sin las restricciones propias de una base de datos externa. Naturalmente, estos problemas no serán subsanados hasta que por parte de los desarrolladores de software de la DTV haya un interés por entender cómo funcionan estos servicios web y descubran los importantes avances tanto en trabajo como en tiempo que ahorran a la hora del diseño de nuevas aplicaciones. En buena parte esto dependerá de las necesidades de actualización de las distintas aplicaciones que se utilizan en la DTV, ya que tampoco procede ahora reprogramar ciertas aplicaciones para integrar los servicios web desarrollados si esto no implica un importante beneficio.

Si consultásemos a los usuarios potenciales de futuras aplicaciones que vayan a aprovecharse de los servicios web desarrollados, esto es, a diferentes trabajadores de la DTV en su gran mayoría administrativos, seguramente no se darían ni cuenta del cambio en la manera de consultar los datos de un paciente mediante su número de SIP. En definitiva, para ellos sería el mismo tanto el proceso (introducir el número en la casilla correspondiente y hacer clic en el botón de “Buscar”) como el resultado (obtener los datos el paciente que contiene la base de datos del SIP). Serán los programadores quienes se vean directamente beneficiados al tener un servicio web que les proporciona estos datos, olvidándose de toda la programación intrínseca de las funciones que permiten obtener este resultado. No obstante, sí que será en lo que refiere al servicio web que consulta la base de datos interna TRIADA donde los usuarios notarán un importante avance para su trabajo diario. Y es que si antes sólo podían consultar los datos de un paciente si conocían su número del SIP, ahora podrán

también buscar al paciente por su DNI, o incluso por su nombre y apellidos. De hecho, habiendo consultado a diferentes empleados de la *Conselleria de Sanitat*, se confirma el deseo de estos de disponer de una herramienta que permitiera este tipo de consultas, siendo por ahora sólo posible en la DTV gracias a su base de datos TRIADA.

Para recapitular, diremos que el uso actual del proyecto desarrollado va a ser únicamente interno para los miembros del Departamento de Informática, siendo estos los únicos que tienen permiso para acceder tanto a los servicios web como a la aplicación web diseñada. Sin embargo, éste es un proyecto ambicioso y de futuro, y recaerá en estos mismos trabajadores el testeo del conjunto de aplicaciones y la decisión final de apostar por las diferentes partes del proyecto en el futuro de las aplicaciones de la Dirección Territorial.

## 8. FUTURAS LÍNEAS DE TRABAJO

Como se puede desprender de todo lo comentado anteriormente, éste es un proyecto pensado para su integración en futuras aplicaciones, y su éxito dependerá en buena parte del uso que se le dé y del interés que despierte en los programadores de la DTV. Las posibilidades que ofrecen los dos servicios web desarrollados son muy diversas, permitiendo su integración tanto en aplicaciones web como de escritorio, y no teniendo que preocuparse del lenguaje de programación escogido puesto que trabajan con métodos HTTP POST y lenguaje XML, compatibles con cualquiera de ellos.

En la actualidad, la mayoría de aplicaciones internas están programadas en Access, e incluso algunas futuras aplicaciones en las que se está trabajando también se valen de este lenguaje, ya que después de tantos años los empleados del Departamento de Informática de la DTV ya se pueden considerar unos expertos en el mismo. Aún así, los tiempos cambian, y se está haciendo un esfuerzo por cambiar esta dinámica y comenzar a implantar Java como futuro lenguaje de programación, así como dedicar más esfuerzos a la programación web, promoviendo la nueva intranet corporativa, pero sin descuidar tampoco la programación de aplicaciones de escritorio. Es por esto que los nuevos avances que proporciona este proyecto van a ir muy ligados a esta evolución, y las nuevas aplicaciones se beneficiarán de ellos.

En lo que se refiere al servicio web *BuscaSIPwebservice.asmx*, son ya muchas las aplicaciones que consultan la base de datos del SIP y esto seguirá siendo así por mucho tiempo. Es por ello que la ventaja que proporciona tener ya un servicio web que realiza todo el trabajo necesario de conexión y consulta al servicio web del Servicio de Información Poblacional será notoria en el momento de desarrollar nuevas aplicaciones para la DTV.

En cuanto al servicio web *BuscaTRIADAwbservice.asmx*, va a ser aquí donde especialmente se marque la diferencia. De ahora en adelante cualquier aplicación que se desarrolle va a disponer con tan solo la integración de este servicio web de una herramienta de consulta de la base de datos TRIADA. Recordemos que esta base de datos privada de la DTV almacena y se actualiza periódicamente con los datos de consultas al SIP, conformando una “réplica” de la misma, pero con las ventajas inherentes de poseer todos los derechos sobre la misma: esto es, poder componer cualquier sentencia *SELECT* que se nos antoje y así definir la búsqueda concreta que deseemos a partir de cualquiera de los campos que contiene la base de datos. Si pensamos en todas las ventajas que esto nos proporciona, ya sólo cabe esperar que los

programadores del Departamento de Informática de la DTV se pongan manos a la obra e imaginen nuevas opciones para su antiguo software, así como para nuevas aplicaciones. Habrá programas de ámbito general como por ejemplo el Abucasis, empleado en todas las delegaciones de la *Conselleria de Sanitat*, así como en hospitales y ambulatorios, que lógicamente no podrán ser mejorados puesto que éste es un proyecto interno de la DTV y su base de datos TRIADA también lo es. Pero son muchos otros los programas que utilizan los empleados, en su mayor parte administrativos, que sí podrán incluir esta opción y hacer mucho más rápido y sencillo su trabajo.

Cabe destacar, que no son ni uno ni otro servicio web la parte más importante del proyecto: El fuerte del proyecto es la interacción entre ambas. Aunando ambos servicios web conseguimos obtener la información actualizada de cualquier paciente que podamos localizar por sus diferentes datos personales. Ésta es la herramienta que se debería explotar, y con un poco de imaginación en cuanto a las restricciones típicas que conlleva el tratamiento de ciertos datos privados, se pueden desarrollar diversas y útiles herramientas que faciliten el trabajo a los empleados de la Dirección Territorial de Valencia de la *Conselleria de Sanitat*.

Para acabar, si nos referimos a la aplicación web, no sería descabellado incluir ésta en la nueva intranet corporativa de la DTV. Como se ha mencionado, tanto los servicios como las páginas web que consultan las bases de datos hacen primero una comprobación en el Active Directory para asegurarse de que el usuario que está intentando acceder las mismas tiene autorización para ello, no permitiéndolo en caso de no ser así. Teniendo en cuenta que la propia intranet ya se encarga de obtener las credenciales del usuario activo, no sería necesario que éste se identificase de nuevo con su login y password, y por ello se podría obviar la página principal accediendo directamente al buscador elegido si se tiene permiso para hacerlo. Para disponer de herramientas así se decide implantar una intranet, y es por ello que el diseño de la aplicación web se ha procurado que respete el estilo propio de la nueva intranet; para que cuando los responsables del Departamento de Informática lo estimen oportuno puedan incorporar de manera cómoda esta herramienta a las demás ya existentes en la intranet.



## 9. BIBLIOGRAFÍA

En el caso de este proyecto, y motivado por la cantidad de información que se encuentra actualmente en la red de redes y además de manera gratuita, todas las referencias corresponden a páginas web de carácter divulgativo.

### 9.1. REFERENCIAS WEB

[W<sup>3</sup>ACTDIR]: <http://www.microsoft.com/spain/windowserver2003/technologies/directory/activedirectory/default.aspx>

[W<sup>3</sup>CSHARP]: [http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp)

[W<sup>3</sup>ASPNET]: <http://es.wikipedia.org/wiki/ASP.NET>

[W<sup>3</sup>XML]: <http://es.wikipedia.org/wiki/XML>

[W<sup>3</sup>GUILLE]: <http://www.elguille.info/NET/cursocsharperik/index.htm>

[W<sup>3</sup>CSERWEB]: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

[W<sup>3</sup>ALGNIF]: [http://es.wikibooks.org/wiki/Algoritmo\\_para\\_obtener\\_la\\_letra\\_del\\_NIF](http://es.wikibooks.org/wiki/Algoritmo_para_obtener_la_letra_del_NIF)

## 10. ANEXOS

### 10.1. TABLA DE ANEXOS

#### **ANEXO A:**

##### ***Estructura de directorios de la carpeta «PruebasSip» en DTV CERES***

Diagrama con el contenido final de la carpeta «PruebasSip», directorio raíz de la aplicación web en DTV CERES.

#### **ANEXO B:**

##### ***Documentación de la aplicación web BuscaSIP-Triada***

Documento aportado a la Dirección Territorial de Valencia de la Conselleria de Sanitat a modo de manual de instrucciones de la aplicación web desarrollada. Útil para todo futuro usuario de la aplicación.

#### **ANEXO C:**

##### ***Documentación del servicio web BuscaSIPwebservice***

Documento aportado a la Dirección Territorial de Valencia de la Conselleria de Sanitat que contiene información relacionada con el servicio web BuscaSIPwebservice y que de manera breve puede servir a futuros desarrolladores para la inclusión de este servicio web en nuevas aplicaciones de la Dirección Territorial de Valencia.

#### **ANEXO D:**

##### ***Documentación del servicio web BuscaTRIADAwbservice***

Documento aportado a la Dirección Territorial de Valencia de la Conselleria de Sanitat que contiene información relacionada con el servicio web BuscaTRIADAwbservice y que de manera breve puede servir a futuros desarrolladores para la inclusión de este servicio web en nuevas aplicaciones de la Dirección Territorial de Valencia.

#### **ANEXO E:**

##### ***CD adjunto a la memoria***

CD que contiene tanto los ficheros empleados para la realización de este proyecto, como el resultado del mismo. Además, incluye una copia de esta memoria y demás material de información.

# ANEXO A

## ESTRUCTURA DE DIRECTORIOS DE LA CARPETA «PRUEBASSIP» EN DTVCERES



### PruebasSip



#### bin

- App\_WebReferences.compiled
- App\_WebReferences.dll
- App\_Web\_buscasip.aspx.cdcb7d2.dll
- App\_Web\_buscasipwebservice.asmx.cdcb7d2.dll
- App\_Web\_buscatriada.aspx.cdcb7d2.dll
- App\_Web\_buscatriadawebservice.asmx.cdcb7d2.dll
- App\_Web\_codigo.aspx.cdcb7d2.dll
- App\_Web\_default.aspx.cdcb7d2.dll
- App\_Web\_denied.aspx.cdcb7d2.dll
- App\_Web\_resultado.aspx.cdcb7d2.dll
- buscasip.aspx.cdcb7d2.compiled
- buscasipwebservice.asmx.cdcb7d2.compiled
- buscatriada.aspx.cdcb7d2.compiled
- buscatriadawebservice.asmx.cdcb7d2.compiled
- default.aspx.cdcb7d2.compiled
- denied.aspx.cdcb7d2.compiled
- resultado.aspx.cdcb7d2.compiled
- SIP.dll
- SIP.pdb



#### Conexión

- conexion.xml



#### Logs



##### 7z

- 7za.exe
- 7-zip.chm
- ComprimeLogs.bat
- copying.txt
- license.txt
- readme.txt



##### Backup

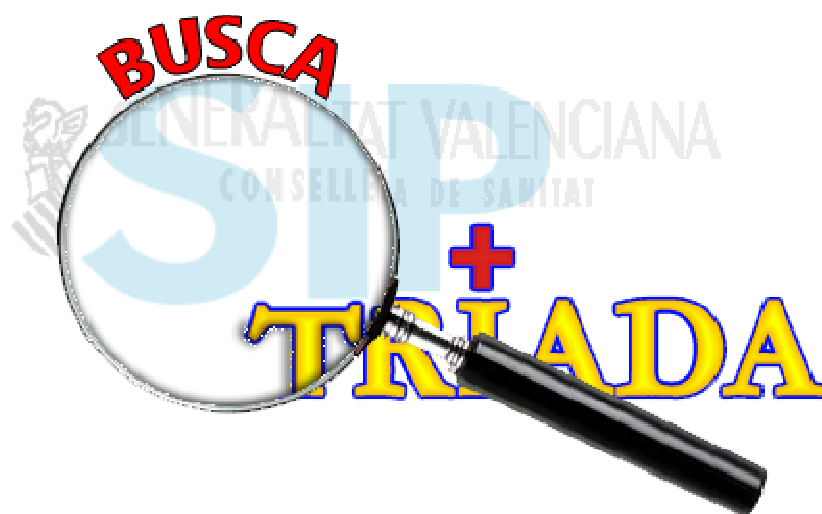
*[aquí se irán almacenando los logs de meses anteriores]*

- BuscaSip\_Log.txt
- BuscaSipWS\_Log.txt
- BuscaTriada\_Log.txt
- BuscaTriadaWS\_Log.txt

AccesoDenegado.png

BuscaSIP.aspx  
BuscaSipTriada.png  
BuscaSIPwebservice.asmx  
BuscaTriada.aspx  
BuscaTRIADAwbservice.asmx  
centro.gif  
centro.png  
CodigoXML.gif  
Default.aspx  
Denied.aspx  
Flecha.gif  
Flecha.png  
fondo.gif  
LogoSip.png  
LogoTriada.png  
Logs  
PrecompiledApp.config  
profesional.gif  
profesional.png  
Resultado.aspx  
sip.png  
web.config

## ANEXO B



**Miguel Ángel Rodríguez Leiva**

**DIRECCIÓN TERRITORIAL DE VALENCIA (CONSELLERIA DE SANITAT)**

**Abril de 2009**

**BuscaSIP-Triada** es una aplicación web destinada al uso interno de la Dirección Territorial de Valencia (*Conselleria de Sanitat*). Su funcionalidad es la misma que la de un buscador, ejecutando éste consultas sobre dos bases de datos: la global del Sistema de Información Poblacional (SIP) y una local **TRIADA** que actúa a modo de copia de la anterior. De esta manera, mediante la búsqueda directa de un número de SIP, o bien valiéndose de una búsqueda previa en la base de datos **TRIADA**, se puede acceder a la ficha personal de cada paciente que aparece en el Sistema de Información Poblacional.

La aplicación consta de cinco páginas web *.aspx* más dos servicios web *.asmx*, utilizando además las dos bases de datos de referencia y el *Active Directory* donde se almacenan todos los usuarios de la DTV para autenticar a los usuarios y comprobar sus permisos. A continuación se procede a describir el funcionamiento de cada una de las páginas webs, pudiéndose consultar más detalladamente el funcionamiento de los dos servicios webs en sus respectivos ficheros de documentación.

## **Default.aspx**



**BUSCA**  
GENERALITAT VALENCIANA  
CONSELLERIA DE SANITAT  
SIP  
+  
TRIADA

INTRODUZCA SU LOGIN Y SU PASSWORD PARA AUTENTICARSE

Login:

Password:


**SELECCIONE EL MOTOR DE BÚSQUEDA**

GENERALITAT VALENCIANA  
CONSELLERIA DE SANITAT  
SIP

GENERALITAT VALENCIANA  
CONSELLERIA DE SANITAT  
+  
TRIADA

Establecida como página de inicio, como no podía ser de otra forma, sirve para presentar la aplicación, ofreciendo la posibilidad al usuario de elegir la base de datos a la cual va a consultar.

Para ello, primero tendrá que identificarse con su login y password, siendo éste el mismo que utiliza para iniciar sesión de Windows. Después, bastará con hacer click sobre el botón de la base de datos que se va a consultar.

A esta página redirigen todas las demás cuando se pulsa el botón .

## **BuscaSIP.aspx**



GENERALITAT VALENCIANA  
CONSELLERIA DE SANITAT  
**SIP**

**Búsqueda en SIP**

Está ud. logueado como: Miguel Ángel Rodríguez Leiva

SIP:



A esta página se debe acceder cuando se conoce de antemano el número SIP del paciente del cual se quiere consultar su ficha.


Antes de cargar la página, el código interno consulta en el *Active Directory* que el login y el password introducidos en la página de inicio **Default.aspx** son correctos, y en tal caso el usuario queda validado. De no ser así, se redirigirá a la página web **Denied.aspx** donde se avisará al usuario de que no tiene autorización para hacer uso de la aplicación.

Una vez validado el usuario en el *Active Directory*, se comprobará que pertenece al grupo *BuscaSip*; o lo que es lo mismo, el grupo de usuarios que tienen permiso para hacer consultas a la base de datos del SIP. En caso de no ser así, se redirigirá también a la página web **Denied.aspx** mostrando el mismo aviso que en el caso de no estar validado.


Es por este motivo que corre a cargo de los administradores de los usuarios el proporcionar permisos para utilizar esta aplicación a las personas que vayan a poder hacer uso de la misma, siendo necesario para ello simplemente agregarlos al grupo de usuarios *BuscaSip*.

Si el usuario pasa estos dos niveles de seguridad, se mostrará la página web, que contiene simplemente una caja de texto en la que se introduce el número SIP a consultar, y al pulsar sobre el botón **Buscar**, se ejecuta la consulta sobre la base de datos del SIP, redirigiéndonos a la página web **Resultado.aspx** que contiene la información del paciente consultado.

Si se pulsa el botón **Buscar** sin haber introducido nada en la caja de texto, la aplicación avisará de esta circunstancia y no ejecutará consulta alguna.

Si se pulsa sobre el botón , se retorna a la página de inicio para efectuar una nueva búsqueda.

## **BuscaTriada.aspx**



**Búsqueda en TRIADA**

*Está ud. logueado como: Miguel Ángel Rodríguez Leiva*

SIP:

DNI:


Nombre:

Apellido 1:

Apellido 2:

Útilice el símbolo "%" como carácter comodín.

Fecha de nacimiento:  /  /



Esta página sirve de preciso buscador contra la base de datos *TRIADA*.



Antes de cargar la página, el código interno consulta en el *Active Directory* que el login y el password introducidos en la página de inicio **Default.aspx** son correctos, y en tal caso el usuario queda validado. De no ser así, se redirigirá a la página web **Denied.aspx** donde se avisará al usuario de que no tiene autorización para hacer uso de la aplicación.

Una vez validado el usuario en el *Active Directory*, se comprobará que pertenece al grupo **BuscaTriada**; o lo que es lo mismo, el grupo de usuarios que tienen permiso para hacer consultas a la base de datos TRIADA. En caso de no ser así, se redirigirá también a la página web **Denied.aspx** mostrando el mismo aviso que en el caso de no estar validado.

Es por este motivo que corre a cargo de los administradores de los usuarios el proporcionar permisos para utilizar esta aplicación a las personas que vayan a poder hacer uso de la misma, siendo necesario para ello simplemente agregarlos al grupo de usuarios **BuscaTriada**.

Si el usuario pasa estos dos niveles de seguridad, se mostrará la página web, que contiene diversos cuadros de texto que permiten introducir valores para diferentes campos de la búsqueda. Cuantos más de estos se rellenen, más acotada (y por tanto precisa) será la búsqueda. Los campos de búsqueda son los siguientes:

**SIP:** Se puede introducir una cadena de entre 4 y 7 dígitos, no permitiéndose ni letras ni ningún otro carácter. De introducirse una cadena incorrecta saltará un validador que indique la cadena que se espera.

**DNI:** Se puede introducir una cadena de exactamente 8 dígitos, siendo estos los correspondientes al número de identificación personal. También se permite, excepcionalmente, que el primer dígito sea o bien una X o bien una Y, posibilitando así la búsqueda del NIE de personas no nacidas en España. El dígito de control del NIF se calcula automáticamente mediante el mismo algoritmo que utiliza el Estado.

**Nombre, Apellido 1 y Apellido 2:** Se puede introducir cualquier cadena de caracteres que se puedan encontrar en el nombre de una persona (esto es, letras y caracteres comunes como el guión, el apóstrofo, etc.). Además, se permite también la inclusión del carácter comodín '%', que actuará como sustituto de cualquier cadena que desconozcamos, ofreciéndose así la posibilidad de hacer consultas parciales. En el caso del campo nombre, además, esta característica se aplica de manera automática como prefijo y sufijo de la cadena, facilitando así la búsqueda de nombres compuestos que a priori desconocemos cómo están introducidos en la base de datos.

**Fecha de nacimiento:** Se puede introducir la fecha de nacimiento de la persona que deseamos encontrar, usando para ello cada casilla (formato de


fecha español y con dígitos, i.e.: dd/mm/aaaa). En caso de no completarse alguna casilla se ignorarán también las casillas situadas a su izquierda, siendo así imposible hacer una búsqueda por día y año, obviando el mes, por ejemplo. Sí que se permitiría, sin embargo, hacer una búsqueda de un mes y un año, obviando el día en el que nació.

Una vez completados los campos por los que se desea efectuar la búsqueda se pulsa el botón **Buscar**, y en caso de tener los permisos necesarios, se abrirá una conexión con la base de datos y se hará la consulta oportuna.

Al recibir la respuesta que nos envía TRIADA en código XML, ésta será analizada y mostrada en forma de tabla dinámica. Ahora ya con todos los resultados mostrados en una cómoda tabla, podemos ordenarla de la manera que nos resulte más útil y proceder a buscar las personas de las que queremos obtener la información. Se mostrará también un mensaje con el número de coincidencias encontradas. En caso de que no se encuentre ninguna coincidencia en la base de datos TRIADA para los campos introducidos se avisará al usuario, al igual que si se pulsa el botón **Buscar** sin haber rellenado ninguna de las cajas de texto.

Para consultar la ficha de cualquiera de las personas mostradas en la tabla bastará con pinchar sobre el icono de la tarjeta del SIP situado en la primera casilla de la fila de la tabla. Esto lo que hace es ejecutar la misma consulta que haríamos manualmente desde la página **BuscaSIP.aspx** pero obteniendo automáticamente de la tabla el número del SIP necesario para consultar la base de datos. Al recibir respuesta desde el centro de información del SIP se nos redirigirá a la página **Resultado.aspx** donde se nos mostrará la ficha del paciente consultado.

Cabe destacar que es posible que un usuario tenga permisos para consultar la base de datos TRIADA, pero no así la del SIP. De ser éste el caso, la aplicación ha de impedir que se consulte la base de datos del SIP, comprobando de nuevo en el *Active Directory* si el usuario tiene permisos para esta segunda consulta. Así pues, la aplicación consulta en el *Active Directory* que el usuario pertenezca también al grupo **BuscaSip**. De no ser así, se mostrará un mensaje de aviso al usuario.

Si se pulsa sobre el botón , se retorna a la página de inicio para efectuar una nueva búsqueda.

## **Resultado.aspx**

Número SIP: \_\_\_\_\_  
 Número Seguridad Social: \_\_\_\_\_  
 CIP: \_\_\_\_\_

**Datos del nacimiento**  
 Fecha de nacimiento: \_\_\_\_\_  
 Lugar de nacimiento: \_\_\_\_\_

**Datos de la residencia**  
 Domicilio: \_\_\_\_\_  
 Teléfono(s): \_\_\_\_\_

**Clave Médica:** \_\_\_\_\_

Hospital Asignado: \_\_\_\_\_  
 Centro Médico Asignado: \_\_\_\_\_  
 Médico de Cabecera: \_\_\_\_\_  
 Especialidad: \_\_\_\_\_

Última actualización: \_\_\_\_\_

En esta página se muestra una ficha con los datos más relevantes del paciente. Es el resultado de obtener la información necesaria del mensaje en código XML que nos envía el centro de información del SIP, obviando ciertos apartados que no nos interesan en este contexto.

En la parte superior de la página se muestra, a modo de título de la misma, los apellidos y el nombre del paciente, seguido de sus números de identificación (i.e.: DNI, número del SIP, número de la Seguridad Social y código CIP).


Después, recuadrado en azul, se muestran los datos referentes a su nacimiento (fecha y lugar) y residencia (domicilio y teléfono(s)).

A continuación, y recuadrado en rojo, se muestran los datos médicos del paciente, siendo estos su clave médica y sus centros (hospital y ambulatorio) y médico (nombre del médico y especialidad) asignados.

Finalmente se muestra la fecha de la última actualización, dato muy a tener en cuenta para según qué datos.

También se ha optado por incluir la cadena entera en código XML que nos devuelve la base de datos del SIP, y que contiene más información de la mostrada en la ficha. Por defecto esta información aparece oculta, y para mostrarla solo es necesario hacer clic sobre el botón . Para volver


a ocultar el código bastará con volver a hacer clic sobre el mismo botón, ya que éste actúa como alternador de la visibilidad de la caja de texto que contiene el mensaje en XML.

Si se pulsa sobre el botón , se retorna a la página de inicio para efectuar una nueva búsqueda.

## **Denied.aspx**



Ésta es la página que se muestra cuando, por cuestiones de permisos, un usuario no está autorizado a hacer uso de la aplicación seleccionada.

Tanto *BuscaSIP* como *BuscaTRIADA* consultan a la base de datos de permisos antes de permitir el acceso a la aplicación, bloqueando el acceso a las direcciones IP no autorizadas. En estos casos la propia aplicación redirige a esta página, que sólo permite pulsando sobre el botón  regresar a la página de inicio **Default.aspx**.

## ANEXO C

### BuscaSIPwebservice

Servicio web que sirve como pasarela entre la aplicación que lo utilice y la base de datos del SIP.

Consta de una función pública principal, **busca\_SIP**, y dos privadas a las que únicamente puede llamar la propia función principal.

**BuscaSIPwebservice**

Haga clic [aquí](#) para obtener una lista completa de operaciones.

---

**busca\_SIP**

Función que dado un número de SIP, envía la petición en código XML para así recibir, también en código XML la respuesta por parte de la base de datos del SIP, incluyendo ésta toda la información solicitada en la petición.

**Prueba**

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

Parámetro	Valor
SIP:	<input type="text"/>
usuario:	<input type="text"/>
contraseña:	<input type="text"/>

En primer lugar el servicio web obtiene la IP que hace la consulta y la almacena en una variable. Es entonces cuando abre una conexión con la tabla auxiliar *TriadaWS* (alojada en *TRIADA*) y consulta que esa IP en concreto tenga permisos para hacer uso del servicio web en cuestión. En caso afirmativo, la ejecución del servicio web continúa. Si por el contrario se detecta que la IP no aparece en la base de datos o ésta no tiene permisos para proseguir con la consulta a la base de datos del SIP, el servicio web devuelve a la aplicación un mensaje de error y termina así su ejecución.

El mensaje de error devuelto es el siguiente:

```
<respuesta>USTED NO TIENE PERMISO PARA CONSULTAR ESTA BASE DE DATOS.</respuesta>
```

La función **busca\_SIP** recibe una cadena de texto que contiene el número de identificación de alguna de las personas que aparece en el Sistema de Información Poblacional (SIP). Ésta, mediante la función **peticiónXML**, crea una petición en formato XML que contiene los detalles de la conexión a la base de datos del SIP, así como los del usuario que realiza la petición y los bloques de información del paciente que se desean consultar. Una vez creada la petición, el servicio web, mediante la función **TestPetición**, conecta con la base de datos del SIP, envía la petición anteriormente generada, y recibe un mensaje en formato XML que contendrá la respuesta que el SIP nos contesta. Finalmente, **busca\_SIP** devuelve este mensaje en formato XML.

Como siempre en estos casos, corre a cargo de la aplicación que haga uso del servicio web la validación de que la cadena que se pasa como parámetro 'SIP' a la función **busca\_SIP** sea del tipo y formato correctos. En este caso necesariamente será una cadena compuesta únicamente por dígitos [0-9], siendo la longitud máxima de ésta 7 caracteres. También corre a cargo de la aplicación el posterior tratamiento de la respuesta XML obtenida.

Los campos **usuario** y **contrasena** serán necesarios para comprobar en primera instancia que el usuario que pretende consultar la base de datos aparece en el *Active Directory* y tiene los permisos requeridos para hacer uso de este servicio web.

En caso de que no sea posible obtener la información sobre esa persona se recibirá una cadena del tipo:

```
<msg_respuesta> <error>El N° SIP no existe</error> </msg_respuesta>
```

Se ha incluido también un log (*~\Logs\BuscaSipWS\_Log.txt.txt*) que se actualiza por cada consulta al servicio web y que almacena: la fecha y hora, el usuario que la realiza, la dirección IP de la request, y finalmente el número SIP consultado.

A continuación se adjunta un anexo en el que se muestran ejemplos de solicitudes para diferentes protocolos, en este caso para **SOAP 1.1**, **SOAP 1.2** y **HTTP POST**.

## Anexo I: Diferentes solicitudes y respuestas

### SOAP 1.1

A continuación se muestra un ejemplo de solicitud y respuesta para SOAP 1.1. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Triada_Prueba_001/BuscaSIPwebservice.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "BuscaSIP/busca_SIP"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <busca_SIP xmlns="BuscaSIP">
      <SIP>string</SIP>
      <usuario>string</usuario>
    </busca_SIP>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <busca_SIPResponse xmlns="BuscaSIP">
      <busca_SIPResult>string</busca_SIPResult>
    </busca_SIPResponse>
  </soap:Body>
</soap:Envelope>
```

### SOAP 1.2

A continuación se muestra un ejemplo de solicitud y respuesta para SOAP 1.2. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Triada_Prueba_001/BuscaSIPwebservice.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <busca_SIP xmlns="BuscaSIP">
      <SIP>string</SIP>
      <usuario>string</usuario>
    </busca_SIP>
  </soap12:Body>
</soap12:Envelope>
```

```
</busca_SIP>  
</soap12:Body>  
</soap12:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">  
  <soap12:Body>  
    <busca_SIPResponse xmlns="BuscaSIP">  
      <busca_SIPResult>string</busca_SIPResult>  
    </busca_SIPResponse>  
  </soap12:Body>  
</soap12:Envelope>
```

## HTTP POST

A continuación se muestra un ejemplo de solicitud y respuesta para HTTP POST. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Triada_Prueba_001/BuscaSIPwebservice.asmx/busca_SIP HTTP/1.1  
Host: localhost  
Content-Type: application/x-www-form-urlencoded  
Content-Length: length
```

```
SIP=string&usuario=string
```

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<string xmlns="BuscaSIP">string</string>
```



## ANEXO D

### BuscaTRIADAwbservice

Servicio web que sirve como pasarela entre la aplicación que lo utilice y una copia de la base de datos de TRIADA, situada en el servidor *cipres* de la DTV.

Consta de una función pública principal, **busca\_TRIADA**, y otra privada llamada **letra\_nif**, únicamente accesible por la función principal.

### BuscaTRIADAwbservice

Haga clic [aquí](#) para obtener una lista completa de operaciones.

---

#### busca\_TRIADA

Función que devuelve un código XML con todas las coincidencias encontradas en TRIADA para los campos solicitados. (Para los apellidos se puede utilizar el símbolo '%' como carácter comodín)

**Prueba**

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

Parámetro	Valor
sip:	<input type="text"/>
dni:	<input type="text"/>
nombre:	<input type="text"/>
apellido1:	<input type="text"/>
apellido2:	<input type="text"/>
día:	<input type="text"/>
mes:	<input type="text"/>
año:	<input type="text"/>
usuario:	<input type="text"/>
contraseña:	<input type="text"/>

La función **busca\_TRIADA** recibe hasta ocho cadenas de texto que contienen información sobre la/s persona/s que queremos localizar en TRIADA. Cada campo hace referencia a una columna de la base de datos TRIADA (excepto *día*, *mes* y *año* que formarán la cadena contenida en la columna *FechaNac*) y cuantos más rellenemos, más precisa será nuestra búsqueda.

Los campos a rellenar son: número de identificación del SIP (*sip*), número del DNI del paciente sin incluir el dígito de control del NIF<sup>1</sup> (*dni*), nombre y apellidos (*nombre*, *apellido1* y *apellido2*), y fecha de nacimiento (*día*, *mes* y *año*).

En primer lugar el servicio web obtiene la IP que hace la consulta y la almacena en una variable. Es entonces cuando abre una conexión con la tabla auxiliar *TriadaWS* (alojada en TRIADA) y consulta que esa IP en concreto tenga permisos

<sup>1</sup> La función **letra\_nif** ya se encarga de calcular el dígito de control, utilizando para ello el mismo algoritmo que emplea el Estado.

para hacer uso del servicio web en cuestión. En caso afirmativo, la ejecución del servicio web continúa. Si por el contrario se detecta que la IP no aparece en la base de datos o ésta no tiene permisos para proseguir con la consulta a la base de datos TRIADA, el servicio web devuelve a la aplicación un mensaje de error y termina así su ejecución.

El mensaje de error devuelto es el siguiente:

```
<respuesta>USTED NO TIENE PERMISO PARA CONSULTAR ESTA BASE DE DATOS.</respuesta>
```

Una vez superada la comprobación de permisos, la función establece una conexión con la base de datos TRIADA del servidor *cipres*. Una vez conectado, y en función de qué campos hayamos completado, se creará la sentencia *SELECT* oportuna, mediante la cual haremos la consulta a la base de datos TRIADA. Con esta sentencia *SELECT* haremos la consulta correspondiente a la base de datos, que nos devolverá una cadena en formato XML con los valores de las columnas de cada coincidencia, i.e. cada paciente cuyos datos se correspondan con los de la búsqueda realizada.

Cabe destacar que, si bien los campos *sip* y *dni* efectúan una búsqueda exacta (operador '='), para los campos *nombre*, *apellido1* y *apellido2* se permite realizar una búsqueda abierta (operador 'LIKE'), haciendo uso del carácter comodín '%' como sustituto de cualquier cadena que desconozcamos. En el caso concreto del campo *nombre* esto no es necesario puesto que la propia sentencia *SELECT* ya contempla esta situación<sup>2</sup>, por lo que el uso del carácter comodín '%' aquí sería redundante (sin causar ningún problema en su correcto funcionamiento). Los campos de la fecha de nacimiento (i.e. *dia*, *mes* y *anyo*) sólo se tendrán en cuenta si se rellenan también los campos posteriores; es decir, será imposible hacer una búsqueda por día y año, obviando el mes, por ejemplo. Sí que sería posible, sin embargo, hacer una búsqueda de un mes y un año, obviando el día en el que nació.

Como siempre en estos casos, corre a cargo de la aplicación que haga uso del servicio web la correcta validación de los campos de la búsqueda, teniendo en cuenta en este caso concreto que:

**sip:** Ha de ser una cadena de hasta 7 dígitos [0-9].

**dni:** Ha de ser una cadena de 8 dígitos, pudiendo ser el primero de ellos las letras 'X' o 'Y' (en mayúsculas o minúsculas) para el caso de personas con número de identidad de extranjero (NIE).

**nombre:** Ha de ser una cadena de texto que no incluya caracteres numéricos pero sí signos de acentuación, apóstrofes, guiones, etc.

**apellido1:** Mismas condiciones que para el campo *nombre*.

**apellido2:** Mismas condiciones que para los campos *nombre* y *apellido1*.

**dia<sup>3</sup>:** Ha de ser una cadena de uno o dos dígitos formando un número entre [1, 31].

---

<sup>2</sup> Se ha optado por esta solución para facilitar la búsqueda de personas con nombres compuestos, que además aparecen en muchas ocasiones abreviados.

<sup>3</sup> En general para los tres campos que conforman la fecha de nacimiento, han de formar una fecha correcta, teniendo en cuenta los días que tiene cada mes así como los años bisiestos.

**mes<sup>3</sup>**: Ha de ser una cadena de uno o dos dígitos formando un número entre [1, 12].

**anyo<sup>3</sup>**: Ha de ser una cadena de cuatro.

También queda a cargo de la aplicación el tratamiento de la respuesta que devuelve la función **busca\_TRIADA**, teniendo en cuenta que ésta será una cadena XML que contendrá un bloque principal *<respuesta>*, y un sub-bloque *<persona>* por cada coincidencia encontrada. Dentro de cada sub-bloque *<persona>*, a su vez, aparecerá un bloque por cada columna de la base de datos para esa persona en concreto, siendo estos los siguientes (por este mismo orden):

*<USUARIO>*: Número de identificación del SIP.

*<APE1>*: Primer apellido.

*<APE2>*: Segundo apellido.

*<NOM>*: Nombre.

*<NIF>*: Número de NIF.

*<FechaNacimiento>*: Fecha de nacimiento.

*<NumSS>*: Número de la Seguridad Social.

En el caso de de no encontrarse ninguna coincidencia para la búsqueda efectuada, el mensaje devuelto será el siguiente:

*<respuesta>NO SE HA ENCONTRADO NINGUNA COINCIDENCIA.</respuesta>*

Los campos **usuario** y **contrasena** serán necesarios para comprobar en primera instancia que el usuario que pretende consultar la base de datos aparece en el *Active Directory* y tiene los permisos requeridos para hacer uso de este servicio web.

Se ha incluido también un log (*~\Logs\BuscaTriadaWS\_Log.txt.txt*) que se actualiza por cada consulta al servicio web y que almacena: la fecha y hora, el usuario que la realiza, la dirección IP de la request, y finalmente la sentencia select que se pasa a la base de datos para realizar la consulta.

A continuación se adjunta un anexo en el que se muestran ejemplos de solicitudes para diferentes protocolos, en este caso para **SOAP 1.1**, **SOAP 1.2** y **HTTP POST**.

## Anexo I: Diferentes solicitudes y respuestas

### SOAP 1.1

A continuación se muestra un ejemplo de solicitud y respuesta para SOAP 1.1. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Triada_Prueba_001/BuscaTRIADAwbservice.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "BuscaTRIADA/busca_TRIADA"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <busca_TRIADA xmlns="BuscaTRIADA">
      <sip>string</sip>
      <dni>string</dni>
      <nombre>string</nombre>
      <apellido1>string</apellido1>
      <apellido2>string</apellido2>
      <dia>string</dia>
      <mes>string</mes>
      <anyo>string</anyo>
      <usuario>string</usuario>
    </busca_TRIADA>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <busca_TRIADAResponse xmlns="BuscaTRIADA">
      <busca_TRIADAResult>string</busca_TRIADAResult>
    </busca_TRIADAResponse>
  </soap:Body>
</soap:Envelope>
```

### SOAP 1.2

A continuación se muestra un ejemplo de solicitud y respuesta para SOAP 1.2. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Triada_Prueba_001/BuscaTRIADAwbservice.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <busca_TRIADA xmlns="BuscaTRIADA">
      <sip>string</sip>
      <dni>string</dni>
      <nombre>string</nombre>
      <apellido1>string</apellido1>
      <apellido2>string</apellido2>
      <dia>string</dia>
      <mes>string</mes>
      <anyo>string</anyo>
      <usuario>string</usuario>
    </busca_TRIADA>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <busca_TRIADAResponse xmlns="BuscaTRIADA">
      <busca_TRIADAResult>string</busca_TRIADAResult>
    </busca_TRIADAResponse>
  </soap12:Body>
</soap12:Envelope>
```

## HTTP POST

A continuación se muestra un ejemplo de solicitud y respuesta para HTTP POST. Es necesario reemplazar los **marcadores de posición** que aparecen con valores reales.

```
POST /Triada_Prueba_001/BuscaTRIADAwebservice.asmx/busca_TRIADA
HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length

sip=string&dni=string&nombre=string&apellido1=string&apellido2=string&
dia=string&mes=string&anyo=string&usuario=string
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="BuscaTRIADA">string</string>
```

# ANEXO E

## CONTENIDO DEL CD



### II-B-DISCA-117-08



#### Documentos

Memoria del PFC.pdf

Contenido del CD.pdf



#### Imágenes

*Contiene las imágenes empleadas en el documento de la memoria*



#### Ficheros de la Aplicación




##### Ficheros del Proyecto en Visual Studio

*Contiene todos los ficheros propios de la realización de este proyecto en el entorno de programación Visual Studio 2005. Estos incluyen tanto el código fuente de las diferentes partes de la aplicación web, como los servicios web desarrollados, además de las imágenes utilizadas y las carpetas y ficheros que el entorno de programación crea de manera automática.*



##### Aplicación Web en el Servidor dtvceres

*Contiene la carpeta  **PruebasSip** que hay que ubicar en el servidor **dtvceres** para hacer posible el acceso a la aplicación web de manera remota y a través de su URL específica. Esta carpeta es el resultado del trabajo desarrollado en Visual Studio y contiene para cada parte de la aplicación web sus ficheros compilados, además de otros ficheros propios del proyecto en Visual Studio necesarios para su correcta ejecución, los logs que almacenan la información de búsquedas e intentos fallidos de acceso a la aplicación, el programa **7zip** y el script **ComprimeLogs.bat** necesarios para la compresión y almacenaje de los logs de meses anteriores, y el archivo XML a modificar en caso de querer cambiar los parámetros de la conexión a la base de datos **TRIADA**.*