

Sistema de Control de Pacientes e Inventario FUVIDA (SCPFI)

Jorge Nuñez T. ⁽¹⁾ Farid Díaz R. ⁽²⁾ Vanessa Cedeño M. ⁽³⁾
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
jorgefren12@gmail.com⁽¹⁾ adiazru@gmail.com⁽²⁾ vcedeno@espol.edu.ec⁽³⁾

Resumen

En el presente artículo se resume como se ha elaborado el Sistema de Control de Pacientes e Inventario FUVIDA (SCPFI) y el sitio web para la fundación Fuvida. El sistema SCPFI fue elaborado para ayudar a fundación FUVIDA a llevar un mejor control de la información de sus pacientes e insumos médicos, fue desarrollado en un ambiente web, utilizando distintos lenguajes y herramientas de programación tales como: Symfony, Html, JQuery, Javascript. SCPFI cuenta con 4 módulos que pueden ser accedidos por medio de los dos roles que tiene el sistema Administrador y Usuario: Pacientes, Préstamos, Usuario y Productos. El sitio web se desarrolló utilizando Joomla una herramienta de administrador de contenidos gratuita más conocido como CMS (Content Management System) muy utilizado para crear sitios web dinámicos, el cual da facilidades de publicación de información como: noticias, enlaces, galerías de imágenes, etc. Este sitio web se concibió para que la fundación FUVIDA de a conocer los servicios que esta brinda a la comunidad.

Palabras Claves: SCPFI, ESPOL, Symfony, Joomla, Fuvida.

Abstract

This article summarizes the development of the Patient Monitoring System and Inventory FUVIDA (SCPFI) and the website for the foundation Fuvida. The SCPFI system was developed to help foundation FUVIDA take better control of patient information and medical supplies. It was developed in a web environment, using different programming languages and frameworks such as Symfony, Html, JQuery, Javascript. SCPFI has 4 modules that can be accessed via the two roles that have the Administrator and User System: Patients, Loans, User and products. The website was developed using a Framework called Joomla that is a content management system better known as CMS (content Management System). Joomla is used to create dynamic websites, which provides facilities for publishing information as: news, links, image galleries, etc.. This web site was designed so that the foundation FUVIDA show awareness of the services it provides to the community.

Keywords: SCPFI, ESPOL, Symfony, Joomla Fuvida.

1. Introducción

FUVIDA es una fundación dedicada a ayudar a niños con diabetes, maneja los datos de los pacientes a través de papel y documentos electrónicos como lo son los archivos Microsoft Excel, teniendo la necesidad de poseer una aplicación para automatizar el control de productos que ingresan a la fundación y préstamos de Implementos médicos, Ficha Médica del paciente y Sitio web informativo de la fundación.

Contando con un sitio web informativo se podrá mostrar al mundo los servicios que se brindan, y de esta manera muchas personas que estén buscando una fundación que pueda ayudarlos con su enfermedad, como es la diabetes, lo encuentren por medio de internet.

No solo podrán ayudar a posibles pacientes que padezcan de la enfermedad de diabetes, sino que al tener un espacio dentro del internet, FUVIDA puede ser encontrada por distintas personas, empresas o entidades anónimas, que puedan ayudar a este tipo de causas y logren un contacto con la fundación colaborando con donaciones económicas, víveres, implementos médicos o cualquier tipo de donación que pueda ser útil.

Considerando que no solo se desarrolla una página web sino también una aplicación web para el control de inventario y ficha médica de los pacientes, los beneficios que se obtienen externamente con la aplicación también se podrán ver reflejados internamente, ya que las donaciones o nuevos pacientes que ingresan a la fundación, al ser

registrados en el nuevo sistema tendrán un verdadero control, así como también lo habrá de los implementos médicos que ingresan a la fundación y que luego son donados, prestados o vendidos a los distintos pacientes.

2. Antecedentes

En Ecuador hay alrededor de 500 mil personas con diferentes tipos de diabetes, uno de los cuales es la diabetes tipo I, que afecta sobre todo a los niños y adolescentes, lo que significa que su organismo no produce insulina todos los días. Por ende pueden existir episodios de Hipoglucemia (baja presencia de azúcar en sangre, dolores de cabeza, temblores), Hiperglucemia (alta presencia de azúcar en la sangre, aumento de sed, aumento de hambre, náuseas, vómitos), Cetacidosis (es la acumulación de cetonas en la sangre debido a la falta de insulina en el organismo).

Existen diferentes organizaciones dentro del país que se encargan de ayudar y de guiar a pacientes que padecen el tipo de diabetes I, debido al desconocimiento y alto costo de esta enfermedad. Una de estas es la Fundación Fuvida, ubicada en la ciudad de Guayaquil, creada en el año 1998, y legalizada en el año 2006, que brinda ayuda a través de: medicina, control médico y educación a más de 300 pacientes con este tipo de diabetes. El sistema de almacenamiento de datos que están utilizando, es Excel, una aplicación que sirve para manejar hojas de cálculo y no datos específicamente como es necesario en este caso.

Al no contar con un sistema informático, que englobe todas las funcionalidades que esta fundación requiere, se les complica tener integrado toda la información del paciente, la ficha médica y el inventario de equipos que posee la fundación. Es así que de los 300 pacientes que se atienden poseen información de 107 en el sistema.

3. Definición del Problema

En la actualidad se convierte en un problema diario, el buscar la información de cada paciente para analizar su ficha médica. Los equipos que adquiere la fundación, se extravían por no llevar un verdadero control de inventario y de préstamos, además los pacientes que hacen uso de estos, afirman devolverlos, pero en realidad no lo hacen.

La página web, estará integrada al mismo tiempo con un sistema de información, el cual tiene acceso a través de usuario y contraseña. Contiene la ficha médica del paciente, el inventario de medicamentos y equipos, el control de préstamos de medidores de glucosa y administración de usuarios, así se podrá

ingresar y actualizar el sistema desde cualquier parte del mundo, difundiendo su labor de ayudar a la gente, ya que con este sistema podrían ir a otras ciudades y solo con una computadora e Internet, almacenarían los datos de los pacientes.

4. Objetivos

4.1 Objetivos Generales

Realizar un sistema de gestión web integral, que ayude a llevar el control de los pacientes, con información y ficha médica de cada uno de ellos, inventario de los equipos y medicamentos que se utilizan para dar el servicio, y controlar el préstamo de estos a los pacientes.

4.2 Objetivos Específicos

- Diseñar y desarrollar un sitio web que conste con información general y servicios que la fundación FUVIDA presta a la comunidad.
- Desarrollar e implementar un módulo de "Ficha Médica", en el cual la Fundación llevará un control y actualización de información almacenada en la ficha médica de un paciente.
- Desarrollar e implementar un módulo de "Control de Préstamos", donde la Fundación puede registrar y llevar el control de préstamo de medidores de glucosa.
- Desarrollar e implementar un módulo de "Inventario", que ayude en el control de entrega de los medicamentos y equipos que ingresan y salen de la fundación FUVIDA

5. Aplicación

5.1. Tecnologías

Para la elaboración total de la aplicación web y de la página web se utilizaron diversas herramientas de programación y lenguajes, de los cuales iremos describiendo a continuación su definición y aplicación en el sistema desarrollado.

Lenguajes de Programación

Entre los lenguajes de programación tenemos.

HTML, siglas de HyperText Markup Language (lenguaje de marcado de hipertexto) fundamentado del World Wide Web (Red Informática Mundial), hace referencia al lenguaje de marcado o lenguaje de

marcas que es una forma de codificar un documento [4], predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto mediante las cuales se determina la forma en la que debe aparecer en un navegador web. El HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). [3]



Figura 1 Gráfico HTML

HTML también puede describir, hasta en cierto punto, la apariencia de un documento, puede incluir script (JavaScript) y hojas de estilo en cascada (CSS – Cascade Style Sheet), que puede afectar el comportamiento de navegadores web y otros procesadores de HTML. El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, es fácil de entenderlo e interpretarlo, razón por la que se usó este lenguaje de marcado

JavaScript, es un lenguaje de programación interpretado, este se define como orientado a objetos, se utiliza principalmente del lado del cliente, implementado como parte de un navegador web, es decir el navegador lo interpreta, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. En nuestra aplicación se utiliza como parte fundamental del framework jQuery del que vamos a detallar mas adelante.[7]

jQuery, es una biblioteca o framework de JavaScript rápida y fácil de utilizar, que simplifica la interacción con documentos HTML, manejo de eventos, animación, y las interacciones Ajax para el desarrollo web rápido. jQuery está diseñado para cambiar la forma en que escribes JavaScript.[8]

jQuery.dataTables [13], se utiliza para la tabulación de datos, este script tiene las siguientes características:

- Paginación.
- Mostrar la cantidad de datos que queremos que se tabulen.

Hacer una búsqueda dentro de los datos tabulados.

Figura 2 Gráfico jQuery Table

jQuery.validate, se utiliza para la validación de los campos que se encuentren como obligatorios en los formularios. [12]

Figura 3 Gráfico JQuery Validate

Sliding.form, se utiliza para darle efecto de slides a los forms de acuerdo a la información que se este pidiendo.

Figura 4 Gráfico Sliding Form

Con este componente se ahorra mucho espacio y es de fácil acceso que básicamente funciona como una presentación de diapositivas, sólo que tenemos conjuntos de campos de un formulario en lugar de imágenes.[14]

Cascading Style Sheets (Hojas de estilo en cascada), del que toma sus siglas CSS, este es un lenguaje usado para definir la presentación y diseño de un documento estructurado escrito en lenguaje de marcado o como se conoce HTML. La idea de programar en este lenguaje es que el desarrollador Web separe la estructura de un documento de su presentación tomando el control del estilo y formato del documento, además nos permite adaptar la presentación a los diferentes tipos de dispositivos, tales como desktops, móviles, tablets, etc, siendo

CSS independiente de HTML lo cual hace que los sitios Web sean mantenibles. [6],[9]

MySQL, es un motor de base de datos SQL relacional, multihilo y multiusuario, rápido, y robusto.

MySql se encuentra bajo dos tipos de licencia una es la GNU GLP para uso libre y una comercial para empresas que quieran incorporarlos en productos privativos.

MySQL es muy utilizado en aplicaciones web de baja concurrencia, y es aceptado en varias plataformas, lo que la hace un motor de base de datos portable. MySQL está escrito en una mezcla de C y C++. En nuestra aplicación web existe baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicación.[10],[11]

Symfony, es un framework PHP para proyectos web basado en el patrón Modelo Vista Controlador, desarrollado para acelerar la creación y el mantenimiento de aplicaciones web PHP.

Uno de los objetivos principales es garantizar una separación de preocupaciones (Separation of Concerns), en informática, la separación de preocupaciones (SoC) es un principio de diseño para la separación de un programa de ordenador en distintas secciones, de tal manera que cada sección responde a la preocupación por separado, por ejemplo, diseños en capas en los sistemas de información es una forma de realización de la separación de preocupaciones (por ejemplo, capa de presentación, capa de lógica de negocio, la capa de acceso de datos, capa de base de datos).[15]

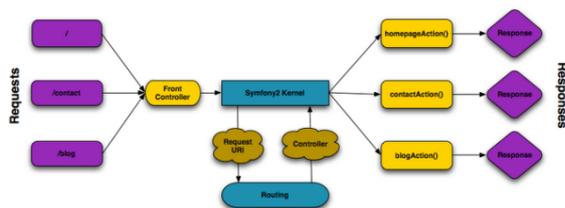


Figura 5 Gráfico Symfony [16]

5.2. Diseño e Implementación

Requerimientos Funcionales

Se refiere a todas las funcionalidades que debe satisfacer el sistema para cumplir con las necesidades de los usuarios.

El sistema debe cumplir los siguientes requerimientos agrupados por diferentes módulos:

Administrador	Usuario
Acceso al sistema mediante usuario y contraseña	Acceso al sistema mediante usuario y contraseña
Permite crear, editar y eliminar usuarios con diferentes roles para su acceso al sistema	Permite editar la información de propio usuario
Permite crear, editar y eliminar la información de los Pacientes	Permite crear y editar la información de los Pacientes.
Permite crear, editar y eliminar productos que ingresen en la Fundación	Permite crear y editar productos que ingresen en la Fundación
Permite crear, editar y eliminar los préstamos de los implementos médicos de la Fundación	Permite crear y editar los préstamos de los implementos médicos de la Fundación

Tabla 1. Requerimientos Funcionales

Requerimientos No Funcionales

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:

Rendimiento	Tiempos de respuesta del sistema aceptable Garantizar velocidad estable de navegación para los usuarios del sistema
Portabilidad	Por lo que el sistema es web y utilizará la tecnología actual, para funcionar tanto en los diferentes navegadores existentes
Mantenibilidad	Implementar un patrón de diseño de desarrollo, para el fácil mantenimiento del sistema
Seguridad	El acceso será controlado por los nombres de usuario y contraseñas
Usabilidad	El sistema debe usar principios de diseño de Interfaz Centrado en el Usuario, para que el affordance del sistema sea intuitivo

Tabla 2. Requerimientos No Funcionales

Arquitectura del Sistema

Cliente-Servidor: Es un modelo de aplicación distribuida, esta arquitectura como su nombre lo indica consta básicamente de dos partes, la parte del servidor que es el encargado de proveer servicios o recursos; y la parte cliente que consume dichos servicios. En la siguiente figura podemos apreciar de manera general su funcionamiento.

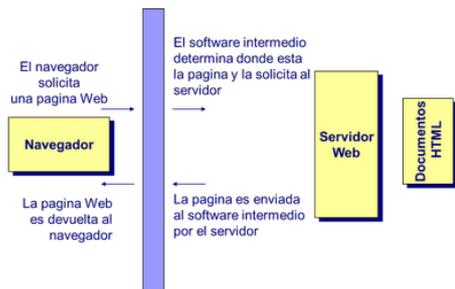


Figura 6 Gráfico Cliente - Servidor [2]

La comunicación comienza cuando el cliente hace una petición y el servidor que está esperando por ellas, las procesa y envía una respuesta. [1],[2]

Modelo Vista Controlador: Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción para el desarrollo de aplicaciones, que se encarga de separar los datos, la interfaz del usuario y la lógica del negocio, los mismo que son llamados respectivamente modelo, vista y controlador. Este modelo es muy utilizado en las aplicaciones Web, de manera general se puede hacer una comparación y decir que la vista es la página HTML, el modelo es el Sistema de Gestión de Base de Datos, y el controlador es el encargado de recibir los eventos y resolverlos, dichos eventos son enviados por el cliente.

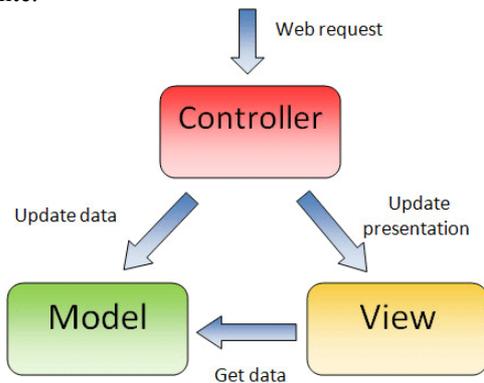


Figura 7 Gráfico Modelo Vista Controlador [17]

Diseño de Módulos

Los módulos del sistema han sido definidos considerando que el Sistema tendrá una arquitectura basada en Cliente Servidor, con un patrón Modelo Vista Controlador, de manera que cada uno de ellos satisfaga los requerimientos especificados en el capítulo anterior.

Cada módulo contendrá vistas, modelos y controladores específicos, que serán los encargados de realizar las tareas correspondientes, logrando de esta manera el correcto funcionamiento del mismo.

A continuación una breve descripción de cada uno de los módulos

Módulo Paciente

Este módulo será la aplicación que manejará como su nombre lo indica los pacientes.

Está encargado de realizar el ingreso, modificación y eliminación de información de los distintos pacientes que tiene la fundación. Adicionalmente podrá imprimir la ficha médica, la cual consta en los datos ingresados del paciente.

Esté módulo puede ser accedido por los 2 distintos roles que tiene el sistema, el Administrador y el Usuario.

Módulo Producto

En el módulo producto, está todo lo que consiste en la administración del producto. Se puede ingresar un nuevo producto, modificar y eliminarlo.

Al momento de que llegue nueva mercadería el producto puede ser ingresado, modificando la cantidad en los productos ya existentes. Este módulo puede ser accedido por los 2 distintos roles que tiene el sistema Administrador y Usuario.

Módulo Préstamo

El módulo “préstamo”, incluye la parte fundamental del sistema que es el control de préstamos, ventas y donaciones de los productos o implementos que tiene la fundación.

En este módulo los préstamos pueden ser ingresados, modificados y eliminados y estas funciones las puede realizar cualquiera de los 2 roles que tiene el sistema: Administrador y Usuario.

Módulo Usuario

El módulo Usuario es el encargado de Administrar los distintos usuarios que pueden acceder a la aplicación.

Al momento de crearlo puede elegir el rol con el cual puede acceder, Administrador o Usuario. En cualquier momento un usuario puede ser modificado o eliminado.

A este módulo solo pueden acceder los Administradores a todas sus funciones, mientras que los que tienen el rol Usuario solo pueden modificarse a sí mismos.

6. Conclusiones

1. FUVIDA, con el nuevo sistema puede viajar a distintas provincias a entregar insumos y equipos, llevando de igual manera un control de los préstamos y pacientes.
2. FUVIDA, puede tener centralizada la información de pacientes, inventario y préstamos de los insumos.
3. FUVIDA, tiene un control de los préstamos de equipos, y disminuye la pérdida de estos, beneficiando a más pacientes.
4. FUVIDA, puede modificar y consultar los datos de los pacientes sin necesidad de volver a ingresar los datos de los pacientes y reimprimir la ficha médica.

7. Recomendaciones

1. FUVIDA, debe migrar la información que tienen almacenada en papeles de los pacientes.
2. FUVIDA, debe recordar que la aplicación del sistema está en internet por tal motivo no podrán usarla si no tienen acceso a internet.
3. FUVIDA, debe recordar que el sistema fue desarrollado en Symfony y la página web en Joomla, por lo tanto si desean agregar algún nuevo módulo o funcionalidad deben de considerar que sea compatible con estas herramientas.
4. FUVIDA, debe de recordar que el sistema y la página web están en un servicio de alojamiento web, y así como el dominio www.fuvida.org.ec, deben de renovarlo anualmente.
5. Se recomienda a Vínculos con la Colectividad solicitar a distintas fundaciones proyectos que necesiten y puedan realizarlo estudiantes de la ESPOL.
6. Se recomienda a FUVIDA, así como nosotros desarrollamos este proyecto, por medio del departamento de Vínculos con la Colectividad de ESPOL, tengan asesoría de estudiantes de Licenciatura en Nutrición, para desarrollar planes de dieta y nutrición de los pacientes.

8. Referencias

[1] Cliente-Servidor, Wikipedia – Recuperado el 9 de Enero de 2013 en:

<http://es.wikipedia.org/wiki/Cliente-servidor>

[2] Luis Eduardo (14 de noviembre de 2012), Ingeniería del Software – Recuperado el 16 de Enero de 2013 en:

<http://ingsoftluisf.blogspot.com/2012/11/cliente-servidor.html>

[3] HTML, Wikipedia – Recuperado el 9 de Enero de 2013 en: <http://es.wikipedia.org/wiki/HTML>

[4] Lenguaje de Marcado, Wikipedia – Recuperado el 9 de Enero de 2013 en:

http://es.wikipedia.org/wiki/Lenguaje_de_marcado

[5] World Wide Web, Wikipedia – Recuperado el 9 de Enero de 2013 en:

http://es.wikipedia.org/wiki/World_Wide_Web

[6] Hojas de estilo en cascada, Wikipedia – Recuperado el 9 de Enero de 2013 en:

http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada

[7] JavaScript, Wikipedia – Recuperado el 9 de Enero de 2013 en:

<http://es.wikipedia.org/wiki/JavaScript>

[8] JQuery, Wikipedia – Recuperado el 9 de Enero de 2013 en: <http://es.wikipedia.org/wiki/JQuery>

[9] What is CSS, W3c – Recuperado el 9 de Enero de 2013 en:

<http://www.w3.org/standards/webdesign/htmlcss#whattocss>

[10] MySQL, Wikipedia – Recuperado el 9 de Enero de 2013 en: <http://es.wikipedia.org/wiki/MySQL>

[11] MySQL Enterprise Edition, Oracle – Recuperado el 9 de Enero de 2013 en:

<http://www.mysql.com/products/enterprise/>

[12] Jörn Zaefferer (2006), jQuery team – Recuperado el 16 de Enero de 2012 en:

<http://docs.jquery.com/Plugins/Validation>

[13] SpryMedia, DataTables – Recuperado el 16 de Enero de 2013 en: <http://www.datatables.net/>

[14] MARY LOU (June 7, 2010), Fancy Sliding form with JQuery – Recuperado el 16 de Enero de 2013 en:

<http://tympanus.net/codrops/2010/06/07/fancy-sliding-form-with-jquery/>

[15] The Big Picture, Symfony – Recuperado el 9 de Enero de 2013 en:

http://symfony.com/doc/current/quick_tour/the_big_picture.html

[16] The Symfony2 and HTTP Fundamentals – Recuperado el 9 de Enero de 2013 en:

http://symfony.com/doc/current/quick_tour/the_big_picture.html

[17] MVC – Model View Controller – Recuperado el 9 de Enero de 2013 en:

<http://www.phpfactory.com/blog/mvc-model-view-controller/>