

Recepción: 31 de octubre de 2013

Aceptación: 23 de enero de 2014

Publicación: 27 de marzo de 2014

SISTEMA INFORMÁTICO PARA LA EVALUACIÓN DE ATRIBUTOS DE CALIDAD EN COMPONENTES BIOMÉTRICOS

**COMPUTER SYSTEM TO EVALUATE QUALITY ATTRIBUTES
IN BIOMETRICS COMPONENTS**

Ing. Grettel Susel Incencio Piñeiro¹

1. Ingeniera en Ciencias Informáticas, Facultad Regional de la Universidad de Granma. Especialista de Experiencia de Usuario en el departamento de Soluciones de Gestión. Cuba. E-mail: gsusel@grm.uci.cu

RESUMEN

La utilización de sistemas informáticos que permitan automatizar los procesos de aseguramiento de la calidad en proyectos de desarrollo de software, garantizan el ahorro de tiempo y dinero, así como la disminución de los posibles errores que se puedan producir debido al proceder humano en dichas evaluaciones. Este documento describe el proceso de desarrollo de una aplicación web, que permite automatizar el proceso de evaluación de atributos de calidad en componentes biométricos en el Centro de Identificación y Seguridad Digital (CISED).

ABSTRACT

The use of systems or applications that allows to automate quality assurance processes in software development projects, guaranteed to save time and money as well as reducing the possible errors that may occur due to human behavior in such assessments. This document describes the process of developing a web application that automates the process of evaluating quality attributes in biometric components at the Identification and Digital Security Center (CISED).

PALABRAS CLAVE

Atributos de calidad, automatizar, componentes biométricos, aplicación web.

KEYWORDS

Quality attributes, automate, biometric components, web application.

INTRODUCCIÓN

La industria de software ha ocupado un lugar prominente dentro del mercado internacional, donde los productos de software ofertados deben poseer la calidad como cualidad inherente, factor determinante en el éxito de un producto. Si alguien dice que la calidad no le da ventaja a una organización, entonces se equivoca. Cualquier institución que deja de lado este importante concepto en el desarrollo del ciclo de vida del software, puede destruir su propia imagen y el valor de la empresa.

En el Centro de Identificación y Seguridad Digital (CISED) perteneciente a la Universidad de las Ciencias Informáticas (UCI), existen numerosos proyectos de desarrollo de software. Estos proyectos están agrupados en departamentos que desarrollan las líneas de investigación: Identificación, Biometría, Tarjetas Inteligentes, Seguridad Digital y Soluciones Integrales.

En el caso específico del Departamento de Biometría se han encontrado algunas dificultades, pues en ocasiones los atributos del software especificados por un usuario final durante la fase de análisis de los requisitos, no satisfacen los requisitos del usuario cuando el producto está en explotación, debido a cambios en los requisitos del usuario y a la dificultad de especificar necesidades implícitas. En este sentido se hace necesaria su evaluación, en aras de garantizar la calidad y fiabilidad posible del producto obtenido, sobre todo si se tiene en cuenta que esta línea productiva se especializa en una rama de la informática relativamente nueva en el centro, por lo que la experiencia acumulada, tanto en el aseguramiento de calidad como en el desarrollo de software de estas características, es aún insuficiente.

Realizar la evaluación de atributos de calidad en componentes biométricos de forma manual se torna complicado, por esta razón se decide la implementación de un sistema informático que permita automatizar el proceso. Dicha aplicación minimizará la posibilidad de que se introduzcan errores en la realización de futuras evaluaciones a dichos componentes.

METODOLOGÍA, TECNOLOGÍA Y HERRAMIENTAS

A continuación se describen las tecnologías, herramientas y metodología utilizadas en el desarrollo de la aplicación. Como resultado de los estudios e investigaciones realizadas, se propone la implementación de una aplicación web para la realización del sistema informático que se pretende, ya que son tecnologías actualmente muy prometedoras, seguras y eficaces.

Además se propone la arquitectura cliente/servidor ya que está estrechamente relacionada con las aplicaciones web y cuenta con una serie de ventajas que le atribuyen gran importancia. Entre sus principales ventajas se pueden mencionar la posibilidad de existir varios usuarios nutriéndose de información simultáneamente, así como mayor rapidez para el mantenimiento y el desarrollo de aplicaciones; además de la facilidad de integración entre diferentes sistemas, el intercambio de información, entre otras.

LENGUAJE DE PROGRAMACIÓN

En la actualidad existe una gran variedad de lenguajes de programación para aplicaciones web. Se define su uso de acuerdo a la plataforma y gestor de base de datos que se desee utilizar. Dichos lenguajes se clasifican en dos partes fundamentales: los lenguajes del lado del servidor y los lenguajes del lado del cliente. Entre los lenguajes del lado del servidor más sobresalientes están: Perl, ASP (Active Server Pages), PHP (Hypertext Pre-Processor), Java, entre otros. Éstos se caracterizan por desarrollar la lógica de negocio dentro del servidor, además de ser los encargados del acceso a bases de datos, tratamiento de la información, etc. Del lado del cliente se encuentra principalmente el JavaScript, encargado de aportar dinamismo a la aplicación en los navegadores. El sistema empleará como lenguaje de programación del lado del servidor PHP en su versión 5.3.8.

PHP (acrónimo de PHP Hypertext Pre-processor), es un lenguaje interpretado de alto nivel que se ejecuta del lado del servidor. Este lenguaje permite a los desarrolladores la rápida creación dinámica de páginas web. Con PHP se puede hacer casi cualquier cosa, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas web dinámicas. También ofrece la integración con varias bibliotecas externas, incorporando una gran cantidad de funciones para realizar útiles tareas relacionadas con la web, dígase generar documentos en PDF hasta analizar código XML, entre otras.

PHP ofrece además una solución simple para las paginaciones dinámicas de la web de fácil programación. Su diseño lo hace más fácil de mantener y ponerse al día en comparación con el código de otros lenguajes. Debido a su amplia distribución y que es un producto de código abierto, PHP está soportado por una gran comunidad de desarrolladores, y goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades [1].

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux,

Microsoft Windows y Mac OS. Además, soporta la mayoría de los servidores web de hoy en día.

SISTEMA GESTOR DE BASE DE DATOS

Los gestores de base de datos son sistemas formados por un conjunto de datos y un paquete de software para la gestión del mismo, de modo que se controla el almacenamiento de datos redundantes; éstos resultan independientes de los programas que los usan y se puede acceder de diversas formas [2]. El sistema emplea como sistema gestor de base de datos (SGBD) MySQL en su versión 5.5.16.

MySQL es un sistema para la administración de bases de datos relacional rápido y sólido. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos y para asegurarse de que sólo obtienen acceso a ellos los usuarios con autorización.

Por lo tanto, MySQL es un servidor multiusuario y de subprocesamiento múltiple. Utiliza SQL (del inglés Structured Query Language), como lenguaje estándar para las consultas de bases de datos utilizado en todo el mundo. MySQL se distribuye bajo una licencia de código abierto en la actualidad, pero también existen licencias comerciales. MySQL cuenta con muchas otras ventajas, entre las que se encuentran las siguientes: [1]

- Alto rendimiento y bajo costo.
- Facilidad de configuración y aprendizaje.
- Portabilidad.
- Accesibilidad a código fuente.
- Soporta gran variedad de sistemas operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

METODOLOGÍA DE DESARROLLO DE SOFTWARE

El desarrollo del software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte están aquellas propuestas más tradicionales, que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Las metodologías tradicionales han demostrado ser muy efectivas en muchos proyectos, pero no en su totalidad. Por otro lado, existe la filosofía de las metodologías ágiles, centrada en otras dimensiones, como por ejemplo, el factor humano o el producto de software, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

No existe una metodología absoluta, por lo cual se debe tener previo conocimiento de las características del proyecto antes de seleccionar la metodología que será utilizada, debido a que la realización de esta actividad constituye un factor determinante en el éxito del

proyecto. Para el desarrollo del software se decidió utilizar metodologías ágiles, específicamente la Programación Extrema (XP).

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. El ciclo de vida de XP es iterativo y define cuatro fases fundamentales: exploración, planificación, implementación y pruebas. Lo esencial en este proceso de desarrollo es lograr la comunicación entre desarrolladores y usuarios, la retroalimentación entre ellos y con los usuarios finales y la simplicidad en el código. Entre los elementos determinantes para su elección se encuentran:

- El tamaño del grupo de desarrollo, en este caso, de dos personas.
- Necesidad de resultados tangibles a corto plazo.
- Imposibilidad, para un grupo de desarrollo pequeño, de asumir una metodología robusta, debido a la cantidad excesiva de roles y documentación generada durante el ciclo de vida del proyecto.

LENGUAJE DE MODELADO

En la modelación de la aplicación, se utilizará el Lenguaje Unificado de Modelado (UML por sus siglas en inglés). Este es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML es un lenguaje estándar para el modelado de software utilizado para visualizar, especificar y documentar los artefactos del sistema. Permite a los desarrolladores visualizar el producto de su trabajo (artefacto) en esquemas o diagramas estandarizados.

UML ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de las piezas que constituirán el modelo [3].

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas, generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica qué es lo que supuestamente hará el sistema pero no cómo lo hará [4].

UML es la interrelación Elementos – Relaciones – Diagramas, es un modelo gráfico que incluye aspectos conceptuales (negocio y sistema) y aspectos concretos (expresiones del lenguaje de programación, esquema de base de datos y componentes de software reutilizables). Se emplea para definir un sistema, sus artefactos, documentos y construir el software.

UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre bloques (asociación, generación) y diagramas (por ejemplo el diagrama de actividades). Los perfiles de UML son las extensiones a las notaciones estándares del UML utilizando los mecanismos de extensión del UML: los estereotipos, los valores etiquetados y restricciones [5]. Se escoge UML porque:

- Permite modelar sistemas mediante técnicas orientadas a objetos.
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

HERRAMIENTA CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son aplicaciones informáticas con el objetivo de aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero; ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. Como herramienta CASE para el modelado de la aplicación se utiliza Visual Paradigm.

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Ordenador, del inglés Computer Aided Software Engineering) que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo de desarrollo de un software, desde la fase de análisis hasta el despliegue del mismo. Permite realizar ingeniería directa o inversa sobre el software y es capaz de, a partir de un modelo relacional en diferentes sistemas gestores de base de datos, desplegar todas las clases asociadas a las tablas.

Visual Paradigm está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros (Team Project). Se propone Visual Paradigm como herramienta a utilizar, debido a que posibilita la representación gráfica de los diagramas que permiten ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clases, actividades, estado, entre otros.

Además, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo a través de un lenguaje gráfico [6].

ENTORNO DE DESARROLLO INTEGRADO

Un entorno de desarrollo integrado IDE (del inglés Integrated Development Environment), está compuesto por un conjunto de herramientas para agilizar el proceso de desarrollo del programador. Un IDE consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Estos pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. Además, proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación. Para la implementación del sistema se seleccionó NetBeans IDE 7.2.

El IDE NetBeans es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y para móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, C/C++, y otras. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de forma formación exhaustivos, así como una amplia selección de complementos de terceros [7].

FRAMEWORK DE DESARROLLO

Un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting, además de otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Este representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Además proporciona estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener. Para el desarrollo del sistema se seleccionó el framework Symfony en su versión 1.4.18.

Symfony es un framework PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes de las aplicaciones web, dejando que el programador se dedique a aportar valor agregado, desarrollando las características únicas de cada proyecto.

También aumenta la productividad y ayuda a mejorar la calidad de las aplicaciones web aplicando todas las buenas prácticas y patrones de diseño que se han definido para la web. Es además el framework más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. Las principales características que posee este framework son las siguientes [8]:

- Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server).

- Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- Basado en la premisa de convenir en vez de configurar, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins.
- Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.
- Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma.

SERVIDOR WEB

Un servidor web o servidor HTTP es un programa que procesa cualquier aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo; generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. Para hospedar la aplicación se utilizará el servidor web Apache 2.2.21.

En la actualidad el servidor web Apache es el más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. El servidor Apache es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor web. El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software [9]. Para realizar la instalación del servidor Apache de forma más sencilla se usará el servidor independiente de plataforma XAMPP (X: cualquier plataforma, Apache, MySQL, PHP, Perl).

DESARROLLO DEL SISTEMA

Como propuesta de solución al problema científico se propone el desarrollo de una aplicación web que garantice el proceso de evaluación de los atributos de calidad a los componentes biométricos del Centro de Identificación y Seguridad Digital (CISED).

La aplicación contará con dos tipos de usuarios principalmente. El administrador del sistema será el encargado de formalizar toda la información referente al sistema y posibilitará la actualización de todas las informaciones contenidas en el mismo. El evaluador será el encargado de seleccionar el componente biométrico que desea evaluar, una vez seleccionado el mismo podrá realizar la evaluación del componente guiándose por una serie de parámetros que le permitirán realizar una evaluación satisfactoria. Además éste podrá ver evaluaciones realizadas a este u otros componentes en otras ocasiones, pero no podrá modificar los datos de las mismas.

La base de datos registrará de forma unificada todas las evaluaciones realizadas, manteniendo así la base de datos de la aplicación actualizada con la información necesaria, garantizando la estabilidad y fácil mantenimiento del sistema.

La aplicación toma como punto de partida un Modelo de Calidad para componentes biométricos previamente propuesto. El mismo cuenta con una serie de características de calidad, las cuales a su vez fueron refinadas en sub-características de calidad, según propone la norma ISO/IEC 9126 [10]. Igualmente la propuesta de métricas para dichas subcaracterísticas se realizó tomando las que resultaran más significativas pues en ocasiones existían varias métricas que podían ser utilizadas, pero se optó por las menos complejas, donde el significado de las variables comprendidas en las fórmulas de medición fueran las más claras.

A continuación se elaboró una lista de chequeo que consta de preguntas correspondientes a cada característica y sub-característica de calidad definidas anteriormente, las cuales están escritas de forma clara y de fácil entendimiento. Esta lista de chequeo permitirá definir a través de una gráfica, el nivel de calidad del componente biométrico. Para otorgar una evaluación a cada pregunta de la lista de chequeo, basado en la valoración del resultado obtenido de la métrica, se toma como base la siguiente tabla:

Rango del resultado	Criterio de evaluación
0 - 0.2	2 puntos
0.3 - 0.5	3 puntos
0.6 - 0.8	4 puntos
0.9 - 1	5 puntos

Tabla I. Escala de métricas - preguntas de chequeo. Fuente: Elaboración propia

En el caso que el resultado de las métricas es más factible mientras más cerca está a cero (2 casos según las métricas propuestas), solo es necesario aplicar el sistema de conversión en el sentido inverso.

Finalmente para otorgar una evaluación al componente biométrico basado en los resultados obtenidos, la misma se emite tomando como base la siguiente tabla:

Rango del resultado	Nivel de calidad
2 - 2.9	No adecuado
3 - 3.9	Medianamente adecuado
4 - 5	Adecuado

Tabla II. Niveles de calidad del componente. Fuente: Elaboración propia

FUNCIONALIDADES DEL SISTEMA

- **Autenticar usuario y administrar usuario:** son las funcionalidades base sobre las cuales se sustentan la seguridad del sistema. Para realizar cualquier acción es necesario identificarse previamente.
- **Gestionar componentes biométricos:** para gestionar los componentes biométricos que pueden ser evaluados.
- **Gestionar atributos de calidad:** permite gestionar los atributos de calidad que se desea evaluar en los distintos componentes.
- **Gestionar subcaracterísticas de calidad:** permite gestionar las subcaracterísticas de calidad que se desea evaluar en los distintos componentes.
- **Gestionar lista de chequeo:** para establecer las preguntas y métricas de software asociadas a los atributos y subcaracterísticas de calidad.
- **Evaluar componente biométrico:** es la funcionalidad principal del sistema. Permite evaluar una o varias características de calidad en determinado componente, introduciendo los datos obtenidos de las mediciones realizadas.
- **Ver evaluaciones realizadas:** permite ver los detalles de las evaluaciones realizadas y obtener distintos reportes de las mismas, principalmente a formato pdf.

REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales son propiedades o cualidades que debe tener el producto, las cuales representan las características que hacen al producto atractivo, rápido y confiable. Seguidamente se especifican los requerimientos no funcionales del sistema propuesto.

Apariencia o interfaz externa:

- Optimizado para una resolución de 1024x768.
- Interfaz amigable y sencilla, con colores suaves a la vista y sin cúmulo de objetos que le sean incómodos al usuario a la hora de interactuar con el sistema.
- Un menú principal que le brinde al usuario rapidez y facilidad para obtener la información que desea.

Software para el cliente:

- Sistema operativo multiplataforma con interfaz gráfica.
- Navegador web (Internet Explorer 6.0 o superior, Mozilla Firefox 3.0 o superior, Opera, Safari).

Software para el servidor:

- Framework Symfony 1.4.10 o superior.
- Servidor web Apache 2.2.12 o superior.
- Versión de PHP 5.3.0 o superior.
- Gestor de Base de Datos MySQL 5.1.37 o superior.
- Software controlador de versiones: CollabNetSubversion-client-1.6.15 o superior.

Hardware para el desarrollo:

- PC con las siguientes características: Intel Pentium 4 o superior, CPU 3GHz o superior, 1GB RAM o superior, 80 GB HDD o superior.

Hardware para explotación del cliente:

- PC con las siguientes características: Intel Pentium 4 o superior, CPU 3GHz o superior, 256 MB RAM mínimo, 512 RAM recomendada o superior.

Hardware para explotación del servidor:

- PC con las siguientes características: Intel Pentium 4 o superior, RAM: 1 GB mínimo (Recomendado 2 GB), 80 GB de disco duro.

Usabilidad:

- Para hacer uso del sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente web en sentido general.

- El diseño debe tener buena visibilidad en los principales navegadores.
- El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación tales como menús laterales y/o desplegables que permitan el acceso rápido a la información.

Soporte:

- Sistema multiplataforma.
- Los sistemas deberán estar abiertos a las modificaciones que se requieran, ya sea ante la detección de un fallo o por nuevos requisitos. Después de su puesta en explotación se le dará mantenimiento una vez al año.

Seguridad:

- Chequear que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.
- La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al tipo de usuario que esté activo.
- Realizar salvallas periódicas de la información y base de datos en otros dispositivos, como solución ante la ocurrencia de problemas.
- Llevar un registro de sucesos donde se archiven los eventos del sistema incluyendo los eventos de error, inicio de sesión, cierre de sesión y modificación de la información.

Disponibilidad:

- El sistema deberá tener un 100% de disponibilidad por lo que podrá ser usado las 24 horas del día por todos sus clientes.

SEGURIDAD DEL SISTEMA

La seguridad es uno de los aspectos más importantes a tener en cuenta a la hora de proteger la información. Actualmente Symfony propone numerosas estrategias y utilidades para hacer frente a los ataques XSS¹ y CSRF² con el objetivo de mantener la integridad, confidencialidad y disponibilidad de la información.

Algunas de estas estrategias son: las opciones `allow_extra_fields` y `filter_extra_fields` las cuales permiten controlar si todos los campos del formulario deben tener asignado un validador y si se deben filtrar todos los campos adicionales, por lo que no tienen efecto los ataques que intentan inyectar campos que no pertenecen al formulario original. Por otra parte se ha creado un plugin llamado `pdCSRFPlugin` para solucionar los problemas contra ataques de tipo CSRF.

DISEÑO

La arquitectura de la aplicación web se basa en la arquitectura que implementa el framework Symfony, el cual está basado en el patrón arquitectónico Modelo - Vista - Controlador (MVC), ya que separa la vista (interfaz) y el modelo (base de datos) mediante el controlador que es el encargado de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o en la vista, como muestra la figura 1.

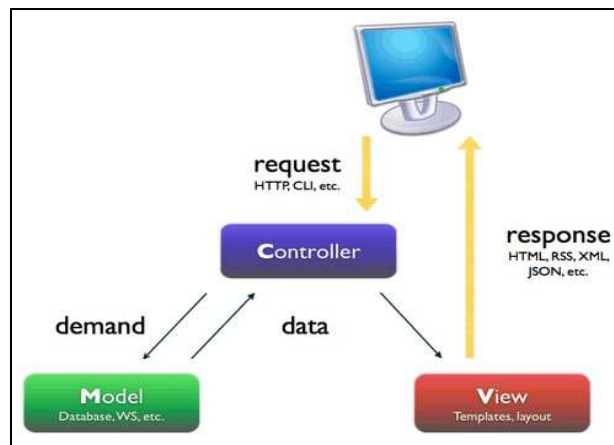


Figura 1: Patrón arquitectónico MVC. Fuente: [11].

El modelo representa la estructura de datos. Las clases modelo contienen funciones para obtener, insertar, eliminar y actualizar información en la base de datos. La vista es la información presentada al usuario; normalmente será una página web, también puede ser un fragmento de una página como un encabezado, pie de página o RSS (Really Simple

¹ **XSS:** del inglés Cross-site scripting, es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.

² **CSRF:** del inglés Cross-site request forgery: es un tipo de exploit en el que comandos no autorizados son transmitidos por un usuario que el sitio web confía.

Syndication). El controlador es un intermediario entre el modelo, la vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web [11].

La aplicación estará desplegada en un solo servidor junto con la base de datos. Las estaciones de trabajo se conectarán a la aplicación mediante el protocolo HTTP de acuerdo a como se muestra en la figura 2.

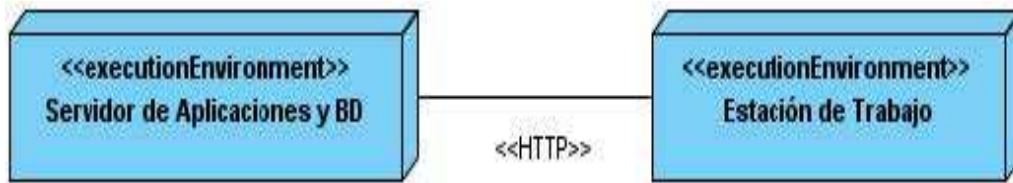


Figura 2. Diagrama de despliegue. Fuente: Elaboración propia

CONCLUSIONES

Una vez concluido el proceso de desarrollo del sistema se puede concluir que: el desglose de las historias de usuario en tareas de la ingeniería permitió facilitar la implementación de las funcionalidades, asimismo la utilización del marco de trabajo Symfony facilitó la correcta implementación del patrón arquitectónico Modelo-Vista-Controlador, asegurando la aplicación contra ataques comunes.

Este trabajo finaliza con la obtención de una aplicación web que dotó al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas de una aplicación web, que permite la evaluación de los atributos de calidad de los componentes biométricos a través de la lista de chequeo establecida en el Modelo de Calidad obtenido para este tipo de productos software.

Este resultado ha sido realizado de manera flexible ante posibles cambios en el proceso de desarrollo de software de la organización donde se aplica, proponiendo tecnologías libres acordes con las políticas que existen en el país y en la UCI para lograr una independencia tecnológica.

REFERENCIAS

1. Welling, Luke y Thomson, Laura. *Desarrollo web con PHP y MySQL*. Madrid : Anaya, 2003.
2. Matamoros, Soldiamar. *Bases de Datos. Introducción a la Informática*. 2011. Disponible en: <http://www.dspace.espol.edu.ec/bitstream/123456789/1416/1/2751.pdf>.
3. Booch, Grady, Rumbaugh, James y Jacobson, Ivar. *El Lenguaje Unificado de Modelado. Manual de referencia*. Madrid : Pearson Educación, 2000. 84-7829-036-2.
4. Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. 970-17-0261-1.
5. Pressman, Roger S. *Ingeniería de Software, un enfoque práctico*. 5th. La Habana : Félix Varela, 2005.
6. Visual Paradigm. *Visual Paradigm for UML*. [En línea] 2011. [Citado el: 17 de Octubre de 2011.] Disponible en: <http://www.visual-paradigm.com/product/vpum/>.
7. NetBeans.org. *Sitio web oficial NetBeans*. [En línea] 2011. [Citado el: 18 de Octubre de 2012.] Disponible en: http://netbeans.org/index_es.html.
8. Eguiluz, Javier. *Symfony.es*. [En línea] 2011. [Citado el: 18 de Octubre de 2012.] Disponible en: <http://www.symfony.es/que-es-symfony/>.
9. Ciberaula. *Una introducción a Apache*. [En línea] 2010. [Citado el: 18 de Octubre de 2012.] Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/.
10. ISO/IEC 9126:1. *Software engineering—Product quality—Part 1: Quality Model*. 2005.
11. Vázquez Acosta, Manuel. *Impacto de implementaciones web del patrón MVC en los requisitos de calidad percibidos*. Arequipa : Latin American and Caribbean Journal of Engineering Education, 2010. 1935-02-95.