ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"HERRAMIENTA WEB PARA EL DIAGNÓSTICO DE VULNERABILIDADES DE SEGURIDAD Y PRUEBAS DE PENETRACIÓN"

INFORME DE PROYECTO DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN TELEMÁTICA

Presentado por:

JOSÉ OLMEDO BEDÓN SÁNCHEZ

Guayaquil – Ecuador

2015

AGRADECIMIENTO

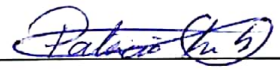
Agradezco el constante apoyo de mis padres que han sido mi inspiración para realizar este trabajo, a todos mis profesores.

DEDICATORIA

Dedicada a Dios, a mis padres José y Azucena, a mi hermano David, a mis familiares, a mis profesores y a mis amigos.

TRIBUNAL DE SUSTENTACIÓN

MSc. Sara Ríos Orellana
SUBDECANA DE LA FIEC



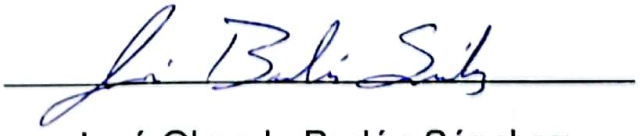
MSc. Patricia Chávez Burbano
DIRECTORA DEL PROYECTO DE GRADUACIÓN

MSc. Gonzalo Luzardo
MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Informe me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral."

(Reglamento de Graduación de la ESPOL)



José Olmedo Bedón Sánchez

RESUMEN

En este trabajo se implementó una herramienta, cuya función principal es de realizar pruebas de penetración de algún sistema en particular, en base a normas de seguridad informática como OSSTMM, uno de los estándares más usados; realizando el análisis de una de las capas de esta metodología para poder generar un informe detallado de las posibles vulnerabilidades del sistema analizado en ese aspecto específico. Previamente se realizó un análisis completo, de todas las herramientas externas que se necesitarían, para poder garantizar que esta herramienta sea suficientemente segura como para realizar un análisis de prueba de penetración de seguridad informática.

Este trabajo está dividido en 4 capítulos. El primer capítulo está enfocado en la descripción del problema, así como de las posibles medidas de seguridad que se puedan implementar, así como también como los objetivos principales para la realización de este trabajo.

El segundo capítulo detalla algunos estándares de seguridad informática, principalmente la metodología OSSTMM, así como las distintas herramientas externas a utilizar, realizando un análisis sobre los distintos frameworks

disponibles para facilitar la implementación de la herramienta y también sobre la arquitectura que tiene la herramienta.

El tercer capítulo contiene la implementación de la herramienta, es decir el diseño modular del mismo, así como el estudio de la base de datos, toda esta información está debidamente documentada para un futuro acoplamiento de algún modulo, y así mejorar el análisis de la prueba de seguridad, obteniendo un mejor informe final.

El cuarto capítulo consta de las pruebas realizas con distintos sistemas, así como una comparativa con otras herramientas que realizan la misma función como son Dradis, Magic Tree entre otras. También contiene los resultados obtenidos de cada análisis con la herramienta.

ÍNDICE GENERAL

ABREVIATURAS

ACL	Lista de Control de Acceso	Access Control List
ARIN	Registro Regional de Internet para América Anglosajona	American Registry for Internet Numbers
ASP	Proveedor de Servicios de Aplicación	Application Service Provider
BSD	Distribución de Software Berkeley	Berkeley Software Distribution
COBIT	Control Objectives for Information and Related Technology	Control Objectives for Information and Related Technology
COMSEC	Seguridad de las Comunicaciones	Communications Security
CRUD	Crear, Consultar, Actualizar y Eliminar	Create, Read, Update and Delete
DMZ	Demilitarized Zone	Zona desmilitarizada
DoS	Denegación de Servicio	Denial of Service
DRY	No lo repita usted mismo	Don't Repeat Yourself
FTP	Protocolo de Transferencia de Archivos	File Transfer Protocol

HTML	lenguaje de Marcas de Hipertexto	HyperText Markup Language
HTTP	Portocolo de Transferencia de Hipertexto	Hypertext Transfer Protocol
HUMSEC	Seguridad Humana	Human Security
ICMP	Protocolo de Mensajes de Control de Internet	Internet Control Message Protocol
IDS	Sistemas de Detección de Intrusiones	Intrusion Detection System
IEC	Comisión Electrotécnica Internacional	International Electrotechnical Commission
ISECOM	Instituto para la Seguridad y Metodologías Abiertas	Institute for Security and Open Methodologies
ISO	Organización Internacional de Normalización	International Organization for Standardization
ISP	Proveedor de servicios de Internet	Internet Service Provider
IT	Tecnología de la información	Information technology
MVC	Modelo-Vista-Controlador	Model-View-Controller
ORM	Mapeo Objeto-Relacional	Object-relational mapping

OS	Sistema Operativo	Operating System
OSSTMM	Manual de la Metodología Abierta de Testeo de Seguridad	Open Source Security Testing Methodology Manual
PHYSSEC	Seguridad Física	Physical Security
PSI	Pruebas de Seguridad Informática	---
SGBDR	Sistema de Gestión de Base de Datos Relacional	Relational Database Management System
SPECSEC	Seguridad del Espectro	Spectrum Security
TCP	Protocolo de Control de Transmisión	Transmission Control Protocol
UDP	Protocolo de Datagrama de Usuario	User Datagram Protocol
WAMP	---	Windows - Apache - Mysql - PHP

GLOSARIO DE TÉRMINOS

Analista: Es el encargado de desarrollar programas desde su diseño y obtención de algoritmos, y lo que respecta al análisis de utilidades y optimizaciones del mismo

Aplicación: Es un programa (muchas veces multiplataforma) que se emplea como herramienta que un usuario usa en su vida cotidiana o laboral.

Arquitectura: Forma de estructurar desde sus cimientos un software, programa o Sistema Operativo.

Cliente: Persona o empresa que utiliza los servicios que proveen otras empresas.

Consultor: Persona encargada de aconsejar a otras en el uso de IT.

Desarrollador: Persona que se dedica al proceso de desarrollo de software, interviene en su diseño creación y programación desde cero.

Framework: Estructura que usan los IDE de desarrollo incluye bibliotecas y otras herramientas que se usan en proyectos.

Introspección: Observación, inspección que se hace uno mismo, para evaluar los pensamientos propios

Multiplataforma: Que se puede usar en varias Plataformas o Sistemas Operativos

Metaprogramación: Es un tipo de programación, donde el programa escrito manipula a otros.

Open Source: Código abierto, así se lo conoce al software que se desarrolla y distribuye libremente

Petición: Es un requerimiento o solicitud que se le hace a un sistema

Tareas: Trabajos que se realizan para conseguir un bien mayor

Tipado Dinámico: Cuando una variable toma diferentes valores en diferentes situaciones.

UNICODE: Estándar para editar caracteres.

Velocidad de Ejecución: Tiempo invertido en un proceso, se expresa en segundos.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Hoy en día en toda empresa grande o pequeña, algún negocio personal o mediano, o cualquier entidad que se maneje información, existe la necesidad de asegurar su información, ya sea para garantizar una buena imagen a la empresa, o debido a que se posea información valiosa. La información que se maneja constantemente puede originar que personas malintencionadas, aprovechen escenarios inseguros y procedan al robo de información. Por este motivo este trabajado está dirigido para instituciones que requieran realizar un análisis de seguridad informática en sus sistemas, de tal forma que sea accesible desde cualquier ordenador, sin necesidad de realizar instalaciones tediosas y garantizando la seguridad de la información proporcionada. Se puede generar un informe detallado del análisis requerido. Se explica los estándares para el correcto funcionamiento de la herramienta.

La herramienta está basada en el análisis que usa la metodología OSSTMM, para la detección de vulnerabilidades en las pruebas de seguridad, mediante el análisis por capas. Cada capa, consta de una serie de análisis, esta división ayuda a una mejor búsqueda del problema, siendo muy eficaz este método. Por lo que se cuantificará los diferentes niveles, es decir, se podrá obtener un grado de seguridad de lo que se esté evaluando. Debido a las limitaciones de tiempo, evalúa solo el nivel de capa de red de la metodología OSSTMM.

Con esta herramienta se busca que los usuarios, mejoren sus sistemas, y repitan el proceso las veces que sean necesarias, para poder corregir sus vulnerabilidades y así minimizar la existencia de problemas informáticos.

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA

En este capítulo se analizó el principal problema de seguridad informática que requiere toda institución, en la cual la protección de información es un asunto muy delicado, y que en ciertos casos, las herramientas existentes necesitan realizar instalaciones específicas para poder hacer uso de la herramienta.

1.1 Descripción del problema

En Ecuador, el ámbito de seguridad informática, se ha convertido en un tema muy importante para las empresas, instituciones públicas o privadas, y todas aquellas entidades desde un simple lugar de trabajo, hasta una gran institución; esto ha generado preocupación en cada una de ellas, ya sea por diferentes motivos, como poseer información de correos electrónicos, contraseñas, número de cuentas bancarias y

mucha información valiosa para la empresa que puede ser fácilmente expuestas si no se cuenta con una seguridad informática adecuada y para los clientes que pueden llegar a verse perjudicados y su posterior desconfianza con la empresa.

Según un artículo emitido por el diario el Universo, menciona que los delitos informáticos en el Ecuador, se han incrementado, y la principal causa de este problema es que el país se encuentra creciendo en su economía, por lo que lo convierte en un blanco fácil de los ataques cibernéticos, además de que estos ataques no se los realizan principalmente dentro del país, sino fuera del mismo de países como Perú, Colombia, o países de Europa. [1]

La Seguridad de la información no solo es de asunto informático, muchas veces es de cuestión social. La mayoría de las empresas en el país no toman en cuenta este factor muy importante, por lo que las personas que atacan al sistema informático de la empresa son muchas veces hasta el personal propio de la empresa, personal despedido o personas que indirectamente proporcionan información.

Otro problema se origina también a nivel de diseño de las redes informáticas, es decir, los aparatos físicos que la conformar como

enrutadores, conmutadores, puntos de accesos, aparatos terminales, y que por una mala configuración pueden estar siendo víctimas de ataques.

Las herramientas actuales, están desarrolladas sobre lenguajes de programación de código abierto, herramientas como Dradis, Magic Tree entre otras, el objetivo de la implementación de esta herramienta, es que sea accesible vía web, es decir, sin previa instalación de paquetes o programas, y brindar el mismo servicio que alguna herramienta de escritorio que realice estas tareas; ya que con esto la herramienta puede ser de mucha utilidad para cualquier tipo de empresa que requiera una análisis sencillo o personalizado.

Todos estos problemas se originan muchas veces por personas, que no protegen su información personal como es debida, es por eso que se realizan campañas o capacitaciones ya sea para el personal de una empresa, como para personas en general, sobre las precauciones de cómo proteger sus datos personales. Sin embargo, este problema no está en los usuarios, sino en el personal encargado de la seguridad en la institución para que laboran, así como irresponsabilidades por parte del personal por dejar claves a disposición de otras personas, fácil

persuasión del personal, mala configuración de los equipos informáticos como son los de redes, aparatos electrónicos y dispositivos terminales, o mala programación de alguna aplicación en particular por parte de desarrolladores.

1.2 Solución del Problema

Las normas de seguridad que cumple la herramienta web, están basadas en OSSTMM (Metodología Abierta de Prueba de Comprobación de Seguridad), que es uno de los estándares más usados en las pruebas de auditoría de seguridad de los sistemas.

Para la elaboración de la herramienta, el lado del cliente se usó lenguaje de marcado HTML, y del lado del servidor el uso de PHP, y la información y recopilación de datos por parte de las pruebas que se realizaron, esta información se aloja en una base de datos MySQL.

En cuanto a la arquitectura de la herramienta que separa los datos de la lógica de negocios, se tomó en cuenta el modelo MVC (Modelo, Vista y Controlador), de esta forma a la herramienta web se podrá adaptar diversos módulos para un análisis detallado en cada nivel de seguridad.

La comparación de la herramienta, se la realizó a través de pruebas con otras herramientas que tienen como fin generar un informe detallado por cada capa, y también se realizaron los respectivos cuadros comparativos.

Las capas usadas por la metodología OSSTMM son:

- Pruebas de seguridad humana PHYSSEC - HUMSEC.
- Pruebas de seguridad física PHYSSEC.
- Pruebas de seguridad inalámbrica SPECSEC.
- Pruebas de seguridad en las telecomunicaciones COMSEC.
- Pruebas de seguridad en las redes de datos COMSEC.

De las capas mencionadas anteriormente, se procederá a realizar el estudio, la cuantificación y la implementación de la capa de redes de datos COMSEC en la herramienta web que se va a desarrollar, por motivos de complejidad en el análisis y en los informes generados.

1.3 Objetivos

Se detalla a continuación los objetivos necesarios para el estudio y la implementación de la herramienta de seguridad informática, de acuerdo al problema previamente planteado.

1.3.1 Objetivo general

Implementar una herramienta web, que seas de utilidad para cualquier tipo de empresa, y que se pueda obtener de ella un informe detallado de la prueba de seguridad informática realizada.

1.3.2 Objetivos específicos

- Analizar y seleccionar la arquitectura y diseño modular adecuado para realizar la implementación de la herramienta.
- Comparar la herramienta desarrollada con otras similares, disponibles en el mercado.
- Generar informes con la herramienta desarrollada.

1.4 Justificación

Hoy en día todas las empresas, instituciones públicas, privadas que manejan grande cantidades de información, sensible para la propia

empresa o para los clientes, es necesario garantizar la seguridad y confidencialidad de la información y así establecer una relación de confianza hacia la empresa. Además, el acceso prolongado a internet, origina otros problemas de seguridad en diferentes aspectos como es la red de la empresa, manipulación del personal para la obtención de información, entre otros factores. Por todos estos motivos, es necesario que se capacite al personal de una empresa, corregir errores de seguridad en la red, entre otras soluciones que den como resultado la seguridad de la información y su posterior confianza con las personas, mediante una herramienta que sea capaz de proveer información sobre que normativas debe seguir y de qué manera; para que cada empresa pueda realizar sus respectivos análisis de seguridad.

Como se ha mencionado anteriormente, existen muchas herramientas que realizan este tipo de análisis de prueba de penetración, pero estas pueden llegar a ser poco amigables con el usuario o de difícil instalación, ya que requieren requisitos especiales del sistema operativo y/o paquetes previamente instalados, entre otros impedimentos, para una rápida prueba de penetración. Por este motivo, se propone una herramienta que sea accesible vía web, sin previos requisitos especiales, y sobre todo adaptable para cualquier tipo de empresa

independientemente de su tamaño, ya sea para pruebas generales o detalladas.

Muchas veces el origen del problema es de difícil detección, sobre todo cuando se refiere a seguridad informática, por este motivo es que se estudia la metodología OSSTMM, ya que sectoriza el análisis por niveles, separando de esta manera la prueba que se esté realizando, y poder encontrar el problema y así poder corregirlo. Lo importante de esta metodología es que no solo se enfoca en los análisis técnicos comunes, sino que abarca términos éticos, responsabilidades, crea un plan temporal de ejecución, etc. [2]

Otro motivo a considerar, es la accesibilidad de los usuarios a una herramienta web, que garantice confidencialidad en su información, y así poder guardar el resultado de sus pruebas de seguridad, en una aplicación garantizada y al alcance de todos. Sin tener que realizar instalaciones tediosas para usuarios comunes o que usaran la herramienta con fines de aprendizaje o de pequeños análisis. Sin embargo también existirán usuarios de grandes empresas, que necesiten de la herramienta sea solo de uso propio, por lo que una herramienta de instalación local sea la adecuada para sus informes.

1.5 Limitaciones del Proyecto

Durante la elaboración del proyecto pueden presentarse una serie de problemas los cuales son considerados para una posible prevención de los mismos, a continuación se detalla cuáles serían estos posibles escenarios, y a que se deberían y bajo qué circunstancias.

Como ya se ha mencionado la metodología OSSTMM consta de cinco niveles, sin embargo para este proyecto se considerará las pruebas de seguridad en las redes de datos, por consiguiente se enfocará el estudio en este nivel. Por otro lado esta herramienta estará diseñada, para que en una posterior modificación de la misma se puedan agregar el resto de niveles de seguridad sin problema alguno.

El enfoque que se da a la herramienta es muy variado, dependiendo del usuario que lo esté usando, por este motivo ya sea una empresa grande o un simple usuario, cualquiera de ellos van a tener distintos análisis, o van a requerir informe más o menos detallados que los otros.

La metodología OSSTMM menciona una serie de informes que pueden ser generados al final de una auditoria como son los siguientes:

- Plantilla de Perfil de Red
- Plantilla de Datos del Servidor
- Plantilla de Análisis de Cortafuegos
- Plantilla de Testeo Avanzado del Cortafuegos
- Plantilla de Testeo de Sistemas de Detección de Intrusiones (IDS)
- Plantilla de Ingeniería Social sobre el Objetivo
- Plantilla de Ataque Telefónico usando Técnicas de Ingeniería Social
- Plantilla de Ataque por Correo Electrónico usando Técnicas de Ingeniería Social
- Plantilla de Análisis de Confianza
- Plantilla de Revisión de Políticas de Privacidad
- Plantilla de Revisión de Medidas de Contención
- Plantilla de Correo Electrónico falseado
- Plantilla de Información Competitiva
- Plantilla de Ataques a Contraseñas
- Plantilla de Denegación de Servicio
- Plantilla de Análisis de Documentos
- Plantilla de Ingeniería Social

Todos estos informes no serán usados de la misma forma para una gran compañía, como de un usuario en particular, por lo que se realizaran informes resumidos de los mismos.

La herramienta web, se la realizará en un lenguaje de programación en particular, sin embargo el uso de herramientas de terceros tanto para el desarrollo de la herramienta de prueba de seguridad, así como de herramientas que ayuden a evaluar los diferentes niveles de seguridad, pueden ir actualizándose, por lo que puede llegar el momento el cual, no se pueda hacer referencia a resultados con esa herramienta en particular, ya que puede a llegar a cambiar totalmente su lógica para entregar resultados, o datos que requieran anteriormente a otra versión de la misma.

1.6 Resultados esperados

Con este proyecto se espera implementar una herramienta web, que sea capaz de generar un informe detallado de cada capa OSSTMM. De igual manera, se podrá proponer una alternativa de pruebas de penetración para la seguridad informática, que sea accesible de manera amigable sin necesidad de instalaciones adicionales, preferiblemente desde un navegador web. Finalmente, nos permitirá determinar los

aspectos en los que la herramienta desarrollada supera soluciones existentes, según las pruebas comparativas que se realicen en el transcurso del estudio.

Además de simplificar el nivel de seguridad de redes de datos de la metodología OSSTMM y a la vez poderla cuantificar, para realizar calificaciones a base de puntuaciones para cada nivel y generar el reporte final por medio de rangos de las calificaciones anteriormente obtenidas, con esto el usuario, podrá corregir y reevaluar su auditoria.

CAPÍTULO 2

2. MARCO TEÓRICO

En este capítulo se detalla el estudio de los distintos componentes con las cuales se va a realizar la herramienta web, así como el estudio de estándares de seguridad informática, particularmente la metodología OSSTMM y en que consiste cada una de sus capas, además del estudio de distintos modelos de arquitectura para las aplicaciones web. Finalmente se analiza los distintos lenguajes con sus respectivos frameworks realizando un estudio de las ventajas y beneficios que pueden proporcionar para la adecuada implementación de la herramienta web.

2.1 Estándares de Auditoría para Prueba de Seguridad Informática

Las auditorías de seguridad informática comprenden el análisis y administración efectuado sobre sistemas, estaciones de trabajos, empresas, redes, servidores, entre otros. El objetivo de estas auditorías

es de identificar, documentar y corregir, fallos de seguridad sobre los sistemas.

Las auditorías realizan una serie de procesos, que de acuerdo al estándar que se esté usando, puede variar, sin embargo de forma general, se realizan inspecciones y se detallan informes sobre los resultados generados por el análisis. Además de verificar el cumplimiento de estándares internacionales, detectar, comprobar y evaluar vulnerabilidades que existan, y proponer medidas preventivas para los sistemas en análisis. [1]

Existen una serie de auditorías de seguridad que pueden ser usados según sea el caso que se esté evaluando, como es auditoria de seguridad interna, seguridad externa, test de penetración, análisis forense, auditoria de páginas web, de código de aplicaciones. Todas estas auditorías deben ser realizadas periódicamente para proteger la integridad de las medidas preventivas de anteriores auditorias, como es actualización de software, cambio en las configuraciones. [1]

Los estándares son una serie de normas de buenas prácticas sugeridas para la realización de un análisis en particular, en este caso para la

detección de vulnerabilidades informáticas. Existen una gran cantidad de estándares que se usan, sin embargo, muy pocos son orientados a todos los niveles de un mismo sistema, es decir, que analice varias capas, como son la humanística, física, enlace de datos, aplicación, inalámbricas, la metodología OSSTMM es una de las más completas. Por otro lado existen otros estándares que realizan estudios en capas particulares como es COBIT, que garantiza la seguridad en los sistemas, el estándar ISO 27002 el cual conforma de un conjunto de buenas prácticas de seguridad informática. [2]

El estándar COBIT, es un excelente guía enfocado a la administración y supervisión de tecnologías de Información, su ideología es de investigar, desarrollar, publicar y promocionar una gran cantidad de objetivos para las TI. Entre los beneficios que proporciona tenemos que, optimiza los servicios de las TI, apoya el cumplimiento de leyes, políticas y reglamentos, y gestiona nuevas TI. [3]

El estándar ISO 27002 proporciona recomendaciones para la gestión de buenas prácticas de seguridad informática, preserva la seguridad de la información, la integridad y la disponibilidad en los sistemas. La última revisión vigente del estándar es ISO/IEC 27002:2013. Anteriormente se

mencionaba que los estándares cumplen un conjunto de normas que pueden ser analizados, este estándar en particular organiza la seguridad de la información, gestiona archivos, controla accesos, maneja la criptografía, seguridad física y ambiental, seguridad de operaciones, de comunicaciones entre otras normas. [4]

2.2 Pruebas de Seguridad en los sistemas de Información (OSSTMM)

OSSTMM es una metodología que involucra pruebas operacionales en campos físicos, interacción humana y en todas las formas de comunicación como inalámbrica, cableada, analógica y digital. Este reúne casos de vulnerabilidad en los sistemas de información de todo tipo y provee soluciones.

Para facilitar la aplicación de esta metodología existe un manual OSSTMM v3.0, la cual es la última versión, La metodología OSSTMM trata los aspectos más importantes de los sistemas de información:

- Seguridad de la Información
- Seguridad de los procesos
- Seguridad en las tecnologías de Internet
- Seguridad en las comunicaciones
- Seguridad inalámbrica

- Seguridad física

Por lo que podemos decir que es bastante completa en relación a los modelos convencionales de protocolos de red, el más general es el TCP/IP que se subdivide en: Capa de aplicación, puede ser abarcada por la seguridad de la Información y la seguridad en las tecnologías de Internet, dependiendo del medio podrá incluir también seguridad inalámbrica; Capa de transporte, entra en la seguridad de los procesos y seguridad en las comunicaciones; Capa de red, tiene que ver con la seguridad en las tecnologías de Internet; Capa de acceso a la red se centra en la seguridad física.

La metodología OSSTMM es mantenida por el ISECOM, desarrollado por una comunidad abierta y con el objetivo de crear conciencia en la revisión o en el testeo de los campos de información más importantes. Por lo que puede decirse que es una metodología que contiene información acerca de testeo de vulnerabilidades en todos los campos y desarrollado por personas de todo el mundo que han disipado estos posibles ataques, o los han sufrido y luego han aprendido como evitarlos. [5]

2.3 Arquitecturas Seguras

Para el desarrollo del software y aplicaciones de toda clase es necesario un diseño adecuado tanto de su interfaz como en su funcionamiento para realizar tareas en específico. Las arquitecturas de software ayudan a que la programación sea mejor legible para los programadores, que en un futuro puedan comprender las líneas de código sin mayor esfuerzo, ya que esto permite tener organizado todo el código fuente. Existen una gran variedad de arquitecturas que han sido establecidas como modelos para el origen de otras secundarias o derivadas de estas, estas son Descomposición Modular, Cliente-Servidor y de Tres Niveles. Entre las arquitecturas derivadas más usadas están, el Modelo Vista Controlador, Pipeline, Orientada a Servicios, entre otras. Todas estas arquitecturas pueden sr implementadas según las necesidades del software. [6]

Además se pueden usar arquitecturas diseñadas por los mismo programadores, sin embargo todas las arquitecturas mencionadas anteriormente han sido estudiadas en todos sus aspectos, como es funcionalidad, reparto de trabajo requerido por el software, seguridad en sus distintas capas, es por este motivo que para el diseño de cualquier software o aplicaciones web como es el caso de este proyecto, que se

usen cualquiera de estas ya existentes, evitándose un estudio más profundizado en el diseño del mismo.

2.3.1 Descomposición Modular.

Esta Arquitectura tienes la finalidad de dividir el sistema en módulos, definiendo sus interfaces, se debe identificar los módulos que van a intervenir en el sistema, luego se procede a describir cada módulo, finalmente se procede a establecer relaciones entre ellos tal como se muestra en la figura 2.1.

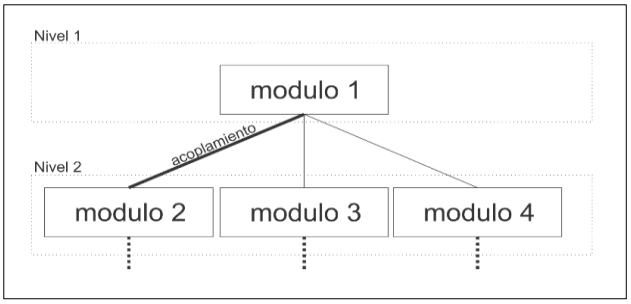


Figura 2.1 Arquitectura Descomposición Modular

Todos estos módulos deben tener ciertas cualidades las cuales se pueden mencionar como ser independientes funcionalmente, acoplamiento entre los módulos del sistema, cohesión y adaptabilidad. [7]

2.3.2 Cliente – Servidor

Este modelo de arquitectura se basa en repartir las tareas que realiza el sistema en dos partes la del servidor y el cliente. Es decir un cliente realiza una petición, la cual es procesada por el servidor, y le da una respuesta, uno de los ejemplos se podría mencionar los programas que son ejecutados en el mismo computador. Esta separación es de tipo lógica ya que no necesariamente cada parte de este modelo está en computadoras distintas

El cliente es el que envía y recibe la petición del servidor, además de poder establecer comunicación con varios servidores, esta parte del modelo es la que se encuentra interactuando directamente con el usuario final, Por otro lado el servidor recibe y procesa las peticiones provenientes del cliente para su posterior respuesta, en gran parte de los servidores pueden recibir varias peticiones a la vez de muchos clientes, sin embargo tiene un limitante el cual es la cantidad de recursos con la que cuenta el servidor para poder procesar las tareas. [8]

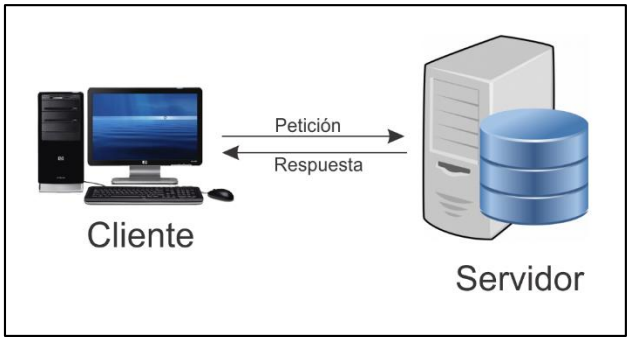


Figura 2.2 Modelo Cliente-Servidor

2.3.3 Arquitectura de Tres Niveles

La arquitectura de Tres Niveles tiene que ver con el modelo Cliente-Servidor, sin embargo se ha establecido una capa intermedia que es la de Negocios la cual actúa de intermediario entre las dos capas, es decir en la capa de negocios se encarga de realizar las operaciones o transacciones requeridas del cliente hacia el servidor, repartiendo más aun las tareas que tienen ambas partes, haciendo que la arquitectura sea más flexible y de fácil detección de fallos.

Es una de las arquitecturas más usadas para aplicaciones web, como es el caso requerido en el proyecto, ya organiza de una manera más eficaz el desarrollo de la misma. El Modelo-Vista-

Controlador es un patrón de arquitectura de tres niveles, y lo usan una gran variedad de frameworks para el desarrollo de aplicaciones web.

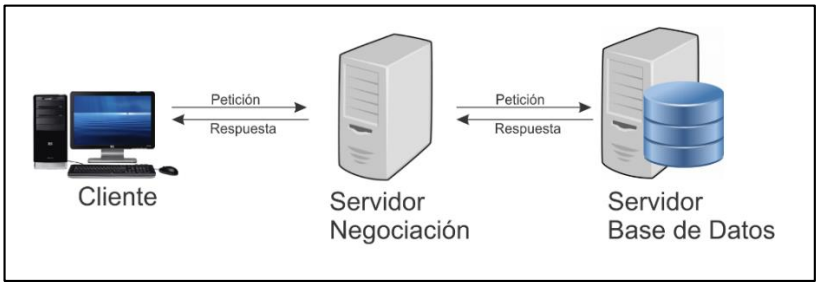


Figura 2.3 Modelo de 3 Capas

2.4 Frameworks Seguros

Los Frameworks son esquemas desarrollados en un lenguaje de programación en particular, que ayudan a la programación del desarrollador, reduciendo tiempo y esfuerzo al momento de programar. Por otro lado el uso de Frameworks en la actualidad ha servido para mejorar prácticas de buena programación, como es la reutilización de código que es importante al momento de realizar un proyecto en un corto periodo de tiempo. Así como también ha mejorado la colaboración entre distintos equipos de trabajo. Además el trabajo colaborativo de herramientas desarrolladas por otras personas, pueden ser fácilmente adaptadas a los nuevos proyectos.

Existe una gran variedad de lenguajes de programación, que son lenguajes formales que fueron diseñados para interpretar líneas de código y realizar procesos con algún fin en específico. Los lenguajes de programación están formados por conjunto de caracteres o símbolos que definen su estructura.

Un informe de Veracode concluye que algunos lenguajes de programación son más vulnerables que otros, en ataques como son de inyección SQL, pero en el más común de los casos, los desarrolladores son los que tienden a cometer más errores de programación.

Lenguajes de programación como C/C++ que son primitivos no son seguros, esto más se refiere, por el descuido que se comete al programar en estos lenguajes, ya que por ejemplo al ingresar alguna variable, esta no es controlada, como qué clase de caracteres son ingresados, es decir, puede fácilmente olvidarse, si no se toma medidas respectivas, lo que generaría una posible inyección SQL en la aplicación que se esté trabajando.

Existe una serie de Frameworks por cada lenguaje de programación, y depende también del uso que se le dé, o la aplicación que se vaya a realizar, para poder seleccionar un framework adecuado.

En la figura 2.4 se menciona algunos Frameworks en base al lenguaje de programación en la que son escritos.

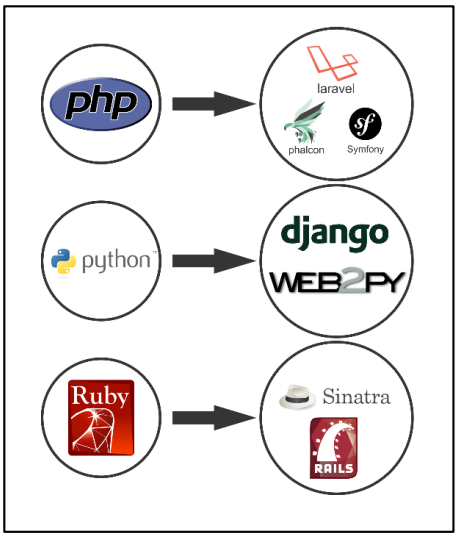


Figura 2.4 Lenguajes de Programación y Frameworks

Para poder analizar esta sección se ha realizado un análisis con los lenguajes de programación más usados en la actualidad, así como los que tienen más contribuciones y comunidades existentes. A continuación se detallan estos lenguajes de Programación con sus respectivos Frameworks.

2.4.1 PHP

PHP es un lenguaje de programación de uso general, es decir, puede ser usado con múltiples propósitos como comunicación entre computadoras, acceso a base de datos, comunicación entre dispositivos, diseño de páginas, diseño de imágenes, captura de datos. Es un lenguaje multiplataforma diseñado para el desarrollo de páginas web dinámicas, es el lenguaje más popular y extendido en la web. [9]

Actualmente se encuentra en vigencia la versión 5.6 de PHP, y se está trabajando para el lanzamiento de la versión 6, este lanzamiento ha sido retrasado porque los desarrolladores decidieron darle un enfoque distinto a las cadenas UNICODE, y están considerando soluciones alternas para esta próxima versión.

Tabla 1 Versiones de PHP [10]

Versión	Fecha de lanzamiento	Fin de soporte
1.0	08-06-1995	-
2.0	01-11-1997	-
3.0	06-06-1998	20-10-2000
4.0	22-05-2000	23-01-2001

Versión	Fecha de lanzamiento	Fin de soporte
4.1	10-12-2001	12-03-2002
4.2	22-04-2002	06-09-2002
4.3	27-12-2002	31-03-2005
4.4	11-07-2005	07-08-2008
5.0	13-07-2004	05-09-2005
5.1	24-11-2005	24-08-2006
5.2	02-11-2006	06-01-2011
5.3	30-06-2009	14-08-201418
5.4	01-03-2012	No especificada
5.5	20-06-2013	No especificada
5.6	20-08-2014	No especificada
6.0	Sin fecha	No especificada

El framework Laravel es de código abierto que fue desarrollado para aplicaciones y servicios web, está programado en php, este framework, está diseñado para el uso de programación elegante y expresiva, lo que nos quiere decir que es de fácil comprensión, que si el día de hoy realizamos un programa con este framework, años más tarde podremos recordar su lógica de una manera muy fácil.

Entre las características que incorpora están: sistema de ruteo, motor de plantillas Blade, petición Fluent, Eloquent ORM, basado en Composer, soporte de Caché, soporte MVC, usa componentes de Symfony, este último punto es muy importante, ya que Laravel tiene ventajas sobre otros frameworks, porque usa la arquitectura MVC, que se ha perdido en otros framework como Symfony, que en un inicio, fueron diseñados para facilitar esta arquitectura. [11]

El framework Phalcon es de alto rendimiento, desarrollado en PHP, es de código abierto con licencia BSD, a diferencia del resto de frameworks se implementan con alguna extensión en C/C++, lo que hace que se obtenga excelente rendimiento al momento de desarrollar aplicaciones web, con esto se evita tener grandes cantidades de códigos que quizá no se usen en la aplicación, y que aumente la velocidad de ejecución, lo que permite procesar muchas más solicitudes por segundo que otros frameworks escritos en php. [12]

Como consecuencia Phalcon es un framework orientado a la optimización de las aplicaciones web, es decir, aquellas

herramientas desarrolladas, en las cuales se tengan grandes cantidades de peticiones que realizar. [13]

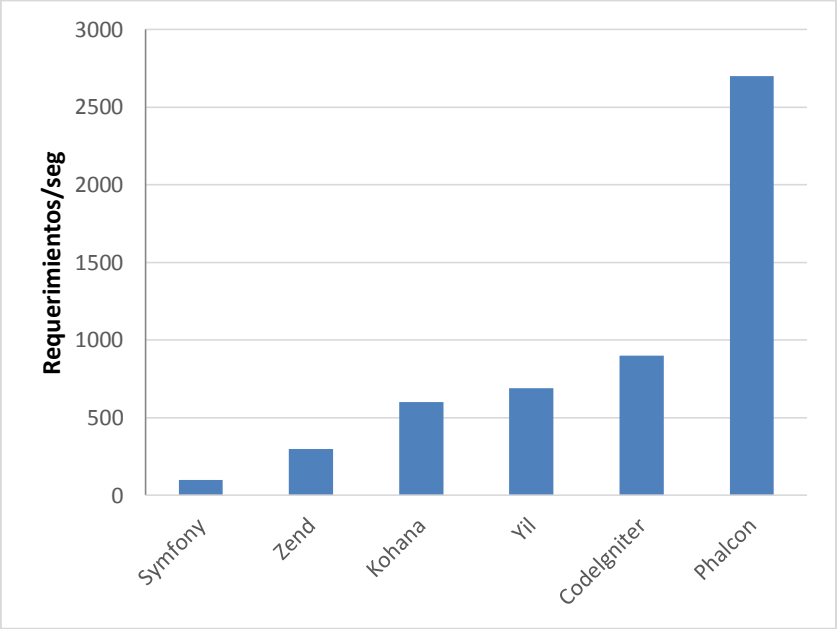


Figura 2.5 Framework vs Requerimientos por Segundos

Phalcon es un framework muy importante, puesto que maneja la arquitectura MVC, además cuenta con un motor de plantillas muy conveniente, como es Volt, posee también su propio ORM para persistir datos en la base, lo cual resulta de una manera muy fácil migrar desde PostgreSQL a MySQL o SQLite sin alterar una línea de código.

El framework Symfony fue diseñado para optimizar el desarrollo de aplicaciones web, bajo la licencia de Open Source, este framework, se desarrolló bajo el concepto de buenas prácticas, la estandarización y la interoperabilidad de las aplicaciones en su origen como lo menciona en el sitio oficial de Symfony, lo que permite la creación de módulos para la realización de varios trabajos determinados.

Según una entrevista a Fabien Potencier, creador de Symfony, el menciona que este framework, no es creado siguiendo la arquitectura MVC, sino más bien a la simple respuesta de las peticiones que se realicen, es decir, un controlador, cuyo trabajo es comunicarse con servicios, que efectúan específicas funciones para cada petición. [14]

2.4.2 Python

Es un lenguaje de programación interpretado, multiparadigma, y multiplataforma, y posee una licencia de código abierto. Fue creado con la necesidad de fortalecer el concepto de código legible al momento de desarrollar aplicaciones. [15]

Alguna de las características que posee Python es que usa tipado dinámico y conteo de referencias para una adecuada administración de la memoria usada cuando una aplicación está en producción. Python es un lenguaje que muchos usuarios de Linux prefieren ya que tiene una filosofía muy parecida. [16]

La última versión de Python vigente es la 3.x, y esta última actualización incluye una cantidad considerable de cambios en el código fuente de versiones anteriores. [17]

El framework Django es de código abierto, con la finalidad de facilitar el desarrollo web, escrito en Python, que se basa en el paradigma conocido como Modelo Plantilla Vista. El objetivo fundamental de este framework es de facilitar la creación de sitios web muy complejos, es decir, que reutiliza código necesario por el programador, y evitar grandes cantidades de líneas de código, desarrollo rápido, y el principio DRY.

La base de datos recomendada es PostgreSQL, sin embargo también soporta MySQL, y SQLite 3. Django incluye un servidor web liviano, destinado para realizar pruebas en la etapa de

desarrollo, sin embargo para la etapa de producción se recomienda el uso del servidor Apache, con el modulo para Python activo. [18]

Aunque Django fue desarrollado inspirado en la arquitectura MVC, los desarrolladores del mismo mencionan que no precisamente siguen este modelo, ya que implementan otros dos partes en esta arquitectura, Estas partes del modelo de arquitectura son Presentación, Control, Mediator, Entity y Foundation. [19]

Otro framework desarrollado en Python es Web2py, es de código abierto, su principal uso, es de dar soporte en el desarrollo de aplicaciones web, que puedan crecer con el tiempo, que sean muy seguras y portables. Este framework fue inspirado en Ruby on Rails, lo que nos quiere decir, que es de desarrollo rápido, además de seguir la arquitectura MVC. [20]

Web2py incorpora componentes para una serie de necesidades como: mecanismos múltiples de autenticación y control de acceso basado en roles, capa de abstracción de base de datos,

API para CRUD, maneja múltiples protocolos, y maneja pedidos y respuestas HTTP, cookies y sesiones. Además como anteriormente es un framework seguro por lo que se alienta a prácticas reconocidos como el manejo seguro de archivos, auto-envío de formularios, y sesiones en el servidor. [21]

2.4.3 Ruby

Es un lenguaje programación multiparadigma, posee una licencia de software libre, es multiplataforma. El creador de Ruby, Yukihiro Matsumoto, menciona que este lenguaje está diseñado para la productividad y la diversión del desarrollador, lo que quiere decir que se enfatiza más las necesidades humanas que las de la máquina. [22]

Entre las características a destacar de este lenguaje de programación es que es orientado a objetos, recolección de basura automática, hilos de ejecución usando Green threads, introspección, reflexión y metaprogramación. La última versión estable fue Ruby 2.1.2

Rails es un framework aplicaciones web de código abierto, el cual está escrito en Ruby, usa la arquitectura MVC, además de proporcionar la facilidad a los desarrolladores de escribir menos código que otros frameworks. Rails se distribuye por medio de RubyGems, que es el medio principal de distribución de paquetes, bibliotecas y aplicaciones.

Para realizar pruebas y el desarrollo, es recomendable el uso de Mongrel o WEBBrick, y para la producción se desarrolló mod_rails para Apache, otras opciones son Nginx, Mongrel, Lighttpd. Con respecto al soporte de base de datos, Rails maneja SQLite pero se recomienda el uso de SGBDR; para Rails la iteración con la base de datos es totalmente abstracta, aunque se puede realizar consultas en la misma. [23]

Sinatra es un framework escrito en Ruby, desarrollado para realizar aplicaciones web, y es de código abierto, Sinatra depende una interfaz de servidor web que es Rack. Este framework usa una arquitectura diferente a la de MVC, lo cual origina la filosofía de Sinatra que es “rápida creación de aplicaciones web en Ruby con el mínimo esfuerzo”. [24]

Algunas compañías que usan Sinatra se pueden mencionar a: BBC, Engine Yard, Heroku, GitHub, y SongBird. Como se mencionó anteriormente Sinatra usa la filosofía del mínimo esfuerzo, lo que llevo a sus desarrolladores a tener menos líneas de código sobre el resto de frameworks, en una comparación ron Rails, podría ser hasta 50 veces más liviano. Sin embargo este framework está orientado al desarrollo de aplicaciones pequeñas, por ser muy rápido y flexible. [25]

Se realizó un estudio de la eficiencia entre los dos frameworks Ruby on Rails, Sinatra, Padrino y Goliat por el equipo de desarrollo de Ruby on Rails, Altoros de Argentina, en una de sus publicaciones se hicieron pruebas sobre la infraestructura; Leandro Fridman miembro del grupo, menciona “ApacheBench 2.4.3 ab -n 1000 -c 100 localhost , un 2,3 GHz Intel Core i5 Sandy Bridge CPU , 4GB de DDR3 1333MHz RAM, 120 GB SSD Intel 330 , OS X Mavericks 10.9.1 , Ruby 2.0.0p247 y MySQL 5.6.14”. El servidor Web usado fue Thin 1.6.1. la prueba de estrés se realizó bajo una prueba de estrés de 1000 peticiones, los cuales originaron resultados para distintas aplicaciones web, como para aplicaciones pequeñas, el framework de Sinatra resultó ser el

más rápido, y para aplicaciones más elaboradas, el framework Ruby on Rails resultó ser el adecuado. [26]

Tabla 2 Rendimiento de Ruby On Rails y Sinatra

Framework	No Views and DB	Views (Slim)	Views (Slim) + MySQL (Sequel)
Sinatra 1.4.4	0.626 ms	5.730 ms	7.159 ms
Rails 4.0.2	1.539 s	1.790 s	2.501 s

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA WEB

En esta sección se procedió a realizar el respectivo estudio para poder desarrollar la herramienta web, así como analizar la metodología OSSTMM y poder obtener un modelo resumido, siguiendo las plantillas de reporte que se usan. Además de proponer la arquitectura que tendrá la herramienta web, así como su diseño modular, el cual es adecuado para poder crear módulos en un futuro.

3.1 Requerimientos

Para la elaboración de la herramienta web se realizó varios estudios previamente realizados en el capítulo 2, en el cual menciona de manera general las diferentes metodologías existentes, así como las diferentes

herramientas existentes, y los distintos lenguajes de programación en los cuales se pudieron desarrollar, por consiguiente y en base a los estudios previamente realizados, se procedió a realizarlo en lenguaje PHP, usando el framework de Symfony para un mejor desarrollo; para la elaboración de la metodología a usar, fue necesario realizar una versión resumida, para un mejor análisis para los usuarios. A continuación se presenta el estudio correspondiente a cada parte ya mencionada.

3.1.1 Metodología OSSTMM

Esta metodología al igual que el resto son un conjunto de reglas y normas para controlar cuando, en qué momento y cómo son realizados los eventos de pruebas de penetración, sin embargo es una metodología que solo realiza un estudio desde un entorno externo, además de presentar diferentes análisis para las diferentes capas que se postulan en esta metodología como es la seguridad Física, seguridad inalámbrica, seguridad de comunicaciones, seguridad de la información, seguridad de las tecnologías de internet y seguridad de procesos. Además un documento elaborado por la ISECOM menciona que para que una prueba de seguridad sea considerada dentro del estándar OSSTMM debe de considerarse algunos puntos como: “ser

cuantificable, consistente y que se pueda repetir, válido más allá del tiempo actual, basado en méritos del consultor y analista pero no en marcas comerciales, exhaustivo, y concordante con leyes individuales y locales y derecho humano a la privacidad.” [26]

Con el desarrollo de esta herramienta, lo que se logró es realizar un análisis basado en la metodología OSSTMM, sin embargo al ser cambiada en ciertos procesos para realizar pruebas de una manera más rápida, se pierde el valor como tal, pero lo que se logra es que la herramienta sea aún más intuitiva para los usuarios que requieren alguna revisión rápida de sus negocios.

Como ya se mencionó anteriormente la metodología OSSTMM realiza estudios en diferentes secciones, sin embargo solo se realizará el estudio respectivo sobre la sección de seguridad de las tecnologías de internet. Además se realizó la herramienta por módulos para que posteriormente se pueda agregar el resto de secciones que conforman la metodología, estos módulos están detallados en la sección de Diseño modular de este capítulo.

La sección de Seguridad de Tecnologías de Internet comprende un gran número de módulos para realizar, con sus respectivas plantillas, y por motivo de que solo se tomará en cuenta a nivel de aplicaciones y redes, se omitirán ciertos módulos los cuales están indicados en la Tabla 3 donde se indica los procesos considerados en la herramienta.

Tabla 3 Procesos considerados en la Sección de Seguridad [27]

Módulos	Considerado
Logística y Controles	Si
Sondeo de Red	Si
Identificación de Servicios de Sistemas	Si
Búsqueda de Información Competitiva	No
Revisión de Privacidad	No
Obtención de Documentos	No
Búsqueda y Verificación de Vulnerabilidades	Si
Testeo de Aplicaciones de Internet	No
Enrutamiento	Si
Testeo de Sistemas Confiados	No
Testeo de Control de Acceso	No

Módulos	Considerado
Testeo de Sistema de Detección de Intrusos	No
Testeo de Medidas de Contingencia	No
Descifrado de Contraseñas	Si
Testeo de Denegación de Servicios	Si
Evaluación de Políticas de Seguridad	No

En el módulo de Logística y Controles el objetivo principal es de disminuir los falsos positivos y negativos, es decir lo que se espera en este módulo son irregularidades en el ancho de banda, paquetes TCP, UDP, ICMP perdidos, problemas en el enrutamiento y tráfico de enrutamiento del ISP y vendedores de tráfico, de los cuales se considerará solo los paquetes perdidos y el ancho de banda y sus irregularidades como se muestra en la Tabla 4.

Tabla 4 Módulo de Logística y Controles [27]

Resultados Esperados	Considerado
Irregularidades de Ancho de banda	SI
Paquetes TCP perdidos	SI

Resultados Esperados	Considerado
Paquetes UDP perdidos	SI
Paquetes ICMP perdidos	SI
Problemas de enrutamiento	NO
Tráfico de Enrutamiento del ISP	NO

El módulo de sondeo de red, tienes 2 partes fundamentales en la realización de un hacking ético, el cual es el reconocimiento y el control del mismo, es decir que es aquí donde se obtiene información de los diferentes host que pudiesen existir, por ejemplo un excelente lugar donde se puede comenzar a recolectar información son en los registros regionales de internet, ya que proporcionan información sobre los dominios en internet.; esto dependerá de la localización del host, es decir si es un host de América Anglosajona es ARIN, existen distintos registros regionales de internet. Por consiguiente en este módulo se espera identificar los nombres de dominios, nombres de servidores, Direcciones ip, mapa de red, Información ISP/ASP, propietarios de los servicios y de los sistemas. Para el desarrollo de la herramienta se considerará los puntos mostrados en la Tabla 5.

Tabla 5 Módulo de Sondeo de Red [27]

Resultados Esperados	Considerado
Nombres de Dominios	SI
Nombres de Servidores	SI
Direcciones IP	SI
Mapa de Red	NO
Información ISP/ASP	NO
Propietarios del Sistema y del Servicio	NO

El módulo de Identificación de los Servicios de Sistemas, se encarga de escanear y averiguar aquellos puertos de los host encontrados, que estén abiertos o cerrados, así como también se deben indicar que servicios están usando los puertos descubiertos, como se realiza una verificación para los puertos y estos son 65.536 puertos TCP y UDP posibles, no es necesario comprobar en todos los puertos, esta consideración es de libre elección de los auditores, que tendrán en cuenta información acerca de que posibles servicios se estén realizando para cada host. En la Tabla 6 se mencionan los posibles resultados esperados en este análisis sin embargo solo se considerará los más relevantes.

Tabla 6 Módulo de Identificación de Servicios de Sistemas. [27]

Resultados Esperados	Considerado
Puertos Abiertos, Cerrados y Filtrados	SI
Direcciones ip de los sistemas activos	SI
Direccionamiento de los sistemas de la red interna	NO
Lista de los protocolos descubiertos de tunelizado y encapsulado	SI
Lista de los protocolos descubiertos de enrutado soportados	NO
Servicios activos	SI
Tipos de Servicios	SI
Tipo y nivel de parcheado de las Aplicaciones de los Servicios	SI
Tipo de Sistema Operativo	SI
Nivel de parcheado	NO
Tipo de Sistema	SI
Lista de sistemas activos	NO
Mapa de la red	NO

El módulo de Búsqueda y Verificación de Vulnerabilidades, tiene como objetivo es identificar, comprender y comprobar errores en las configuraciones en un servidor o en la red, se deben

considerar no solamente sitios web, si no también uso del IRC, noticias, sitios FTP escondidos, es necesario que los auditores identificar aquellos scripts o exploits que pudiesen ser usados, ya que como son códigos accesibles para cualquier persona, pueda que existan otras formas para generar vulnerabilidades. En la tabla 7 se estipulan los resultados esperados por la metodología, además se menciona cuales serán considerados.

Tabla 7 Módulo de Búsqueda y Verificación de Vulnerabilidades. [27]

Resultados Esperados	Considerado
Tipo de aplicación o servicio por vulnerabilidad	SI
Niveles de parches de los sistemas y aplicaciones	SI
Listado de posibles vulnerabilidades de denegación de servicio	SI
Listado de áreas seguras a través de ocultación o acceso visible	NO
Listado de vulnerabilidades actuales eliminando falsos positivos	NO
Listado de sistemas internos o en la DMZ	NO
Listado de convenciones para direcciones de e-mail, nombres de servidores, etc.	SI
Mapa de red	NO

El módulo de Enrutamiento, tiene como finalidad analizar la seguridad a nivel de enrutadores, los cuales son los encargados de las políticas de seguridad y los cuales usan ACLs, que aceptan o rechazan los paquetes, es decir el auditor debe cerciorarse de que los paquetes que son permitidos lo sean, y el resto sea rechazado, en la Tabla 8 se detalla los resultados esperados en este análisis y también se indica cuales se considerarían.

Tabla 8 Módulo de Enrutamiento. [27]

Resultados Esperados	Considerado
Tipo de Router y Propiedades implementadas	SI
Información del router como servicio y como sistema	SI
Perfil de la política de seguridad de una red a partir de la ACL	SI
Lista de los tipos de paquetes que deben entrar en la red	SI
Mapa de las respuestas del router a varios tipos de tráfico	NO
Lista de los sistemas vivos encontrados	NO

El módulo Descifrado de Contraseñas, es el proceso de verificar de que tan robustas se encuentras las contraseñas, mediante herramientas que resuelven contraseñas usando algoritmos de descifrado débiles o muy comunes, los auditores pueden evadir fácilmente ciertas credenciales, ya que los usuarios están mal acostumbrados escribir contraseñas débiles o guardarlas en notas físicas, que son averiguadas mediante ingeniería social, que es un tema que puede ser tratado en otro apartado. De los resultados esperados de este módulo, solo son considerados los indicados en la Tabla 9.

Tabla 9 Módulo de Descifrado de Contraseñas. [27]

Resultados Esperados	Considerado
Ficheros de Contraseñas descifrados o no descifrados	SI
Lista de cuentas, con usuario o contraseña de sistema	SI
Lista de sistemas vulnerables a ataques de descifrado de contraseñas	NO
Lista de archivos o documentos vulnerables a ataques de descifrado de contraseñas	NO
Mapa de las respuestas del router a varios tipos de tráfico	NO

Resultados Esperados	Considerado
Lista de sistemas con usuario o cuenta de sistema que usan las mismas contraseñas	SI

El módulo de Denegación de Servicios, tiene como objetivo probar a la red, que sea eficiente a un ataque DoS, ya que en circunstancias normales las redes, no son diseñadas para soportar mucha carga, alcance o parámetros que abusen de ellos. El ataque de DoS casi siempre puede ser perjudicial, ya que afecta a toda la red, sobre todo a los routers. En la Tabla 10 se muestran los resultados esperados que se han considerado para este módulo.

Tabla 10 Módulo de Denegación de Servicios. [27]

Resultados Esperados	Considerado
Lista de puntos débiles en presencia de Internet incluidos los puntos individuales por averías	SI
Establecer un punto de referencia para un uso normal	NO
Lista de comportamientos de sistema por un uso excesivo	NO
Lista de sistemas vulnerables a DoS	SI

3.1.2 Lenguaje de Programación

En el capítulo 2, en la sección de Frameworks Seguros se detalló cada uno de los posibles lenguajes de programación en la cual la herramienta se diseñaría, así como las ventajas y desventajas por parte de cada lenguaje. Sin embargo los requisitos para nuestra aplicación sea confiable y segura, ya que se estará almacenando registros de pruebas de seguridades, es necesario mencionar algunas cualidades que tienen los lenguajes de programación antes mencionados como Python, Ruby y PHP, en un análisis entre ellos.

UDEMY es un mercado para el aprendizaje en línea, donde se promueve la enseñanza de lenguajes de programación, sin embargo esta compañía realizó un estudio, en base a tres lenguajes de programación muy populares los cuales son PHP, Python y Ruby. En el Anexo A se puede observar los diferentes criterios que se usaron para la evaluación de estos lenguajes, entre ellos se menciona, el nivel de aprendizaje, el nivel de usabilidad, el más popular, los lenguajes más discutidos en foros, el que más ha sido implementado en sitios, donde la comunidad de desarrolladores es mayor, que tan rápidos son, y que tantas líneas de código se deben escribir para una misma tarea. En

base a todos estos análisis podemos considerar, los puntos del soporte por parte de la comunidad, la velocidad de ejecución, y que tantas líneas se codifican, Con estos parámetros se puede concluir que PHP es un lenguaje adecuado para la elaboración de la herramienta. En la tabla 11 se muestran las características consideradas con el lenguaje más destacado, la calificación que se menciona para cada lenguaje de programación en las distintas características que se están evaluando, están ponderadas en base a la investigación realizada por UDEMY, donde la calificación de 10, es para el mejor lenguaje destacado sobre el resto.

Tabla 11 Características de PHP, Python y Ruby [27]

	PHP	RUBY	PYTHON
Soporte de la Comunidad	10	1	1
Líneas de Código	6	10	6
Velocidad de Ejecución	10	6	5
Seguridad	8	10	10

3.1.3 Interfaz Gráfica

Para el diseño de la parte gráfica de la herramienta web, se consideraron varios aspectos, como es la iteración con el

usuario, además de como pudiese ser más intuitiva la herramienta. Para poder realizar ajuste en las respectivas pantallas, se procedió a realizar el modelo de 3 plantillas, los cuales se diferenciaban por los colores, o por la posición de los paneles, en el Anexo B se puede revisar el diseño de las tres plantillas propuestas, por lo que se procedió a realizar una encuesta de diseño, el cual fue dirigido para personas que laboren en ambientes de redes de datos, o analistas de la información.

Las encuestas origino resultados en los cuales se tuvieron que corregir en el diseño de la misma, entre las características establecidas en las encuestas se puede revisar en la Tabla 12, la tendencia de ciertas preferencias de cada plantilla propuesta.

Tabla 12 Tendencia de los diseños de plantillas. [27]

	Plantilla 1	Plantilla 2	Plantilla 3
Colores (tonalidades)	X	-	X
Diseño (gráfica)	-	-	X
Usabilidad	X	X	

	Plantilla 1	Plantilla 2	Plantilla 3
Intuitiva (fácil acceso a las herramientas)	-	-	X

En base a los resultados obtenidos, se creó una cuarta plantilla, la cual está diseñada para que la herramienta sea intuitiva para el usuario, y de fácil acceso a las herramientas que posee. En el anexo B se puede observar la plantilla resultante. Estas plantillas fueron elaboradas bajo consideraciones de la investigación, pero por el diseñador gráfico José Luis Landívar.

3.2 Diseño Modular

La metodología de OSSTMM como ya se ha mencionado anteriormente se encuentra dividida en sectores, y cada sector tiene una serie de procesos a elaborar, por este motivo es que la elaboración de un diseño de datos y como se va a manejar la información dentro de la herramienta, resulta ser más adecuada para implementar, es por este motivo, que se ha decidido crear módulos para la elaboración de secciones, y módulos para los procesos. La ventaja de hacer esta consideración, es que en futuros cambios sobre la metodología, estos cambios puedan ser realizados de una manera muy intuitiva, ya sea por

aumentar un proceso, o aumentar alguna sección o modificarla, lo que evitará algún conflicto con el resto de procesos o plantillas establecidas. En la Figura 3.1 se especifica un esquema de cómo se realizarán los módulos, los cuales representan una respectiva tabla por parte de la base de datos.

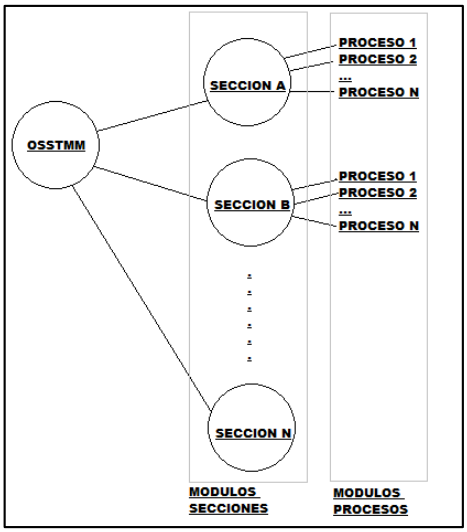


Figura 3.1 Esquema Modularizado de la Metodologías OSSTMM

Por lo tanto se procedió a realizar el Diagrama de Entidad Relación como se muestra en el Anexo C, por lo que se estableció la elaboración de unas tablas núcleo, que será la de psi_seccion y psi_proceso, los cuales tendrán todos las secciones y procesos respectivamente relacionados, por lo tanto cada sección y proceso se tendrá una tabla, que se relacionará con la principal. Es decir, si se requiere acoplar más

procesos o secciones para completar la metodología OSSTMM, la implementación sea eficaz. Además, esta estructura nos permite tener en la herramienta un creador de perfiles, los cuales configuraremos a nuestras necesidades, es decir dar la posibilidad al usuario de elegir que análisis puede realizar, y generar reportes solo de los configurado.

En el anexo D se especifica el correspondiente diccionario de datos, con respecto a las tablas creadas, y su debido flujo entre ellas, esta información es fundamental, para futuras implementaciones de la herramienta.

3.3 Usabilidad

Las herramientas que actualmente se encuentran disponibles, para realizar este tipo de tareas, o algo similar usando otros estándares de pruebas de seguridad, son muy complejas ya que han adoptado muchas herramientas adicionales, convirtiéndolas en aplicaciones de difícil uso, o que no sean amigables con el usuario, es decir, un analista que tiene el objetivo de realizar una prueba de penetración, o al menos tomar los correctivos correspondientes sobre su red, y usa estas herramientas, tiene que tener conocimiento adicional, o al menos saber exactamente donde se encuentran todas las utilidades necesarias para

cumplir su objetivo. Por ende se desarrolló una herramienta intuitiva, donde el único objetivo es generar reportes basados en una metodología, y que el usuario pueda seguir pasos sencillos, para llenar plantillas de forma guiada, y al final generar un reporte, el cual podrá ser entregado en los informes de las pruebas de seguridad. Ya que cuenta con plantillas ya diseñadas para ser válidas para la metodología OSSTMM.

La herramienta tiene una opción, la cual le permitirá al usuario crear proyectos, donde especificará que metodología usará, como es la herramienta inicial, solo tendrá la opción predeterminada de trabajar sobre OSSTMM. En la sección de requerimientos de este capítulo se especificó la sección la cual se iba a implementar, con sus respectivos procesos, el usuario podrá seleccionar que procesos analizará la herramienta, así como de permitirle elegir uno o más secciones por analiza, sin embargo para la presentación de este proyecto, solo está considerada la sección C Seguridad de Tecnologías de Internet, además habrán procesos que dependerán de otros por lo que se controla el acceso a estos procesos.

En un análisis de prueba de penetración, aunque el objetivo es revisar toda la red objetivo, existirá momentos en las cuales los consultores tendrán la necesidad de evaluar solo secciones de alguna metodología establecida, para el caso de la herramienta será la metodología OSSTMM, por esta razón, se ha desarrollado en la herramienta una utilidad lo cual le permite al usuario crear perfiles, que consiste en programar alguna prueba, es decir, escoger una metodología, la sección a analizar, y los procesos de los cuales van a generar reporte. Esta es una ventaja que se puede aprovechar, ya que no se pretende generar todo el reporte que OSSTMM exige, sin embargo por ser solo fragmentos no es recomendable incluir en alguna auditoría puesto que no cumple estrictamente con los estándares necesarios para la metodología.

Se mostrarán todos los procesos de la metodología OSSTM como se muestra en la Figura 3.1, pero estarán deshabilitados los procesos que no están considerados en este proyecto (marcados como deshabilitados en color naranja).

El usuario podrá agregar tantos documentos sea necesario para los procesos que se configuraran previamente, estos documentos se

mostrarán en el panel izquierdo, donde están clasificados según el proceso al que pertenecen, así mismo tendrá acceso a los reportes generados, por los mismos documentos creados, estos informes seguirán el estándar de seguridad informática usando la metodología OSSTMM.

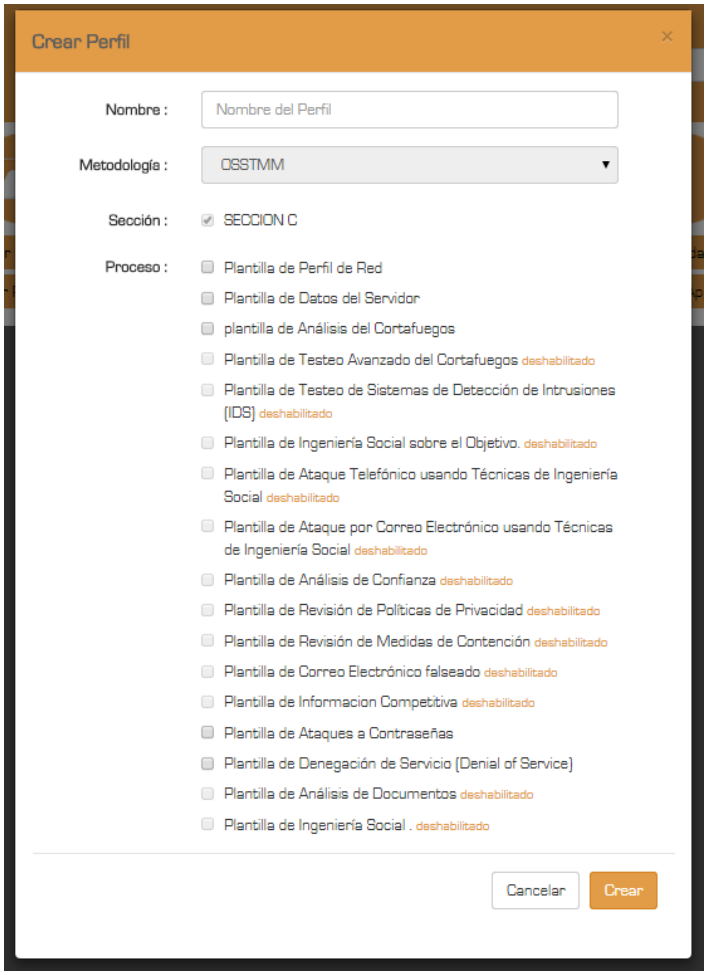


Figura 3.2 Panel de Creación de Perfiles

Además de generar los reportes que se configuraron, el usuario tiene la posibilidad de generar el reporte completo a la prueba de seguridad informática realiza

3.4 Consideraciones Operacionales

Hay dos entornos de trabajo que hay que considerar al momento de elaborar la herramienta web, estos entornos son el de producción y el de desarrollo. Por este motivo hay que tener una serie de precauciones antes de poner en funcionamiento la herramienta como son las versiones tanto del sistema operativo, como la del framework y las versiones de PHP y apache. El proyecto de la herramienta está desarrollado en el framework de Symfony, sin embargo existe 2 versiones del mismo, la versión 1.x es la antigua, y usa una jerarquía distinta en la distribución de carpetas y funcionamiento del mismo, por lo que lo esencial es que sea sobre la versión 2.x ya que bajo esas normas se realizó la herramienta web, además hay que considerar la versión de WAMP, en caso de estar en entorno de desarrollo, ya que es un conjunto de servicios como apache y php, dependiendo de la versión de WAMP, dependerá las versiones de PHP y Apache, por lo que se recomienda revisar la documentación respectiva [28]

Se mencionó también que el proyecto estaba realizado en Symfony 2.x esto quiere decir que admite cualquier subversión de la 2, por lo que hay que tener consideraciones previas al momento de instalar el resto de servicios como WAMP, ya que cada versión de Symfony maneja requerimientos mínimos en cuanto PHP se refiere. Por lo que en la tabla se puede visualizar algunas versiones de Symfony con el mínimo de requerimiento de PHP que se requiere, además se indica si es recomendado ya que de esas versiones se dejaron de dar soporte.

Tabla 13 Versiones de Symfony y PHP.

Symfony Versión	PHP Versión	Recomendado
2.0	≥5.3.2	No
2.1	≥5.3.3	No
2.2	≥5.3.3	No
2.3	≥5.3.3	Si
2.4	≥5.3.3	No
2.5	≥5.3.3	Si
2.6	≥5.3.3	Si

La herramienta web está diseñada usando MySQL como gestor de base de datos, el mismo que viene integrado en WAMP, sin embargo para

poder realizar modificaciones en el mismo es necesario, importar todas las tablas que se estén usando en el proyecto. Aunque los comandos de Symfony nos permiten crear automáticamente las tablas en la base de datos, esto genera el problema de que las configuraciones iniciales como la de los módulos de secciones y procesos no se encuentren disponibles. Además se debe realizar las configuraciones necesarias como la conexión a la base de datos que se vaya a usar, Symfony trae un servicio de manejo de base de datos el cual es Doctrine ORM, este paquete maneja ciertas bases de datos relacionales como MySQL, PostgreSQL, Microsoft SQL, además de otro paquete que es Doctrine ODM, que maneja base de datos no relacionadas como MongoDB, por lo que se debe de indicar las configuraciones que base de datos se usará; además de proveer las credenciales respectivas para la conexión.

3.5 Puesta en marcha

Considerando los estudios realizados, para la elaboración de la herramienta de pruebas de seguridad informática, se decidió realizarla en lenguaje de programación PHP, usando el framework Symfony, además de usar MySQL como gestor de la base de datos. Para poder realizar el desarrollo de esta aplicación, se la realizo sobre un equipo con sistema operativo Windows 8, con procesador Intel® Core™ i7-

4702MQ CPU @ 2.20 GHz, con memoria RAM de 8Gb, y para el entorno de desarrollo en un entorno de infraestructura de internet WAMP Server 2.4, que usa las herramientas de Apache 2.2.4, PHP 5.4.12 y MySQL 5.6.12, El framework de Symfony 2.6. Todos estos elementos son con los que se realizó el entorno de desarrollo. Sin embargo puede haber variantes sobre las versiones que se usen al momento de poner la aplicación en producción, ya que en un servidor web, se puede realizar instalaciones de cada servicio de manera independiente sin usar WAMP, por este motivo se especifica los requerimientos mínimos para que la herramienta web pueda ser puesta en producción, para obtener estos datos se procedió a revisar los requerimientos mínimos para desarrollar en Symfony, es decir según el sitio oficial de Symfony requiere como mínimo PHP 5.3.3. [29]

CAPÍTULO 4

4. COMPARATIVA DE RESULTADOS

En esta capítulo se describe las pruebas de seguridad que se realizaron, hacia objetivos específicos que legalmente están disponibles para realizar estas pruebas, así como de realizar estas mismas pruebas usando distintas herramientas de pruebas de seguridad informática que existen en el mercado, como son Dradis y Magic Tree.

4.1 Pruebas de seguridad en sistemas de pruebas

Las pruebas de seguridad que se realizaron estaban dirigidas específicamente a sitios con permisos para realizar pruebas de seguridad, donde existen vulnerabilidades destinadas a que sean descubiertas, uno de los sitios victimas es scanme.nmap.org también se realizó una prueba de seguridad informática en los salones de

computación de un Colegio de la ciudad de Duran, el cual por motivos de confidencialidad se reservará el nombre del mismo.

El sitio web scanme.nmap.org, es un servicio creado por Nmap, para que los usuarios de Nmap puedan comprobar y asegurar la seguridad en los puertos de todo los host asociados a este dominio, es decir, es un sitio que propone análisis de descubrimiento, escaneo y enumeración en un Hacking Ético, por lo tanto es una buena herramienta para el uso de las herramientas de pruebas de seguridad.

En este sitio web, ofrece una serie de vulnerabilidades, que se las puede descubrir, haciendo un correcto uso de las herramientas adecuadas, como nslookup, que nos permite averiguar todas las direcciones IP asociadas a un dominio web en particular, Nmap que nos permite averiguar la seguridad en los puertos, ya sea estos que estén abiertos, cerrados o filtrados, su estado dependerá de las destrezas del consultor, y que tanto sepa sobre el manejo de las herramientas.

Se realizaron las pruebas del sistema en un colegio del cantón de Duran. Dicho colegio cuenta con 3 laboratorios de informática donde se imparten clases para los alumnos, cada laboratorio está formado por 30,

20 y 35 máquinas las cuales por laboratorio se encuentran conectados a un conmutador, es decir hay tres de ellos, y el enrutador que se comunica con el ISP. El esquema se detalla en la figura 4.1 como se encuentra distribuida la red.

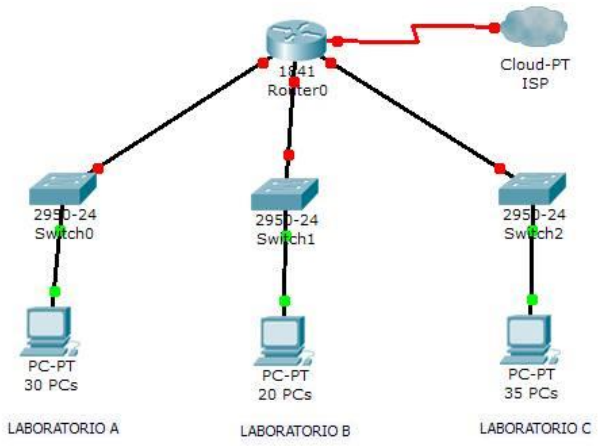


Figura 4.1 Topología de Red de los laboratorios de informática

4.2 Pruebas de seguridad con distintas herramientas

Para poder hacer una valoración detallada de que tan adecuada es la herramienta PSI sobre las otras herramientas, es necesario realizar las pruebas de seguridad anteriormente descritas usando cada uno de estas herramientas, sin embargo para poder evaluar la herramienta se

ha considerado a otras dos que son algo parecidas a lo que propone la herramienta PSI, estas herramientas son MagicTree y Dradis.

Existe muchos usuarios que mencionan que diseñar en Magic Tree es como Dradis ya que usan la misma implementación de árbol para desarrollar su lógica, pero existen ciertas diferencias que se pudieran considerar para el análisis y diferenciación entre las distintas herramientas a evaluar, como por ejemplo Dradis sigue un esquema de árbol por sucesos de la fuente, y Magic Tree realiza arboles de acuerdo a la situación presentada en las pruebas de seguridad, es decir en un escaneo de puerto se decide primero escanear por un puerto TCP y luego un puerto NMAP, Dradis establecerá dos ramas en el árbol lógico por cada importación de archivos, pero en Magic Tree, los dos escaneos realizados se fusionaran en un solo reporte creando una sola rama.

Dradis está escrito en Ruby y es de código abierto, es decir los usuarios pueden realizar cambios en su funcionamiento y compartirlo con la comunidad de Dradis, por otro lado Magic Tree es de código cerrado, por lo que no se permite modificar su código fuente, sin embargo, se pueden exportar escaneo realizados en Magic Tree hacia otras herramientas.

4.3 Comparación de resultados

Para los resultados obtenidos se recopiló información en base a los criterios y escenarios anteriormente mencionado es decir, con dos escenarios, uno el dominio de prueba scanme.nmap.org y pequeña red de laboratorios. Por este motivo se procedió a realizar el estudio de los tres primeros pasos en un hacking ético los cuales son: reconocimiento, escaneo y enumeración.

Para poder realizar los análisis de Reconocimiento necesitaremos herramientas de terceros como es un traductor de nombres de dominio nslookup, conocimiento de google hacking, investigación en los directorios de Who is, o haciendo el uso de herramientas como SmartWhois, o Maltego una importante herramienta para realizar minería de la información, para el análisis se usó la versión gratuita de la misma ya que no era necesario un análisis tan profundo del mismo; otras herramientas como Visual IP Trace, para averiguar la ruta que siguen los paquetes hasta llegar a su destino.

Para la etapa de Escaneo, que es donde se va a identificar los puerto que se encuentran abiertos, cerrados o filtrados, para esto se usó

herramientas adicionales como Ping Scanner Pro, que escanea un rango de direcciones IPs, sin embargo aquí se tomó precauciones para realizar las pruebas lo más real posible, ya sea planteando escenarios donde pueden existir personal de sistemas monitoreando los ping que se realizan hacia los servidores, es por eso que se realizó técnicas de escaneo de puerto, para poder inferir en los estados de los puertos, para esto se hizo uso de herramientas como Nmap.

La etapa de enumeración que consiste en conseguir la mayor cantidad de información acerca de la víctima, estas vulnerabilidades son aprovechados de los escaneos realizados previamente, se hizo uso de conocimiento sobre sesiones nulas, y mediante herramientas en línea de comando de Windows como “net view/domain”, nbtstat, nbtscan y manejo de sid de los usuario de las máquinas.

En base a los resultados obtenidos se procedió a usar las herramientas para poder documentar los resultados obtenidos, en la Tabla 14 se muestran las diferentes características tomadas en cuenta para la evaluación de las herramientas, donde las características que se deben recalcar es que la herramienta P.S.I. no cuenta con utilidades de terceros integradas, es decir que vienen incorporados en la aplicación,

sin embargo es necesario tomar en consideración que la herramienta P.S.I. maneja la generación de reportes basados en plantillas de la metodologías OSSTMM, por lo que resulta una gran ventaja para la documentación final de las auditorías.

Tabla 14 Comparativa de las Diferentes Herramientas Usadas

Características	P.S.I.	MagicTree	Dradis
Plataforma	Multiplataforma: PHP	Multiplataforma: Java	Multiplataforma: Ruby
Código Abierto	Sin especificar	No	Si
Arquitectura	Aplicación Web	Aplicación de Escritorio	Aplicación Web
Nmap	No	Si	Si
OpenVas	No	Si	Si
Agregar documentos	Si	Si	Si
Buscar información en los documentos	Si	si	Si
Manejo de Metodologías	Si	No	No
Reportes usando la Metodología OSSTMM	Si	No	No

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Los reportes en una consultoría para evaluar la seguridad de la información en una empresa son esenciales, ya que es el resultado del trabajo realizado en documentos, por lo que al hacer uso de herramientas que nos faciliten estas tareas, como de generar reportes en base a una metodología como la OSSTMM, resulta muy ventajoso ya que nos reduce el tiempo para la etapa de la entrega del informe en una auditoría.

2. El diseño en módulos de cualquier herramienta o aplicación es muy importante ya que nos permite entender de una mejor forma la arquitectura de cómo está diseñada, por lo que al hacer uso de esto, se pudo proponer

un proyecto en el cual pueda ser entendible para el desarrollador y con un reducido problemas de conflicto en los diferentes módulos.

3. Para poder modificar la metodología que se usa, es adecuado que se agregue un nuevo espacio de trabajo sobre el proyecto desarrollado en Symfony pues así se evita, tener inconsistencias en cuanto a la funcionalidad y arquitectura que está siguiendo la herramientas PSI.

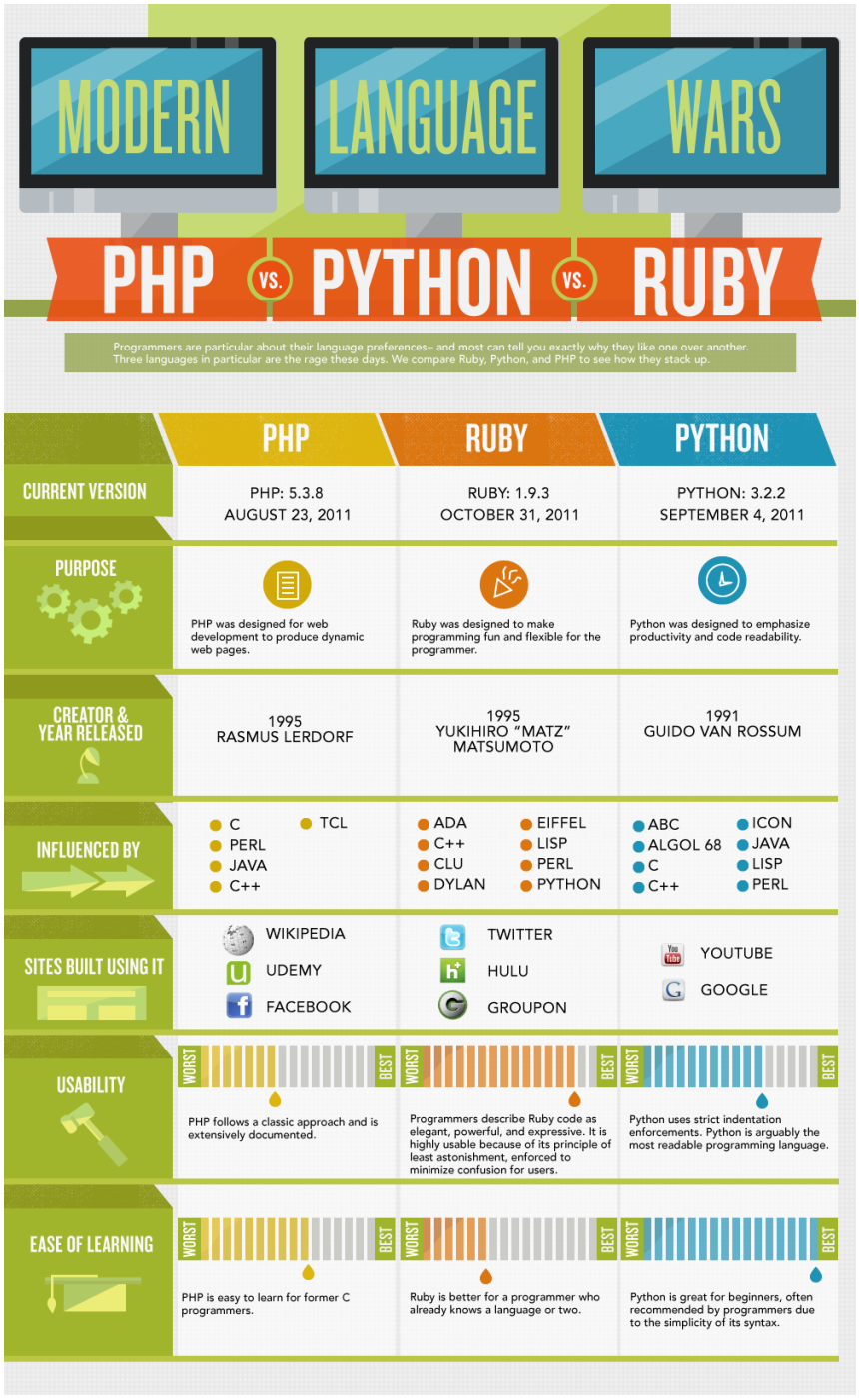
Recomendaciones

1. A pesar de que la herramienta de PSI no cuenta con utilitarios para la búsqueda de vulnerabilidades como Magic Tree y Dradis, sería adecuado trabajar con módulos para estas utilidades, y así poder implementarlas, sin embargo el agregar utilitarios se ve la necesidad de cambiar el entorno de trabajo donde se lo realiza actualmente ya que necesitaría utilitarios propios de distribuciones como Kali.

2. La herramienta web PSI, es adecuada para usuarios que son inexpertos en el asunto de seguridad informática, ya que le facilita crear perfiles sobre l metodología OSSTMM y así realizar pequeñas evaluaciones o las que crea convenientes.

ANEXOS

ANEXO A: Comparativa de PHP, Python y Ruby



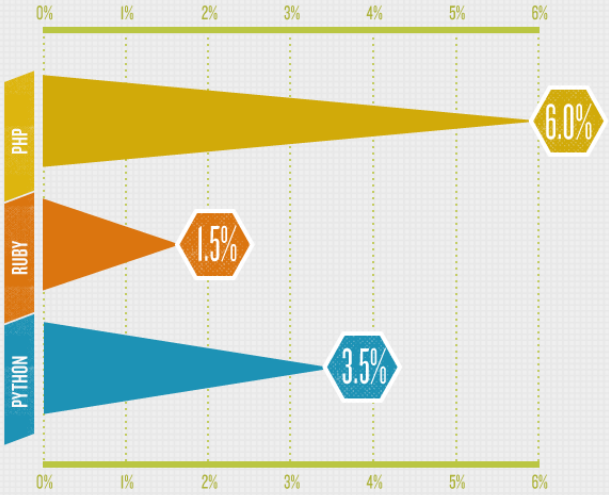
POPULARITY

Thanks to the TIOBE Programming Community we can find out which languages programmers prefer each year.

The TIOBE Programming Community Index is updated once a month. The ratings are based on the number of skilled engineers worldwide, courses, and third-party vendors.

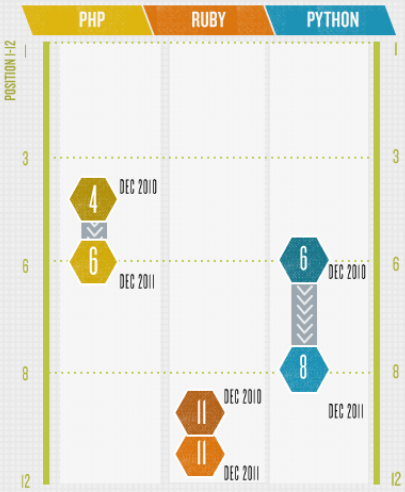
RATINGS AS OF DECEMBER 2011

The rating refers to the current preference rating on the TIOBE index.



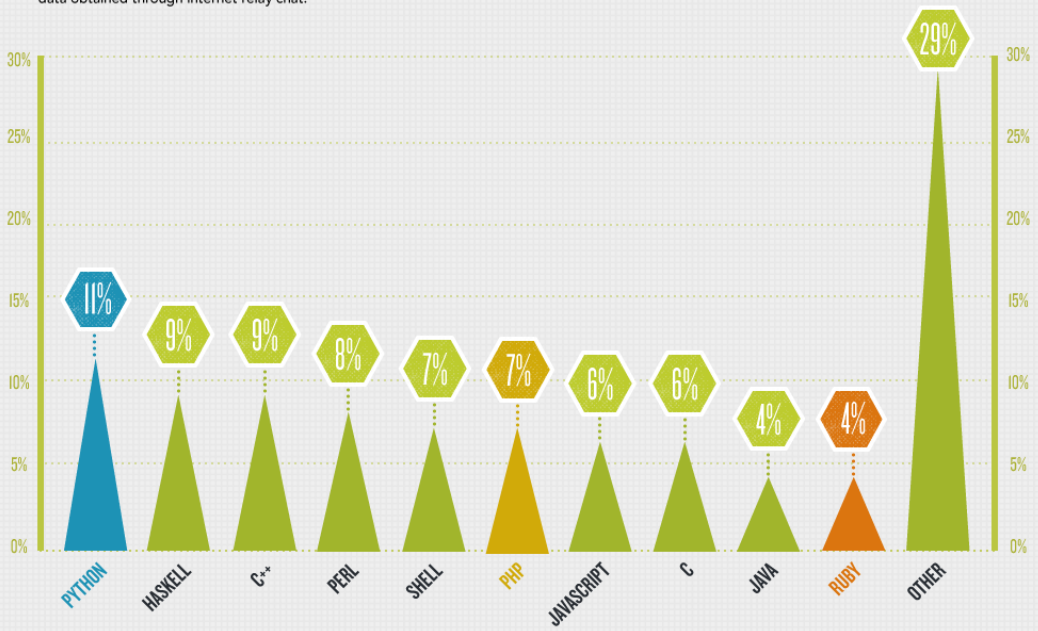
POSITION SHIFTS

This shows how each language's ranking has changed from December 2010 to December 2011.



MOST-DISCUSSED LANGUAGE

The popularity index below comes from IEEE Spectrum, which used data obtained through internet relay chat.



MARKETABILITY

When comparing these programming languages it is important to consider which is the most marketable at the moment. The type of developing you'll be doing will dictate which language is best for you to learn.

MOST JOB POSTINGS

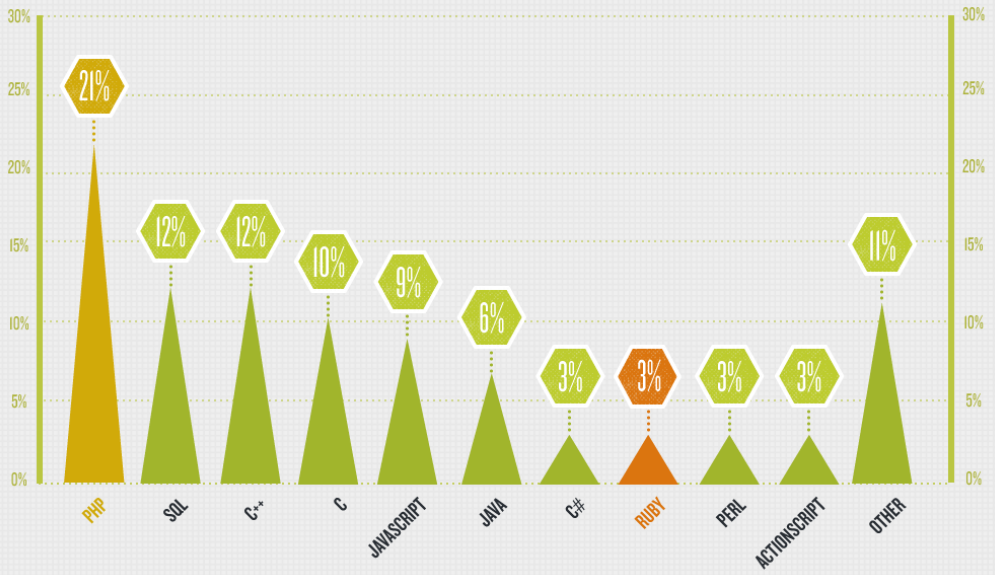
Looking for a programmer? According to Craigslist.com, most

MARKETABILITY

When comparing these programming languages it is important to consider which is the most marketable at the moment. The type of developing you'll be doing will dictate which language is best for you to learn.

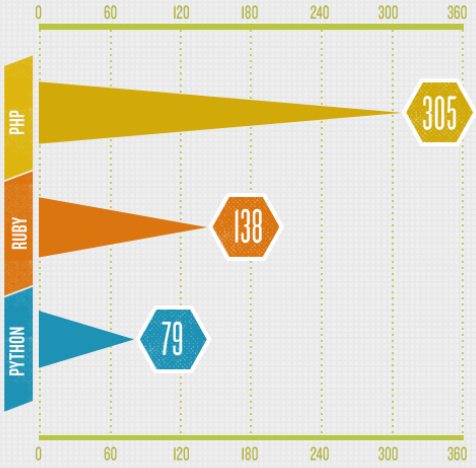
MOST JOB POSTINGS

Looking for a programmer? According to Craigslist.com, most companies are looking for developers fluent in PHP.



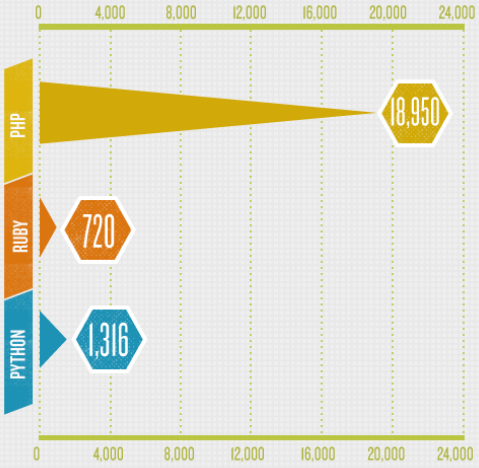
JOB POSTINGS ON MONSTER.COM

Below are the number of job postings as of December 12, 2011.



NUMBER OF DEVELOPERS

Here are the number of developers on LinkedIn as of December 15, 2011.



*Determined by searching for professionals with "Ruby developer," "Python developer," and "PHP developer."

BEST FOR

Which language would be best for the small business owner or job seeker?

SMALL BUSINESS

Based on the LinkedIn findings above, those businesses looking for developers will have the easiest time finding a PHP expert, followed by Python, then Ruby.

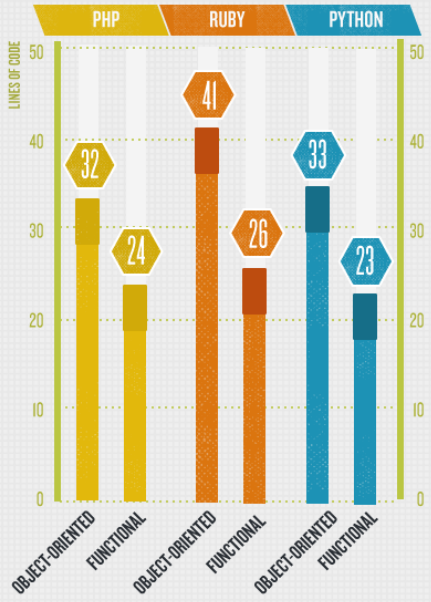
THE JOB SEEKER

Of all three languages, PHP is the most pervasive in the programming world followed by Python. Those two languages would be best to know if you are seeking a job.

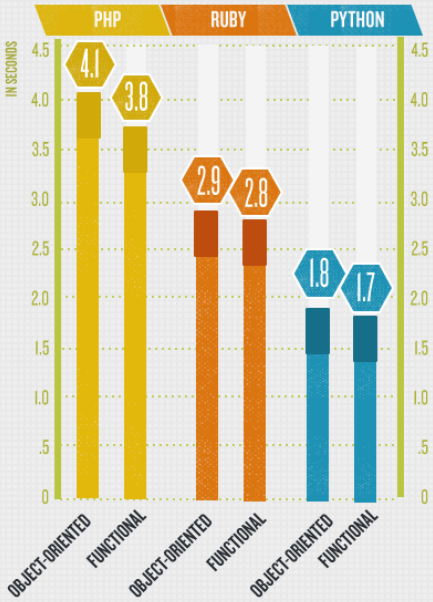
HOW FAST IS IT?

Using benchmark tests, we compare which languages are the fastest in terms of lines of code and average run time.

LINES OF CODE:



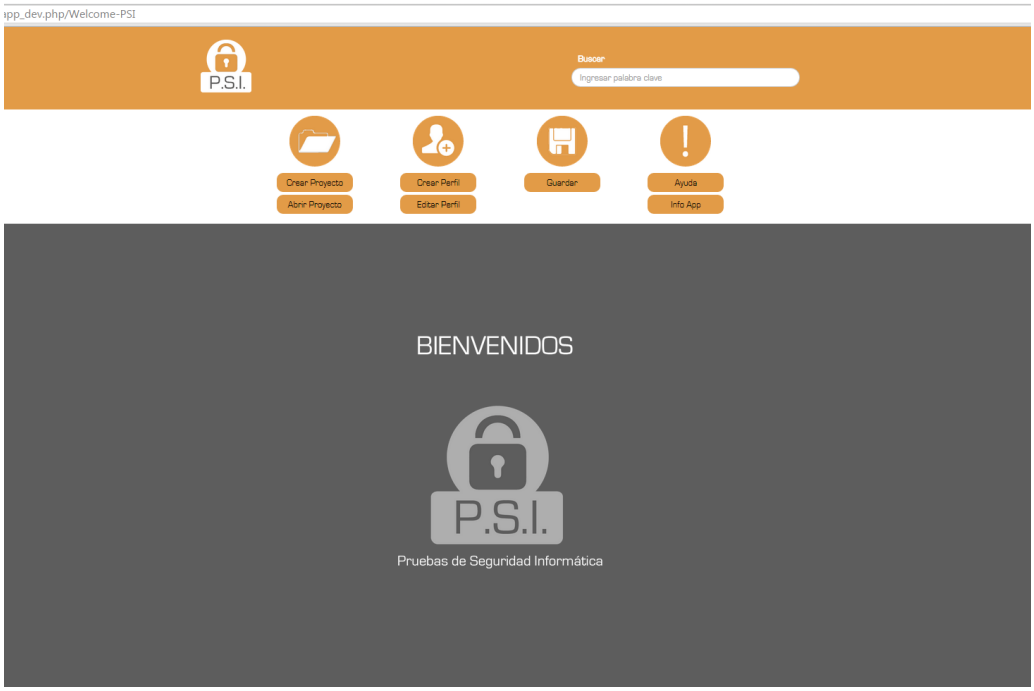
AVERAGE RUN TIME:

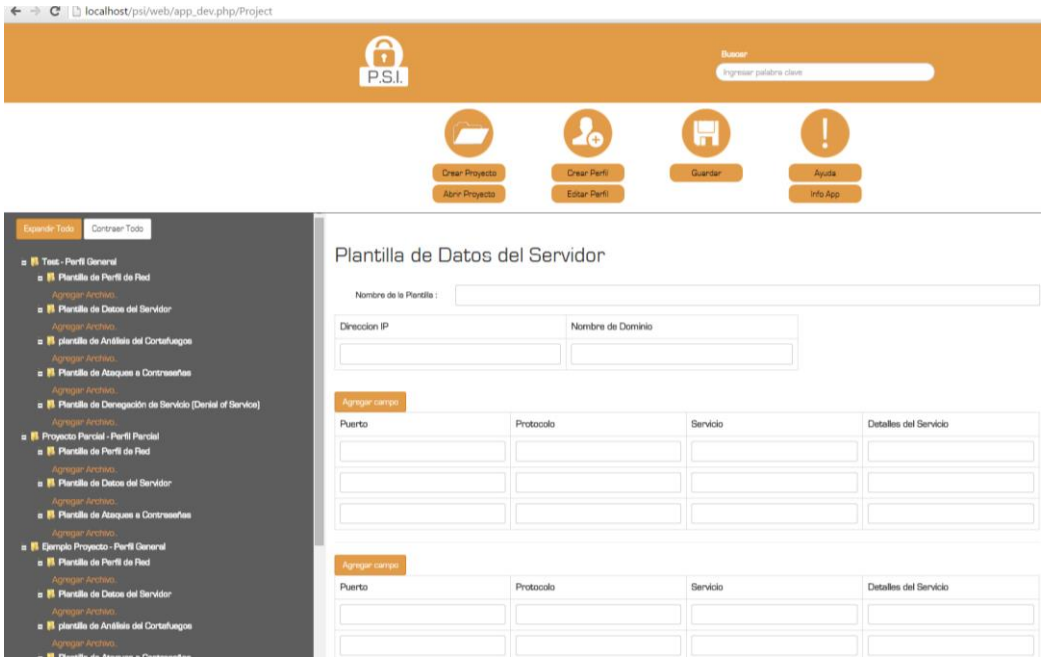


SOURCES: MJTSAI.COM | LCGEUROPE.COM | SHOOTOUT.AILOTH.DEBIAN.ORG | GITHUB.COM | C2.COM | RUBY-LANG.COM | PYTHON.ORG | "PHP VS. PYTHON VS. RUBY," KLAUS PURER | MONSTER.COM | LINKEDIN.COM | XODIAN.NET

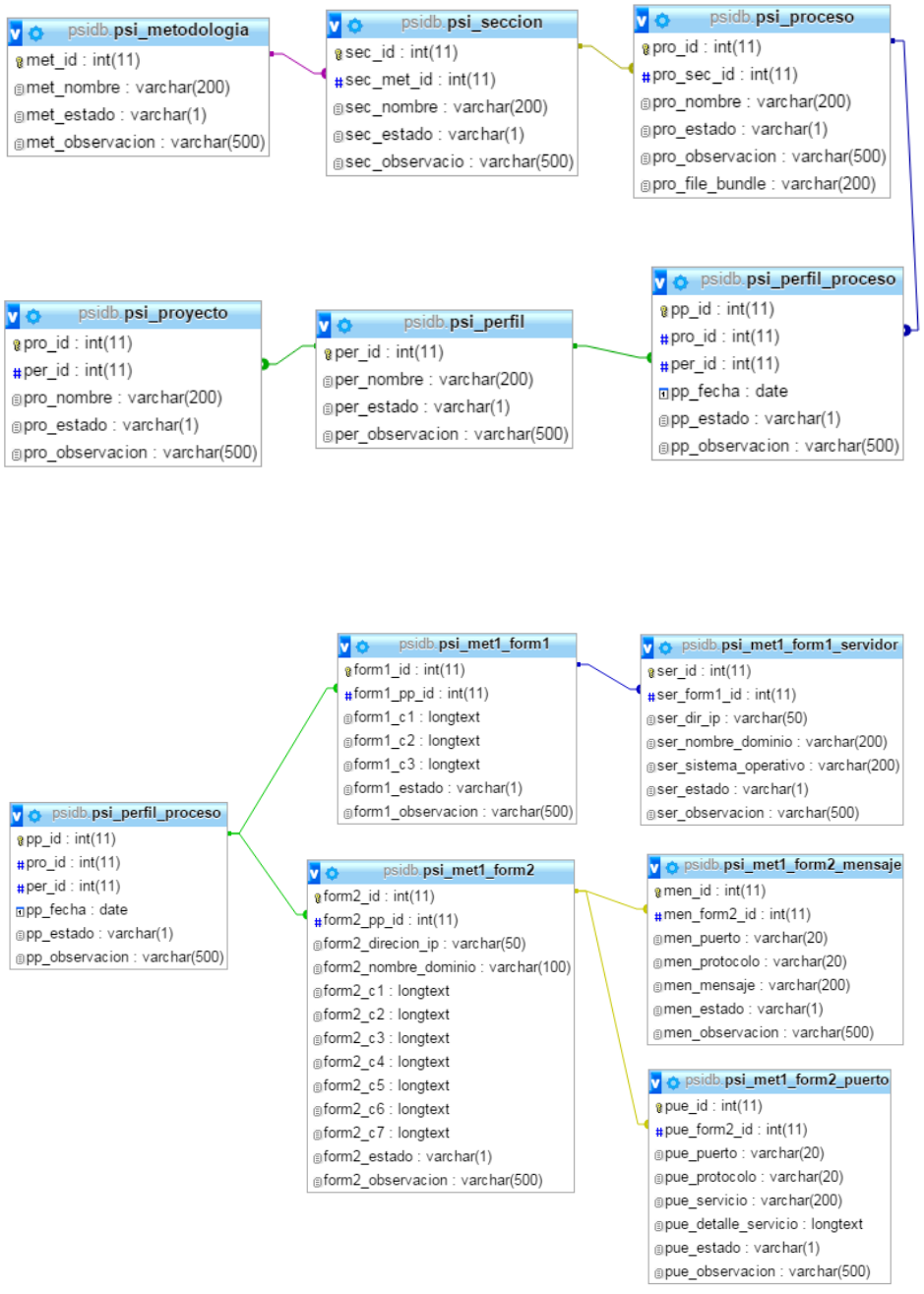
udemy

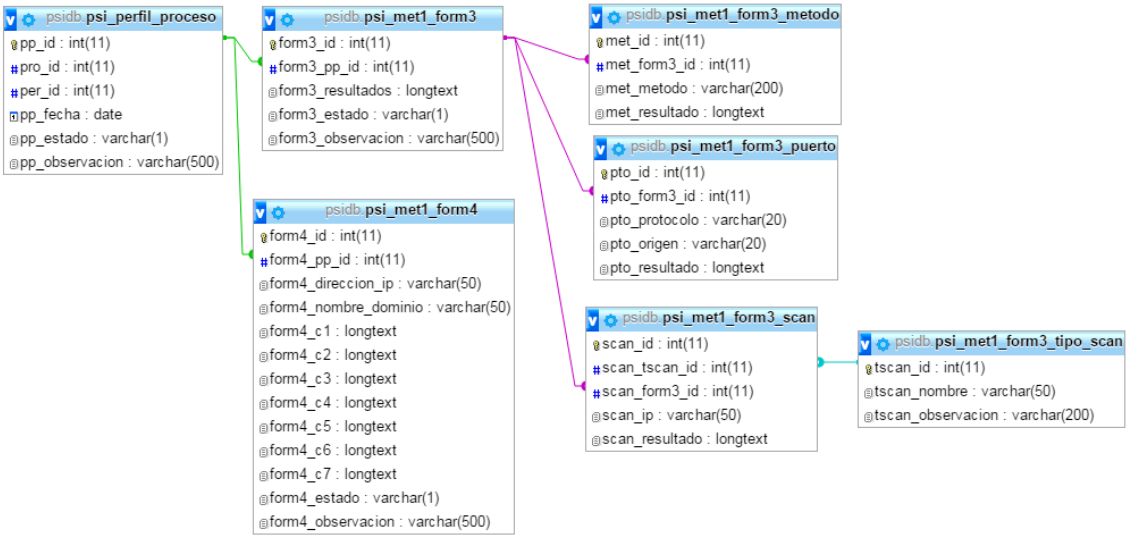
ANEXO B: Diseño de la herramienta web P.S.I.

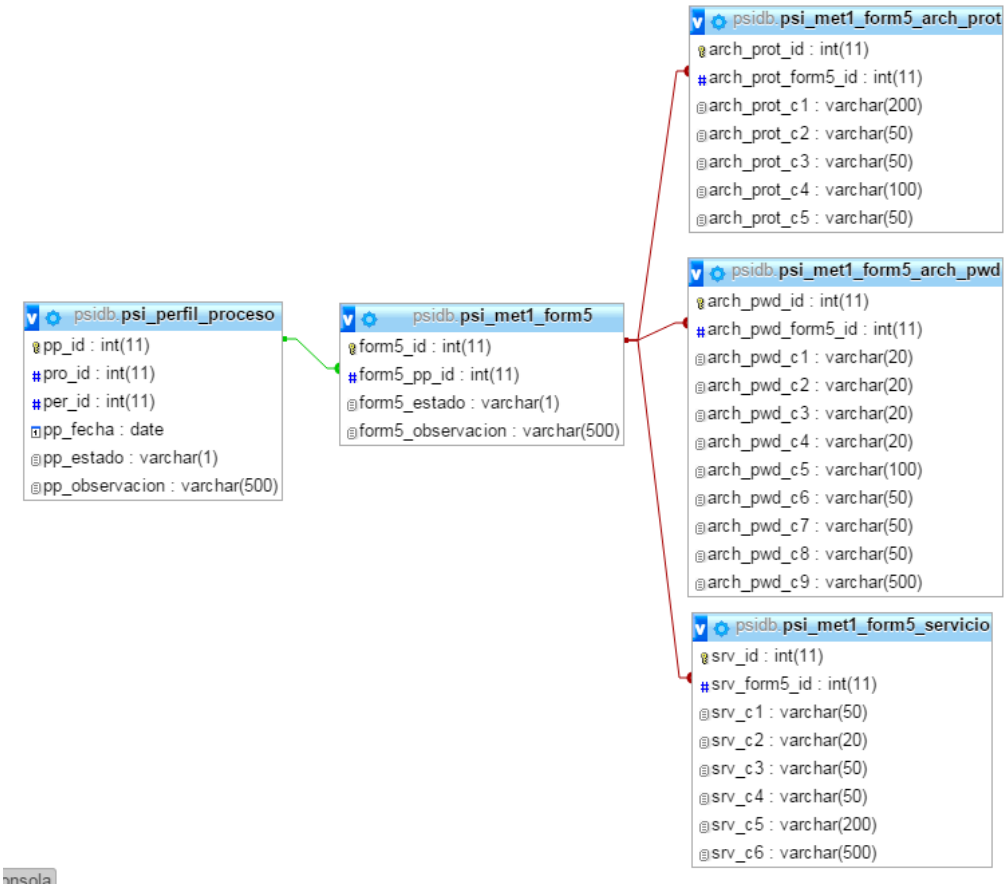




ANEXO C: Modelo Entidad Relación




psidb.psi_perfil_proceso
pp_id : int(11)
pro_id : int(11)
per_id : int(11)
pp_fecha : date
pp_estado : varchar(1)
pp_observacion : varchar(500)
psidb.psi_met1_form3
form3_id : int(11)
form3_pp_id : int(11)
form3_resultados : longtext
form3_estado : varchar(1)
form3_observacion : varchar(500)
psidb.psi_met1_form3_metodo
met_id : int(11)
met_form3_id : int(11)
met_metodo : varchar(200)
met_resultado : longtext
psidb.psi_met1_form3_puerto
pto_id : int(11)
pto_form3_id : int(11)
pto_protocolo : varchar(20)
pto_origen : varchar(20)
pto_resultado : longtext
psidb.psi_met1_form4
form4_id : int(11)
form4_pp_id : int(11)
form4_direccion_ip : varchar(50)
form4_nombre_dominio : varchar(50)
form4_c1 : longtext
form4_c2 : longtext
form4_c3 : longtext
form4_c4 : longtext
form4_c5 : longtext
form4_c6 : longtext
form4_c7 : longtext
form4_estado : varchar(1)
form4_observacion : varchar(500)
psidb.psi_met1_form3_scan
scan_id : int(11)
scan_tscan_id : int(11)
scan_form3_id : int(11)
scan_ip : varchar(50)
scan_resultado : longtext
psidb.psi_met1_form3_tipo_scan
tscan_id : int(11)
tscan_nombre : varchar(50)
tscan_observacion : varchar(200)


psidb.psi_met1_form5_arch_prot
arch_prot_id : int(11)
arch_prot_form5_id : int(11)
arch_prot_c1 : varchar(200)
arch_prot_c2 : varchar(50)
arch_prot_c3 : varchar(50)
arch_prot_c4 : varchar(100)
arch_prot_c5 : varchar(50)
psidb.psi_met1_form5_arch_pwd
arch_pwd_id : int(11)
arch_pwd_form5_id : int(11)
arch_pwd_c1 : varchar(20)
arch_pwd_c2 : varchar(20)
arch_pwd_c3 : varchar(20)
arch_pwd_c4 : varchar(20)
arch_pwd_c5 : varchar(100)
arch_pwd_c6 : varchar(50)
arch_pwd_c7 : varchar(50)
arch_pwd_c8 : varchar(50)
arch_pwd_c9 : varchar(500)
psidb.psi_met1_form5_servicio
srv_id : int(11)
srv_form5_id : int(11)
srv_c1 : varchar(50)
srv_c2 : varchar(20)
srv_c3 : varchar(50)
srv_c4 : varchar(50)
srv_c5 : varchar(200)
srv_c6 : varchar(500)
psidb.psi_perfil_proceso
pp_id : int(11)
pro_id : int(11)
per_id : int(11)
pp_fecha : date
pp_estado : varchar(1)
pp_observacion : varchar(500)
psidb.psi_met1_form5
form5_id : int(11)
form5_pp_id : int(11)
form5_estado : varchar(1)
form5_observacion : varchar(500)

ANEXO D: Diccionario de Datos de la Base de Datos

psi_met1_form1

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
form1_id *(Primaria)*	int(11)	No				
form1_pp_id	int(11)	Si	*NULL*	psi_perfil_proceso -> pp_id		
form1_c1	longtext	No				
form1_c2	longtext	No				
form1_c3	longtext	No				
form1_estado	varchar(1)	No				
form1_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	form1_id	0	A	No	
IDX_28A3A9609E90A4E6	BTREE	No	No	form1_pp_id	0	A	Si	

psi_met1_form1_servidor

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
ser_id *(Primaria)*	int(11)	No				
ser_form1_id	int(11)	Si	*NULL*	psi_met1_form1 -> form1_id		
ser_dir_ip	varchar(50)	No				
ser_nombre_dominio	varchar(200)	No				
ser_sistema_operativo	varchar(200)	No				
ser_estado	varchar(1)	No				
ser_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	ser_id	0	A	No	
IDX_A188593BCB707DF3	BTREE	No	No	ser_form1_id	0	A	Si	

psi_met1_form2

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
form2_id *(Primaria)*	int(11)	No				
form2_pp_id	int(11)	Si	*NULL*	psi_perfil_proceso -> pp_id		
form2_direcion_ip	varchar(50)	No				
form2_nombre_dominio	varchar(100)	No				
form2_c1	longtext	No				
form2_c2	longtext	No				
form2_c3	longtext	No				
form2_c4	longtext	No				
form2_c5	longtext	No				
form2_c6	longtext	No				

form2_c7	longtext	No				
form2_estado	varchar(1)	No				
form2_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	form2_id	0	A	No	
IDX_B1AAF8DAAF78BE7B	BTREE	No	No	form2_pp_id	0	A	Si	

psi_met1_form2_mensaje

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
men_id *(Primaria)*	int(11)	No				
men_form2_id	int(11)	Si	*NULL*	psi_met1_form2 -> form2_id		
men_puerto	varchar(20)	No				
men_protocolo	varchar(20)	No				
men_mensaje	varchar(200)	No				
men_estado	varchar(1)	No				
men_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	men_id	0	A	No	
IDX_BA47F13A17E8D99	BTREE	No	No	men_form2_id	0	A	Si	

psi_met1_form2_puerto

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
pue_id *(Primaria)*	int(11)	No				
pue_form2_id	int(11)	Si	*NULL*	psi_met1_form2 -> form2_id		
pue_puerto	varchar(20)	No				
pue_protocolo	varchar(20)	No				
pue_servicio	varchar(200)	No				
pue_detalle_servicio	longtext	No				
pue_estado	varchar(1)	No				
pue_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	pue_id	0	A	No	
IDX_D61936DDC5DF5EB7	BTREE	No	No	pue_form2_id	0	A	Si	

psi_met1_form3

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
form3_id *(Primaria)*	int(11)	No				

form3_pp_id	int(11)	Si	*NULL*	psi_perfil_proceso -> pp_id		
form3_resultados	longtext	No				
form3_estado	varchar(1)	No				
form3_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	form3_id	0	A	No	
IDX_C6ADC84C90FB5CF	BTREE	No	No	form3_pp_id	0	A	Si	

psi_met1_form3_metodo

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
met_id *(Primaria)*	int(11)	No				
met_form3_id	int(11)	Si	*NULL*	psi_met1_form3 -> form3_id		
met_metodo	varchar(200)	No				
met_resultado	longtext	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	met_id	0	A	No	
IDX_7382A20C894EFBDD	BTREE	No	No	met_form3_id	0	A	Si	

psi_met1_form3_puerto

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
pto_id *(Primaria)*	int(11)	No				
pto_form3_id	int(11)	Si	*NULL*	psi_met1_form3 -> form3_id		
pto_protocolo	varchar(20)	No				
pto_origen	varchar(20)	No				
pto_resultado	longtext	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	pto_id	0	A	No	
IDX_1AB33643ABCF76DB	BTREE	No	No	pto_form3_id	0	A	Si	

psi_met1_form3_scan

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
scan_id *(Primaria)*	int(11)	No				
scan_tscan_id	int(11)	Si	*NULL*	psi_met1_form3_tipo_scan -> tscan_id		
scan_form3_id	int(11)	Si	*NULL*	psi_met1_form3 -> form3_id		
scan_ip	varchar(50)	No				
scan_resultado	longtext	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	scan_id	0	A	No	
IDX_FB6C0EEF3948F16B	BTREE	No	No	scan_tscan_id	0	A	Si	
IDX_FB6C0EEF93B5DAC7	BTREE	No	No	scan_form3_id	0	A	Si	

psi_met1_form3_tipo_scan

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
tscan_id *(Primaria)*	int(11)	No				
tscan_nombre	varchar(50)	No				
tscan_observacion	varchar(200)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	tscan_id	0	A	No	

psi_met1_form4

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
form4_id *(Primaria)*	int(11)	No				
form4_pp_id	int(11)	Si	*NULL*	psi_perfil_proceso -> pp_id		
form4_direccion_ip	varchar(50)	No				
form4_nombre_dominio	varchar(50)	No				
form4_c1	longtext	No				
form4_c2	longtext	No				
form4_c3	longtext	No				
form4_c4	longtext	No				
form4_c5	longtext	No				
form4_c6	longtext	No				
form4_c7	longtext	No				
form4_estado	varchar(1)	No				
form4_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	form4_id	0	A	No	
IDX_58C95DEFCCA88B41	BTREE	No	No	form4_pp_id	0	A	Si	

psi_met1_form5

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
form5_id *(Primaria)*	int(11)	No				
form5_pp_id	int(11)	Si	*NULL*	psi_perfil_proceso -> pp_id		
form5_estado	varchar(1)	No				

form5_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	form5_id	0	A	No	
IDX_2FCE6D796ADF80F5	BTREE	No	No	form5_pp_id	0	A	Si	

psi_met1_form5_arch_prot

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
arch_prot_id *(Primaria)*	int(11)	No				
arch_prot_form5_id	int(11)	Si	*NULL*	psi_met1_form5 -> form5_id		
arch_prot_c1	varchar(200)	No				
arch_prot_c2	varchar(50)	No				
arch_prot_c3	varchar(50)	No				
arch_prot_c4	varchar(100)	No				
arch_prot_c5	varchar(50)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	arch_prot_id	0	A	No	
IDX_92AC3FD2C28C14D1	BTREE	No	No	arch_prot_form5_id	0	A	Si	

psi_met1_form5_arch_pwd

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
arch_pwd_id *(Primaria)*	int(11)	No				
arch_pwd_form5_id	int(11)	Si	*NULL*	psi_met1_form5 -> form5_id		
arch_pwd_c1	varchar(20)	No				
arch_pwd_c2	varchar(20)	No				
arch_pwd_c3	varchar(20)	No				
arch_pwd_c4	varchar(20)	No				
arch_pwd_c5	varchar(100)	No				
arch_pwd_c6	varchar(50)	No				
arch_pwd_c7	varchar(50)	No				
arch_pwd_c8	varchar(50)	No				
arch_pwd_c9	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	arch_pwd_id	0	A	No	
IDX_9B7E14533324D5D5	BTREE	No	No	arch_pwd_form5_id	0	A	Si	

psi_met1_form5_servicio

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME

srv_id *(Primaria)*	int(11)	No				
srv_form5_id	int(11)	Si	*NULL*	psi_met1_form5 -> form5_id		
srv_c1	varchar(50)	No				
srv_c2	varchar(20)	No				
srv_c3	varchar(50)	No				
srv_c4	varchar(50)	No				
srv_c5	varchar(200)	No				
srv_c6	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	srv_id	0	A	No	
IDX_9870249BCA349B96	BTREE	No	No	srv_form5_id	0	A	Si	

psi_metodologia

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
met_id *(Primaria)*	int(11)	No				
met_nombre	varchar(200)	No				
met_estado	varchar(1)	No				
met_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	met_id	0	A	No	

psi_perfil

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
per_id *(Primaria)*	int(11)	No				
per_nombre	varchar(200)	No				
per_estado	varchar(1)	Si	*NULL*			
per_observacion	varchar(500)	Si	*NULL*			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	per_id	2	A	No	

psi_perfil_proceso

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
pp_id *(Primaria)*	int(11)	No				
pro_id	int(11)	Si	*NULL*	psi_proceso -> pro_id		
per_id	int(11)	Si	*NULL*	psi_perfil -> per_id		
pp_fecha	date	No				
pp_estado	varchar(1)	No				

pp_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	pp_id	8	A	No	
IDX_3A06D4DBC3B7E4BA	BTREE	No	No	pro_id	8	A	Si	
IDX_3A06D4DBB304206A	BTREE	No	No	per_id	4	A	Si	

psi_proceso

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
pro_id *(Primaria)*	int(11)	No				
pro_sec_id	int(11)	Si	*NULL*	psi_seccion -> sec_id		
pro_nombre	varchar(200)	No				
pro_estado	varchar(1)	No				
pro_observacion	varchar(500)	No				
pro_file_bundle	varchar(200)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	pro_id	17	A	No	
IDX_FB6D90B67B37EE69	BTREE	No	No	pro_sec_id	2	A	Si	

psi_proyecto

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
pro_id *(Primaria)*	int(11)	No				
per_id	int(11)	Si	*NULL*	psi_perfil -> per_id		
pro_nombre	varchar(200)	No				
pro_estado	varchar(1)	No				
pro_observacion	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	pro_id	2	A	No	
IDX_B2B60FA4B304206A	BTREE	No	No	per_id	2	A	Si	

psi_seccion

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
sec_id *(Primaria)*	int(11)	No				
sec_met_id	int(11)	Si	*NULL*	psi_metodologia -> met_id		
sec_nombre	varchar(200)	No				
sec_estado	varchar(1)	No				
sec_observacio	varchar(500)	No				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	sec_id	0	A	No	
IDX_89CCC1A6FB1C6350	BTREE	No	No	sec_met_id	0	A	Si	

BIBLIOGRAFÍA

[1] Anónimo, Delitos informáticos en la web podrían aumentar en Ecuador, http://www.eluniverso.com/noticias/2014/11/17/nota/4226966/ataques-web-podrian-aumentar, fecha de consulta 10 Enero 2015.

[2] Junta de Andalucia, Manual de la Metodología Abierta de Testeo de Seguridad (OSSTMM), http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/551, fecha de consulta 10 Enero 2015.

[3] CONSULTING INFORMATION TECHNOLOGY S.A, COBIT 5, http://es.slideshare.net/cbonilla1967/resume-cobit-5., fecha de consulta 25 Febrero 2015.

[4] Anónimo, ISO 27002, http://www.iso27002.es/, fecha de consulta 27 Febrero 2015.

[5] Pete Herzog, Open Source Security Testing Methodology Manual (OSSTMM),: http://www.isecom.org/research/osstmm.html, fecha de consulta 28 Febrero 2015.

[6] S. F. Wilson, Analyzing Requirements and Defining Solution Architectures, Redmond: Microsoft Press, 1999.

[7] Richardisai's Blog, DESCOMPOSICION MODULAR, https://richardisai.wordpress.com/descomposicion-modular/, fecha de consulta 2 Mazo 2015.

[8] Anónimo, El modelo cliente - servidor, http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html, fecha de consulta 05 Marzo 2015.

[9] Anónimo, PHP, http://php.net/manual/es/history.php.php, fecha de consulta 25 Enero 2015.

[10] Anónimo, PHP, http://php.net/manual/es/about.phpversions.php, fecha de consulta 25 Enero 2015.

[11] Laravel, Laravel, 2015. http://laravel.com/docs/4.2/introduction, fecha de consulta 25 Enero 2015.

[12] Phalcon, Phalcon Documentation, 2014. http://docs.phalconphp.com/es/latest/reference/motivation.html, fecha de consulta 25 Enero 2015.

[13] Víctor Ruiz, Phalcon, el Framework PHP más rápido del Mundo, 2 Julio 2014. http://mycyberacademy.com/phalcon-el-framework-php-mas-rapido-del-mundo/, fecha de consulta 21 Enero 2015.

[14] F. Potencier, SensioLabsNetwork, 25 Octubre 2011. http://fabien.potencier.org/article/49/what-is-symfony2, fecha de consulta 28 Enero 2015.

[15] J. Knowlton, Python, Anaya Multimedia-Anaya Interactiva, 2009.

[16] V. Bill , The Making of Python, 13 Enero 2003. http://www.artima.com/intv/pythonP.html, fecha de consulta 27 Enero 2015.

[17] PYTHON, Python Documentation, 01 Enero 2015. https://docs.python.org/3.5/license.html, fecha de consulta 27 Febrero 2015.

[18] Django Software Foundation, ¡Descubre Django! http://www.django.es/, fecha de consulta 29 Enero 2015.

[19] Django Software Foundation, Django Wiki, Septiembre 2014. https://code.djangoproject.com/, fecha de consulta 29 Enero 2015.

[20] Web2py, web2pyTM Documentation & Resources, http://web2py.com/init/default/documentation, fecha de consulta 01 Febrero 2015.

[21] web2py, Security, http://www.web2py.com/book/default/chapter/01#Security, fecha de consulta 29 Enero 2015.

[22] Y. Matsumoto, Interviewee, *The Philosophy of Ruby.* [Entrevista]. 29 Septiembre 2003.

[23] David, Rails 2.0: It's done!, 7 Diciembre 2007. http://weblog.rubyonrails.org/2007/12/7/rails-2-0-it-s-done, fecha de consulta 02 02 2015.

[24] Sinatra, Getting Started, http://www.sinatrarb.com/intro.html, fecha de consulta 03 Febrero 2015.

[25] Andrearrs, Sinatra: minimalismo para el desarrollo web en Ruby,.http://hipertextual.com/archivo/2014/08/sinatra-minimalismo-para-desarrollo-web-ruby/, fecha de consulta 05 Febrero 2015.

[26] L. Fridman, Una comparación del rendimiento de los frameworks de Ruby: Sinatra, Padrino, Goliat y Ruby on Rails, http://altoros.com.ar/blog/una-comparacion-del-rendimiento-de-los-frameworks-de-ruby-sinatra-padrino-goliat-y-ruby-on-rails/, fecha de consulta 08 Febrero 2015.

[27] ISECOM, OSSTMM 2.1, 23 Agosto 2003. http://isecom.securenetltd.com/osstmm.en.2.1.pdf, fecha de consulta 20 Abril 2015.

[28] WAMPSERVER, WampServer, http://www.wampserver.com/en/, fecha de consulta 29 Abril 2015.

[29] SensioLabs, Symfony, http://symfony.com/doc/current/reference/requirements.html, fecha de consulta 20 Abril 2015].