



# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



## SISTEMA DE SOPORTE ONLINE CON ASIGNACIÓN AUTOMÁTICA

Carlota Katusca Alvarado Vásquez, Manuela Elizabeth Flores Tomalá, Carlos Manuel Martín Barreiro

Facultad de Ingeniería en Electricidad y Computación

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil-Ecuador

[ckalvara@fiee.espol.edu.ec](mailto:ckalvara@fiee.espol.edu.ec) , [mflores@fiee.espol.edu.ec](mailto:mflores@fiee.espol.edu.ec) , [cmmartin@espol.edu.ec](mailto:cmmartin@espol.edu.ec)

### Resumen

*El propósito de este trabajo es el de crear un sistema capaz de gestionar de manera centralizada todas las incidencias de soporte técnico de una red de usuarios, para mantener una comunicación oportuna y eficaz con los mismos y poder tomar medidas necesarias que cubran sus expectativas y mejoren la calidad del servicio que ofrece un departamento de Sistemas.*

*Se identifican los objetivos del proyecto, se presenta el marco teórico de la tecnología a emplearse, explicando las ventajas de las herramientas seleccionadas, debido a que los avances en tecnología demandan aplicaciones cada vez más rápidas, ligeras y robustas que permitan utilizar de manera óptima los recursos que le proporciona la Web.*

*Se describe el Análisis y Diseño de la Solución, detallándose los aspectos de arquitectura como: funcionalidad, diseño, interfaz de usuario del sistema, y las opciones con las que cuenta de acuerdo al rol del usuario.*

*Finalmente se presentan las conclusiones obtenidas en el desarrollo del sistema; también recomendaciones en cuanto a mejoras complementarias al alcance definido en este trabajo.*

**Palabras Claves:** *Soporte en línea, Soporte Online, Ayuda Técnica.*

### Abstract

*The purpose of this project is developing a powerful web application which will be able to generate an accurate record of every single tech support provided to the users of a local network.*

*We offer a very profound study of the web tools we used to work on this project. We focused on the constant developing of web technology, which demands not only secure and steady applications but also friendly environments where users can easily interact with it.*

*Based on the requirements and needs provided by the client, the project main goals are explained in detail, as well as the fields on which the system could be applied.*

*We abundantly discuss about the improvements this application may need in the future, what is more, we also talk about the "Do's" and "Don'ts" we found from the design process to the final conclusions.*

**Keywords:** *Support Online, Helpdesk.*

## 1. Introducción

El Sistema de Soporte Online propuesto es una herramienta que comprende una solución completa para la gestión de la atención a los usuarios de una red LAN en una institución, tiene como objetivo principal ser una herramienta que permita una comunicación oportuna y eficaz con los usuarios para poder tomar medidas necesarias que cubran sus expectativas y mejoren la calidad del servicio que ofrece un departamento o empresa.

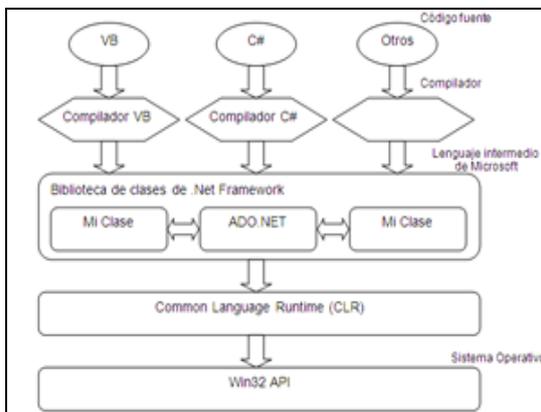
Una de las cosas más importantes a destacar en este sistema es que se mantienen las seguridades necesarias para que el administrador y técnicos realicen una gestión adecuada en lo que se refiere a la atención que brinda el departamento o empresa.

Cualquier usuario puede con su nombre de usuario y clave respectiva, realizar consultas de soluciones a problemas, enviar un nuevo requerimiento o problema que se le presentare.

El Sistema de Soporte Online es una aplicación orientada a ser una ayuda tanto para los usuarios, administradores y responsables técnicos de cualquier empresa ó institución que brinda servicios al público en general.

## 2. Marco Teórico

### 2.1 Framework .NET



**Figura 2.1:** Arquitectura de .Net Framework [2]

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones.

Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear

aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- *Lenguajes de compilación*
- *Biblioteca de clases de .Net*
- *CLR (Common Language Runtime)*

**2.1.1. Common Language Runtime (CLR).** Es el verdadero núcleo del Framework de .Net, ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios que ofrece el sistema operativo estándar Win32.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .Net en un mismo código, denominado código intermedio (MSIL, Microsoft Intermediate Language). Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear código MSIL compatible con el CLR.

**Biblioteca de clases de .Net** La biblioteca de clases de .Net Framework incluye, entre otros, tres componentes clave:

- *ASP.NET* para construir aplicaciones y servicios Web.
- *Windows Forms* para desarrollar interfaces de usuario.
- *ADO.NET* para conectar las aplicaciones a bases de datos.

**2.1.2. ASP .Net** Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET esta construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

**2.1.3. C#** Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET, similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi). Aunque C# forma parte



# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



de la plataforma.NET, ésta es una interfaz de programación de aplicaciones (API), mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco de DotGNU - Mono que genera programas para distintas plataformas como Win32, UNIX y Linux.

## 2.2 Tecnología AJAX

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax el mismo que utiliza el objeto XMLHttpRequest mediante el cual se permite el acceso a los datos, este objeto se encuentra disponible en los navegadores actuales.

## 2.3 Plugin JavaScript

**2.3.1 JQuery** Es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2,[1] permitiendo su uso en proyectos libres y privativos.[2] jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

## 2.4 Web Service

Un web service es básicamente una función o procedimiento que puede ser accedida vía web por cualquier programa o aplicación sin importar en que plataforma reside el servicio o en que lenguaje ha sido desarrollado, el término "web" implica que el acceso se hace mediante una conexión a internet habitualmente vía http aunque otros protocolos de transporte pueden ser utilizados. De esta forma, una aplicación Visual Basic puede, por ejemplo, usar una rutina de conversión de metros a yardas que fue realizada en Python y reside en un servidor de la India.

## 2.5 SOAP

SOAP es un protocolo que define el formato XML para los mensajes de intercambio en el uso de un Web Service. Para aquellos programadores que solían utilizar llamadas del tipo RPC, SOAP también las soporta. Adicionalmente, es posible mediante SOAP definir un mensaje HTTP y este punto es de especial interés puesto que el protocolo imprescindible para Internet es HTTP.

## 2.6 Control de Versiones

**2.6.1 Subversión** Herramienta de código abierto, multiplataforma (Win32, Linux, Mac, etc), para el control de versiones de ficheros electrónicos, como son el software o la documentación.

Se basa en un repositorio central que actúa como un servidor de ficheros, con la capacidad de recordar todos los cambios que se hacen tanto en sus directorios como en sus ficheros.

**2.6.2 TortoiseSVN** Es un cliente gratuito de código abierto para el sistema de control de versiones Subversion.

Integrado en la shell de Windows, por ejemplo el "explorador". Y ni siquiera está obligado a usar el Explorador de Windows. Los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos, y en la ventana Fichero/Abrir que es común a la mayoría de aplicaciones estándar de Windows.

## 3. Análisis de la Solución

### 3.1 Análisis de Requerimientos

Luego de poseer un conocimiento formal del problema y de haber definido las posibles herramientas tecnológicas en el desarrollo de la solución para apoyar y solventar la situación o

situaciones problemáticas identificadas, en esta fase se detallan los requisitos identificados durante la fase de definición del proyecto con la finalidad de identificar nuevos requisitos funcionales, de información y operacionales que la solución debe satisfacer para lograr los objetivos planteados.

### 3.2 Requerimientos Funcionales

Los principales requerimientos funcionales que va a presentar la aplicación propuesta son las siguientes:

- Crear usuarios para la aplicación y especialistas que se harán cargo de los diferentes casos.
- Hacer asignaciones automáticas y reasignaciones de los diferentes casos a los responsables.
- Obtener una serie de reportes para conocer el avance de los requerimientos.
- Disponer de una base de conocimientos que faciliten llegar a las soluciones a los diferentes especialistas.
- Registrar las actividades que se realizan para hacer un seguimiento (historial) de cada requerimiento.
- Consultar los requerimientos y actividades que se han realizado para llegar a una solución.
- El usuario no necesita poseer grandes conocimientos en informática para manejar la aplicación.
- No necesita instalar programas adicionales para interactuar con la aplicación.
- Poseer una interfaz sencilla de manejar y comprender.

### 3.3 Requerimientos no funcionales

Los requerimientos no funcionales son todas aquellas características que debe cumplir el sistema para responder de manera adecuada a todos los requerimientos funcionales y a las características que requiera el usuario.

A continuación listaremos los requisitos contemplados para la aplicación:

#### Rendimiento:

- Proporcionar tiempos de respuesta aceptables en los procesos en línea del sistema.
- Garantizar velocidad estable de navegación para los clientes del sistema.
- Optimizar la ejecución de procesos de sistema para disminuir la congestión de recursos.

#### Mantenibilidad:

- Emplear un modelo unificado de desarrollo, uso de un marco de trabajo.

#### Seguridad:

- El acceso será controlado con nombres de usuario y contraseñas.
- Almacenamiento cifrado de contraseñas.

#### Usabilidad:

- Que sea familiar a las aplicaciones web y de escritorio comúnmente utilizadas.
- Apoyar la comprensión usando principios de diseño de Interfaz Centrado en el Usuario.

## 4. Diseño de la Solución

### 3.4 Diseño de la Arquitectura del Sistema



**Figura 4.1:** Diseño de la Arquitectura del Sistema

Se utilizó una Arquitectura orientada a Aplicaciones Web. Las capas de la arquitectura son las siguientes:

**3.4.1 Capa de presentación** Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios. En nuestro caso creamos un Sitio Web llamado SO\_Soporte, el mismo que contiene nuestros Web Forms.

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. Aún así, existen oportunidades para identificar servicios que residen en componentes separados.

**3.4.2 Capa de negocios** Los servicios de negocios son el “puente” entre un usuario y los servicios de datos. Responden a peticiones del usuario para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Cuando los datos necesarios residen en un servidor de bases de datos, garantizan los servicios de datos indispensables para cumplir con la tarea de negocios o aplicar su

regla. Esto aísla al usuario de la interacción directa con la base de datos.

En nuestro caso creamos una librería llamada SO\_LibSoporte, la misma que contiene nuestras entidades y su respectivo comportamiento.

**3.4.3 Capa de datos** Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

En nuestro caso creamos un Web Service que nos servirá como una capa de Acceso a Datos, pues uno de los objetivos es poder migrar en algún momento a otra base de datos y que este proceso sea fácil de realizar.

## 4 Implementación de la solución

### 4.1 Tecnología

Dentro de la implementación del sistema se utilizaron diferentes tipos de tecnologías Web aplicadas al servidor y al cliente para mejorar e interacción:

- Lenguaje de Programación
- Lenguaje de intercambio de Datos
- Framework del Servidor
- Framework del Cliente
- Web Service.

**4.1.1 Lenguaje de Programación.** Los lenguajes de programación escogidos para la implementación del sistema fueron ASP.NET como generador de webforms y C#.NET (code behind) o código asociado en el servidor y Javascript en el cliente.

**4.1.2 Lenguaje de intercambio de Datos** Una parte esencial de nuestro sistema es el envío de mensajes a través de llamadas AJAX .

**AJAX:** no es una tecnología en sí mismo, en realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

**CSS:** es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML .

**DOM:** Document Object Model o en español Modelo de objetos del documento, es un conjunto de estándares de objetos para representar documentos HTML y XML, además lo utilizamos para acceder y modificar el contenido y estructura de los mismos.

**JSON:** es un formato ligero para el intercambio de datos. JSON, es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

**4.1.3 Framework del Servidor.** Es una plataforma, entorno o marco de trabajo para el desarrollo de aplicaciones Web, brinda una estructura de soporte definida para poder organizar y desarrollar otro proyecto de software. Incluyen soporte de programas, bibliotecas, Lenguaje de scripting, software para unir diferentes componentes de desarrollo de programas.

**4.1.4 Framework del Cliente.** El framework que se utiliza del lado del cliente es JQUERY.

JQuery es una librería de java script que ayuda a simplificar la manera de interactuar los documentos html. El uso de JQuery como framework de desarrollo facilita el uso de AJAX integrado como parte de su solución.

**4.1.5 Implementación por capas.** El sistema de Soporte Online esta desarrollado por una arquitectura de tres capas.

### 4.2 Diagrama de la Base de Datos

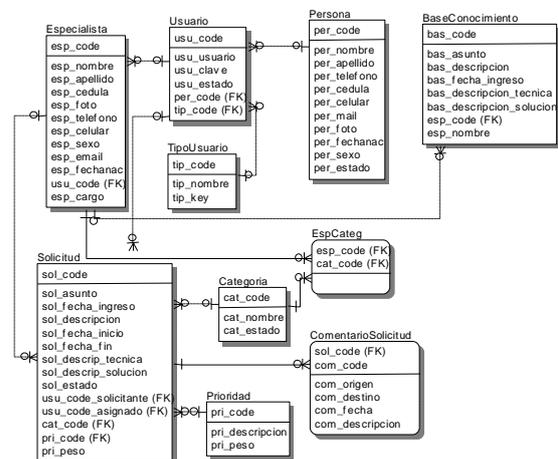


Figura 4.2: Diseño de la Base de Datos

## 5. Conclusiones

Con la implementación del Sistema de Soporte Online, se espera conseguir:

1. La reducción de costos operativos, ya que se busca que la mayor parte de las solicitudes de soporte técnico sean procesados a través del sistema.



# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



2. Facilidad en la solución de problemas mediante el conocimiento de soluciones anteriores.

3. Facilitar la comunicación y la atención a los usuarios a través de las opciones que brinda el sistema, reduciendo el tiempo de espera para la resolución de los casos.

## 6. Recomendaciones

En base al desarrollo del sistema y de sugerencias de los usuarios en las pruebas iniciales, podemos realizar las siguientes recomendaciones las mismas que permitirán mejorar y ampliar los servicios que se brindan.

Agregar nuevos servicios en el sistema tales como:

1. Índices de gestión por Especialista, con la finalidad de medir el rendimiento del equipo de trabajo.
2. Incluir una opción de chat.
3. Gráficos estadísticos de cantidad de Solicitudes pendientes, resueltas, etc.
4. Publicar documentos de interés (manuales) para los usuarios de acuerdo a las especialidades.

## 7. Agradecimientos

Agradecemos a Dios, a nuestras familias, profesores y amigos, los que nos proporcionaron fuerzas, ayuda, conocimiento y apoyo para salir adelante.

## 8. Referencias Bibliográficas:

[1] Crystal Reports, “Alta Productividad en creación de reportes e integración de datos”.

<http://www.crystalsolutions.com.ar/productos/crystalreports.html>.

[En línea][Fecha visita: 20 de Septiembre de 2010]

[2] ASP.NET, “Arquitectura básica de la Plataforma .net”.

<http://www.desarrolloweb.com/articulos/1328.php>

[En línea][Fecha visita: 20 de Septiembre de 2010]

[3] The JQuery Project JQuery Framework de JavaScript.

<http://jquery.com>

[En línea][Fecha visita: 5 de Octubre de 2010]

[4] Wikipedia, Web 2.0.

[http://en.wikipedia.org/wiki/Web\\_2](http://en.wikipedia.org/wiki/Web_2)

[En línea][Fecha visita: 10 de Octubre de 2010]

[5] Marcio arrios, Ajax el futuro?

<http://www.marcioarrios.com/ajax>

[En línea][Fecha visita: 15 de Octubre de 2010]

[6] Slide Share, Arquitectura de 3 capas.

<http://www.slideshare.net/Decimo/arquitectura-3-capas>

[En línea][Fecha visita: 25 de Octubre de 2010]

[7] Microsoft, Soluciones SQL Server 2008.

<http://www.microsoft.com/latam/sql/2008/default.aspx>

[En línea][Fecha visita: 10 de Noviembre de 2010]

[8] TechNet, Evalúe Microsoft SQL Server 2008 hoy mismo.

<http://technet.microsoft.com/es-es/evalcenter/bb851664.aspx>

[En línea][Fecha visita: 11 de Noviembre de 2010]

[9] Microsoft, Las 30 características principales de SQL Server 2005.

<http://www.microsoft.com/spain/sql/productinfo/features/top30features.msp> [En línea][Fecha visita: 11 de

Noviembre de 2010]

[10] Clikear, Lenguaje C#.

<http://www.clikear.com/manuales/csharp/c10.aspx>

[En línea][Fecha visita: 11 de Noviembre de 2010]