UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS

Escuela de Ingeniería Informática

Carrera de Ingeniería en Sistemas de Información

PRACTICA PROFESIONAL DIRIGIDA

Para optar por el grado de Bachillerato en Sistemas de Información

Desarrollo de un prototipo funcional de un plugin para Visual Studio .NET que permita realizar pruebas estáticas de seguridad de aplicaciones (SAST)

SUSTENTANTE

Michael Hidalgo Fallas

TUTOR

Ing. Leonardo Delgado Arroyo, MAP

San José, Costa Rica

Septiembre, 2014

# Análisis FODA

Las siglas FODA hacen referencia a los conceptos de fortalezas, oportunidades, debilidades y amenazas respectivamente. El autor Ioannou (2012), refiriéndose a este tema detalla que “El análisis FODA, también conocido como matriz FODA es un método usado para determinar las fortalezas, debilidades, oportunidades y amenazas dentro de un proyecto o dentro del entorno de una empresa.”. Este mismo autor además define un punto verdaderamente importante para poder identificar cada uno de estos elementos al decir que :

Las fortalezas y las debilidades son factores internos, los cuales son atribuidos a las acciones directas o a control o a la compañía o a los empleados. Las oportunidades y las amenazas por otro lado, son factores externos y están fuera de su control. (p. 2).

La importancia de este análisis es definido ampliamente por Lawrence G. Fine (2011) cuando indica que:

Decisiones importantes son tomadas todos los días por cada uno de nosotros. Hay momentos en que necesitamos hacer un rápido juicio y basamos esas decisiones en la información que tenemos disponible. Sin embargo, hay otras situaciones donde hay que observar diferentes factores disponibles y en éstos casos es cuando se necesita usar el análisis FODA (p. 1).

Partiendo de los dos enfoques propuestos por los autores, se puede comprender que el análisis FODA es una herramienta o técnica muy importante para el proceso de toma de decisiones. Inclusive Fine (2011) sostiene que “El resultado del análisis FODA es ver la realidad de su negocio o sus ideas. También le dará una lista de puntos de acción, los cuales usted deberá seguir”.

Es relevante comprender cada uno de los elementos que forman parte del análisis FODA, a fin de proceder de forma correcta en el momento de recopilar la información y de principalmente identificar como afectan dichos elementos el prototipo funcional propuesto.

* Fortaleza: Las fortalezas forman parte del análisis interno de la organización. Una definición más amplia brindada por Ioannou (2012) detalla que “La fortaleza de una compañía son sus recursos y las capacidades que pueden ser usadas para desarrollar una ventaja competitiva”.
* Debilidades o Limitaciones: Las debilidades también forman parte del análisis interno. Claramente Ioannou (2012) dice que una debilidad es “la ausencia de una fortaleza en la organización”. Es muy importante reconocer las debilidades, principalmente porque en términos de Fine (2011), se puede concluir en que las debilidades existen en toda organización y es necesario determinar como se va a lidiar con ellas.
* Oportunidades : Las oportunidades son situaciones que se presentan y que pueden generar un beneficio para la organización. Identificar las oportunidades es un proceso importante, ya que palabras de Ioannou (2012), el proceso de identificar oportunidades consiste en “Identificar oportunidades de crecimiento o de mejorar la rentabilidad de un negocio”. Así mismo, refiriéndose a oportunidades Fine (2011) dice que “Hay oportunidades alrededor de nosotros, pero la mayoría del tiempo las perdemos porque estamos demasiado enfocados en aspectos negativos”.
* Amenazas: Una amenaza es un factor externo que puede causar un efecto adverso o negativo a la organización. En términos de riesgo, el experto en seguidad Paul(2011) se refiere a una amenaza como “… es meramente la posibilidad de que un evento potencialmente dañino o no deseado ocurra”. Para Ioannou (2012), algunos ejemplos de amenazas lo constituyen los competidores que venden el mismo producto, cambios bruscos en los impuestos, trabajadores habilidosos escasos.

Tomando en consideración el punto de vista de los autores y de los elementos que constituyen el análisis FODA, se puede rescatar que realizar éste análisis de forma correcta ayuda de forma sustancial en el proceso de tomar decisiones de forma más asertiva.

## Análisis FODA para el prototipo funcional.

A continuación se presenta en forma de matriz los principales elementos que forman parte del FODA para el prototipo funcional. Dicha matriz agrupa los factores internos y externos de forma tal que su apreciación sea clara.

|  |  |  |
| --- | --- | --- |
| Internas | | |
| Fortalezas | | Debilidades |
| * Amplia experiencia en el mercado de la seguridad de las aplicaciones. * Mayor comercialización de los productos. * Integración con empresas en el mercado de la seguridad de aplicaciones. * Empresa cuenta con áreas de investigación y desarrollo donde se produce tecnología de vanguardia. * Facultar a empresas a desarrollar aplicaciones de software más seguras. | | * Poca o nula inserción en el campo del análisis estático de código. * Dependencia de terceras empresas para realizar el análisis estático de código. * Proyectos de código abierto y gratuito ofrecen productos similares a muy bajo costo. * Plugin limitado a un lenguaje de programación y a un entorno integrado de desarrollo. |
| Externas | | |
| Oportunidades | Amenazas | |
| * Creciente demanda en seguridad de aplicaciones por parte de la industria. * Rápida evolución del lenguaje de programación C#. * Herramienta integrada en el ambiente de desarrollo. * Acercamiento de nuevos clientes potenciales. | * Competencia ofrece productos similares e integrados. * Clientes prefieren productos unificados. * Proliferación de herramientas en línea para capacitar personal. | |

## Fortalezas

### Amplia experiencia en el mercado de la seguridad de las aplicaciones.

La empresa Security Innovation desde su fundación en el año 2001, se ha convertido en una autoridad en el tema de seguridad en el software, brindando herramientas y mejores prácticas para facultar a la industria de mecanismos para desarrollar software que sea seguro. La elaboración del prototipo funcional acá propuesto cuenta con todas las lecciones aprendidas a lo largo de los años en la organización, incluyendo el talento de profesionales en materia de seguridad informática, a los cuales se acudirá de forma recurrente, que constituye un pilar fundamental para desarrollar los principios básicos de seguridad de forma efectiva.

### Mayor comercialización de los productos :

La elaboración del prototipo funcional permitirá que los productos desarrollados por la organización, principalmente las guías y estándares, puedan ser comercializados más fácilmente, ya que el prototipo hará referencia a las mejores prácticas desarrolladas por la organización así como las recomendaciones pertinentes, de forma tal que clientes potenciales puedan a su vez conocer más de los productos que la empresa ofrece.

### Integración con empresas en el mercado de la seguridad de las aplicaciones.

La organización ha trabajado en paralelo con gigantes de la industria informática en el mercado de la seguridad de la aplicaciones entre las que se encuentran HP Fortify, IBM y Checkmarx. Estas empresas a su vez tienen trayectoria elaborando herramientas para mejorar la seguridad en las aplicaciones de software. El hecho de trabajar en conjunto con terceras partes faculta que el conocimiento en áreas de investigación y desarrollo sea incorporado en los nuevos productos que se desarrollan.

### Empresa cuenta con áreas de investigación y desarrollo donde se produce tecnología de vanguardia.

La empresa Security Innovation ha estado inmersa en proyectos de investigación y tecnología de punta en temas como lo son criptografía y algoritmos de encripción modernos y eficientes hasta trabajar en conjunto con la industria de los sistemas de transporte inteligente, donde se busca disminuir la cantidad de accidentes de tránsito en carretera por medio de la tecnología. El hecho de participar en proyectos innovadores ha hecho que la empresa se reconocida aún más, de forma tal que el prototipo funcional propuesto busca formar parte de la lista de aplicaciones innovadoras ofrecidos por la organización.

### Facultar a empresas a desarrollar aplicaciones de software más seguras.

El prototipo funcional propuesto viene a solventar una necesidad recurrente en el mercado, la cual consiste en brindarle a los desarrolladores de software retroalimentación en términos de seguridad cuando desarrollan aplicaciones. A lo largo de los años, Security Innovation ha tenido como objetivo brindar herramientas, guías y recursos educación (entre los que se destacan entrenamientos en línea y bases de datos de conocimiento) para que las organizaciones implementen software seguro. El prototipo funcional para realizar análisis estático de código se alinea perfectamente con los objetivos de la empresa y sus áreas de acción, facultando a los clientes a optar por nuevos productos.

## Oportunidades

### Creciente demanda en seguridad de aplicaciones por parte de la industria.

En un panorama como el actual donde la cantidad de incidentes en seguridad informática va en aumento, surge la necesidad de implementar mecanismos para evitar ser víctima de un ataque cibernético. Paul, Mano (2011) es claro cuando indica que :

En una época cuando las brechas de seguridad en el software le están costando a las compañías multas más grandes y cargas regulatorias, desarrollar software que sea confiable en su funcionalidad, resistente a ataques, y que sea recuperable cuando se interrumpen las operaciones del negocio es una necesidad. (p 1).

A medida que esta necesidad de desarrollar software que sea resistente a ataques informáticos se vuelve imperativa, las empresas están solicitando herramientas que les ayuden a protegerse y evitar ser víctimas de un eventual ataque cibernético. El prototipo funcional propuesto para desarrollar análisis estático de código se ajusta perfectamente a esta realidad.

### Rápida evolución del lenguaje de programación C#

El lenguaje de programación C# (pronunciado C Sharp) ha tenido una constante y rápida evolución, permitiendo incorporar nuevas características donde se desarrolla aplicaciones de software más rápido que a su vez presenta código más atractivo semántica y sintácticamente. Aunado a la rápida evolución se destaca la apertura del compilador de C# por medio de una interfaz de programación de aplicaciones (API por sus siglas en inglés), facultando a los ingenieros a desarrollar aplicaciones que anteriormente no eran posibles de construir, abriendo de esta forma un nuevo camino en el campo de la investigación y desarrollo de aplicaciones y componentes dentro del ambiente de desarrollo integrado. El prototipo funcional utilizará las características proporcionadas por el lenguaje de programación a fin de poder crear un producto que sea eventualmente extensible.

### Herramienta integrada en el ambiente de desarrollo

El componente de seguridad se instalará en el ambiente de desarrollo integrado denominado Visual Studio .NET. Tradicionalmente, los desarrolladores de software están familiarizados con componentes de terceras empresas que les permitan personalizar el ambiente, entre los que se incluyen componentes para generación de código, reingeniería, búsquedas, ordenamiento entre otras. Este componente utiliza el mismo concepto de las aplicaciones mencionadas anteriormente de forma tal que para el desarrollador de software será transparente el uso del mismo, reduciendo de esta forma la resistencia al cambio.

### Acercamiento de nuevos clientes potenciales

La empresa Security Innovation recientemente adquirió a la empresa Safelight[[1]](#footnote-1), lo cual le permite aumentar la cartera de clientes y la diversificación de los productos. Este enfoque permite que clientes potenciales puedan evaluar los productos existentes y posteriormente evaluar y utilizar el complemento de seguridad a desarrollarse. De forma periódica nuevos clientes solicitan periodos de evaluación de los productos desarrollados en Security Innovation y muchos de ellos se convierten a su vez en nuevos clientes.

## Debilidades

### Poca o nula inserción en el campo del análisis estático de código

Pese a que la empresa Security Innovation ha sido una institución respetable que ofrece los mejores productos de la industria para mejorar la seguridad en las aplicaciones de software, ha tenido poca si no es que nula participación en el campo de el análisis estático de código. Debido a ello, se ha integrado con herramientas de terceros que son los que verdaderamente realizan el análisis del código fuente y las mejores prácticas de la empresa son luego despegadas. Esta dependencia en herramientas de terceros es importante pero a su vez es una limitación, puesto que no se pueden comercializar los productos de forma independiente. Debido a la poca participación en este mercado, existen otras empresas que han formado un sólido ecosistema y que de forma más madura ofrecen sus productos a los clientes en varias industrias.

### Dependencia de terceras empresas para realizar el análisis estático de código.

Existe una dependencia con otras empresas en el mercado para integrar productos en conjunto. Este enfoque es necesario pero no es suficiente. Cuando se trabaja en conjunto con otras organizaciones existen contratos de comercialización del producto que se deben respetar, al igual que el tema de licenciamiento. Esto obliga a que los clientes tengan dos productos instalados en lugar de tener uno solo. Muchas veces incluso relucen problemas de compatibilidad entre las partes que generan la tendencia a errores y al trabajo coordinado. De igual forma si las herramientas de terceros de las cuales se depende tienen un costo elevado, pequeñas y medianas empresas no podrían cubrir el precio de las respectivas licencias y por ende no pueden hacer uso de la integración.

### Proyectos de código abierto y gratuito ofrecen productos similares a muy bajo costo.

En algunos casos el factor económico es determinante para decidir si se adquiere o no un producto. Este proceso se convierte en un poco más complicado si existen herramientas de código abierto y gratuito, que si bien es cierto no se asemejan a los productos comerciales, tienen un funcionamiento un tanto parecido. Este factor genera que se prefiera implementar controles internos y políticas basados en estándares gratuitos . Las organizaciones que comercializan productos se ven confrontados cuando la calidad de los estándares y herramientas gratuitas es ampliamente reconocida.

### Plugin limitado a un lenguaje de programación y a un entorno integrado de desarrollo.

El prototipo funcional del componente o plugin a desarrollarse está destinado al ambiente de desarrollo Visual Studio.NET y al lenguaje de programación C# . Esto significa que se limita a estas dos tecnologías y no abarca otras plataformas ampliamente utilizadas como lo es Java por ejemplo. Incluso impide que se realice el análisis de código en otros lenguajes dentro de la plataforma de programación de Microsoft, como lo es el caso del lenguaje Visual Basic.NET.

## Amenazas

### La competencia ofrece productos similares e integrados

A pesar de los retos en el momento de implementar seguridad en las organizaciones, la necesidad de proteger los activos de la empresa es vital. Por esta razón, se cuenta hoy con variedad de productos similares que le facultan a la industria mejorar e incluso identificar defectos de seguridad en el software que se desarrolla. Muchas de estas organizaciones, con años de prevalencia en el mercado, crean productos innovadores con el respaldo de calidad del fabricante y que son atractivas por su grado de confianza.

Dentro de la lista de organizaciones líderes en el mercado de la seguridad de las aplicaciones figuran IBM con el producto denominado AppScan, Hewlett-Packard ofrece el producto Fortify Software Security Center, Checkmarx a su vez de forma emergente ha ganado territorio incursionando con la herramienta llamada Static Code Analysis y la empresa WhiteHat ofrece el producto denominado WhiteHat Sentinel.

El enfoque proporcionado por las herramientas mencionadas anteriormente, brindan la posibilidad de realizar un análisis del código fuente, con el objetivo de encontrar vulnerabilidades y generan de esta forma un reporte con los hallazgos, eventualmente el desarrollador analiza detalladamente el informe y procede con las correcciones dictadas por la herramienta.

Así mismo los clientes, usuarios de estas aplicaciones, se preocupan por implementar las herramientas que sean efectivas y que les brinden el grado de protección que ellos requieren. Por tal razón es normal que una empresa pueda escoger y evaluar más de un producto donde deberá decidir cuál le bridará una mejor opción.

### Clientes prefieren productos unificados

Siempre existe la opción de que un cliente elija otro producto que satisfaga las necesidades y que le permita cumplir a cabalidad con sus objetivos estratégicos de la organización. De igual forma buscará aquellas soluciones de software que puedan ser implementadas y adaptadas fácilmente. Una herramienta unificada es aquella que ofrece múltiples y comunes operaciones sin necesidad de que el usuario necesite adquirir componentes de terceros, lo cual facilita entre muchas otras cosas los costos de licenciamiento.

En seguridad de aplicaciones por ejemplo, herramientas como lo son HP Fortify son ampliamente utilizadas a nivel corporativo porque aparte del respaldo comercial que existe, las empresas encuentran en esta herramienta un producto unificado donde pueden hacer análisis periódico del código fuente, dentro del mismo entorno de desarrollo integrado, incluso permitiendo que aplicaciones en tecnologías no adyacentes puedan ser analizadas de forma transparente. El usuario a su vez puede decidir la periodicidad con la que el análisis es realizado, definir ciertas reglas intrínsecas de la organización, extender la funcionalidad de la herramienta y predefinir la forma en que los reportes son generados, todo esto desde la misma configuración de la herramienta.

De esta forma existe gran variedad de productos que tienen como objetivo brindar un ecosistema donde todas las tareas relacionadas con seguridad dentro del ciclo de vida del desarrollo del software estén disponibles ahí mismo, sin necesidad de adquirir otros productos.

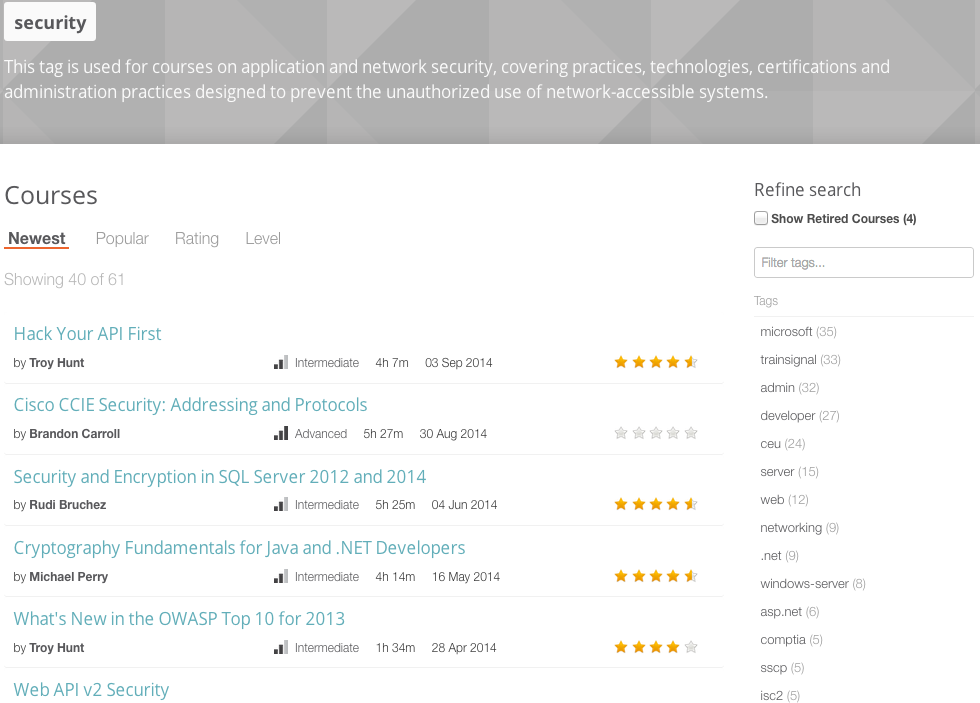
Cuando existe complejidad en el momento de implementar una solución de software en la organización, se pierde la funcionalidad de la misma, ya que se crea una atmósfera negativa donde se ve como un verdadero reto poder utilizarla de forma adecuada. Esto se evidencia cuando no existe un producto unificado y problemas de compatibilidad. Bajo ésta perspectiva es claro notar que una empresa tienda a preferir productos maduros especializados en brindar soluciones a aplicaciones de software vulnerables a ataques informáticos.

### Proliferación de herramientas en línea para capacitar personal.

Internet se ha convertido en una herramienta indispensable para la sociedad y aunado a ello la proliferación de dispositivos móviles genera que de forma indirecta el modelo tradicional de hacer negocios cambie. Por tal razón existen diversos mecanismos por medio de los cuales los desarrolladores de software se pueden capacitar en línea, desde la comodidad de las oficinas o desde sus hogares. También es importante observar que el precio por esas capacitaciones es relativamente bajo e incluso muchas organizaciones cubren estos entrenamientos a sus empleados. El hecho que exista la facilidad de realizar entrenamiento, especialmente en temas como desarrollo seguro de software, hace que las empresas prefieran invertir en adquirir el material de capacitación en lugar de adquirir productos de software por varias razones incluyendo el hecho de que el entrenamiento se puede realizar varias veces al año, en conjunto con el departamento de recursos humanos, se puede acceder al contenido de los cursos en cualquier momento, a medida que nuevos cursos están disponibles, los usuarios pueden tener acceso a ellos y principalmente el costo es relativamente bajo. Empresas como Pluralsight, contienen un catálogo bastante amplio en cursos enfocados a seguridad en el software, tal como se puede percibir en la imagen mostrada a continuación, donde existen alrededor de 60 cursos virtuales.

Imagen 1

Cursos virtuales enfocados en seguridad de la empresa Pluralsight.



Fuente : http://www.pluralsight.com/tag/security

# CAPÍTULO ll MARCO TEÓRICO

En el capítulo presente, se consignan una serie de términos y conceptos fundamentales para el prototipo propuesto, cuyo significado es relevante para comprender mejor aspectos técnicos y específicos al prototipo. Así mismo, este capítulo tiene como objetivo servir de referencia ante cualquier eventual término utilizado en éste documento, de forma tal que exista una definición clara y concisa que ayude a trazar una hoja de ruta y un panorama explicativo de los términos aquí utilizados.

## Sistemas de Información

Un sistema de información se compone de un conjunto de elementos, que tienen como objetivo común recolectar, almacenar procesar y analizar datos. Por medio de un sistema de información es que los datos almacenados y recopilados son debidamente procesados y convertidos en información, la cual es utilizada como base durante el proceso de toma de decisiones.

Los autores Kendall & Kendall (2011), refiriéndose a este tema indican que “Los sistemas de información se desarrollan para distintos fines, dependiendo de las necesidades de los humanos y la empresa.” (p. 2). En una sociedad globalizada como en la que se vive actualmente, los sistemas de información juegan un rol fundamental; tareas que en antaño se hacían de forma manual, con el advenimiento de los sistemas de información se han podido automatizar, permitiendo que exista un aprovechamiento consistente de recursos dentro de la organización.

Existen varios tipos de sistemas de información entre los que se destacan :

1. Sistemas de procesamiento de transacciones:

Dentro de esta categoría se agrupan los sistemas que realizan volúmenes inmensos de transacciones para la organización, como por ejemplo sistemas de procesamiento de datos de una entidad financiera. Los autores Kendall & Kendall (2011) sostienen que éstos sistemas “…eliminan el tedio de las operaciones transaccionales necesarias y reduce el tiempo que se requeriría para hacerlo de forma manual” (p. 2).

1. Sistemas de automatización de oficinas :

Tienen como objetivo servir de soporte a aquellas persas que analizan la información y hacen ciertas transformaciones antes de poder distribuirlos. Por ejemplo se podría mencionar el caso de una persona que aprueba el pago de horas extra, ya que él debe transformar y manipular los datos para que el pago sea efectivo o no.

1. Sistemas de información administrativa:

Bajo ésta categoría se citan los sistemas que sirven de soporte a los empleados de una organización con el objetivo de hacer el trabajo más eficiente. Kendall & Kendall (2011) cuando se refieren a este tipo de sistemas indican que “Los sistemas de información administrativa producen información que se utiliza en el proceso de toma de decisiones.” (p. 3).

1. Sistema de soporte de decisiones:

Son sistemas de información especializados cuyo objetivo es brindarle al usuario información de forma tal que él pueda tomar decisiones basado en la información disponible. Un aspecto importante a mencionar es que según los autores Kendall & Kendall (2011) estos sistemas “… está enfocado a brindar respaldo a la toma de decisiones en todas las fases, aunque el la decisión misma le corresponde de manera exclusiva al usuario”. (p. 3).

1. Inteligencia artificial y sistemas expertos:

Este tipo de sistemas utilizan las técnicas utilizadas por la ingeniería artificial y que tienen como función ayudar de forma sustancial a las empresas en sus negocios. Kendall & Kendall (2011) establecen que “Los sistemas expertos son una clase muy especial de sistemas de información que han demostrado su utilidad comercial gracias a la disponibilidad extendida de software y hardware.” (p. 2).

## Desarrollo de Sistemas

El proceso de desarrollo de sistemas, considerado como una forma de arte para algunos autores entre los que figuran Donald E. Knuth en su reconocida obra de cuatro volúmenes denominada The Art of Computer Programming, Addison-Wesley Professional 1997, contempla las diferentes etapas y metodologías necesarios para desarrollar una aplicación de software. Pese a que la ingeniería en computación es una rama de la ciencia relativamente nueva, el auge de sistemas de información en la industria y en la sociedad, amerita que exista un proceso unificado para el desarrollo de sistemas eficientes que no solamente cumpla con los objetivos para los cuales fue diseñado si no que también sea resistente a fallos y recuperable ante cualquier eventualidad.

En el desarrollo de sistemas existen varias metodologías, a veces llamadas también procesos por varios autores tal es el caso de Sommerville(2011) el cual argumenta que “Un proceso de software es una serie de actividades relacionadas que conduce a la elaboración de un producto de software.” (p. 28). El objetivo de la metodología es definir las actividades y la secuencia necesarias para la correcta implementación de sistemas. A continuación se muestran varios enfoques.

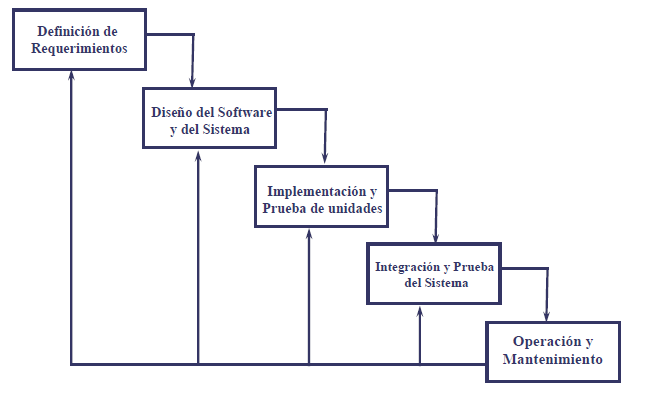
### Modelo en Cascada (waterfall)

Este modelo propone una planificación detallada de las diversas actividades del proceso antes de trabajar en ellas. De esta forma se identifican las actividades fundamentales del desarrollo de sistemas tales como desarrollo, validación de datos, recopilación de requerimientos, ejecución de pruebas y luego se representan como fases separadas. El autor Sommerville(2011) identifica cinco fases o etapas en el modelo en cascada, las cuales son:

1. Análisis y definición de requerimientos : Tomando en consideración a los usuarios del sistema, se definen las restricciones y los objetivos del sistema.
2. Diseño del sistema : Sommerville(2011) afirma que “El proceso de diseño de sistemas asigna los requerimientos, para sistemas de hardware o de software, al establecer una arquitectura global” (p. 31). En este proceso de diseño se busca crear una abstracción e identificación de los componentes del sistema y como interactúan entre ellos.
3. Implementación y prueba de unidad : En esta fase el diseño de software se implementa a manera de un conjunto de programas. Osherove(2009) cuando se refiere al concepto de pruebas de unidad afirma que “Una prueba de unidad es una pieza de código automatizada que invoca el método o clase que está siendo probada y luego valida algunas asunciones del comportamiento de la clase” (p. 11). Así mismo Sommerville (2011) supone que “La prueba de unidad consiste en verificar que cada unidad cumpla con su especificación.” (p. 31).
4. Integración y prueba de sistema : Durante la fase de integración los diferentes módulos o subprogramas se conciben como un único sistema y se realiza un proceso de pruebas completo a fin de determinar que se haya cumplido con los requerimientos iniciales.
5. Operación y Mantenimiento : En esta fase el sistema está listo para ser implementado y utilizado por los usuarios. Para Sommerville (2011) “…ésta es la fase más larga del ciclo de vida, donde el sistema se instala y se pone en práctica” (p. 31). Durante las etapas de mantenimiento se busca solventar errores presentados en el momento de utilizar el sistema y además se utiliza para agregar mejoras sustanciales y desarrollo de nuevos requerimientos.

Imagen 2

Modelo en cascada (waterfall)



Fuente : Ingeniería de Software, Novena Edición Pearson 2011

### Desarrollo Incremental

Este modelo aboga por realizar una implementación inicial, la cual es expuesta para que el usuario brinde retroalimentación y comentarios; una vez recibida esta retroalimentación, se desarrollan diferentes versiones del producto hasta que se consigue un producto adecuado. Según Sommerville(2011) “El desarrollo incremental refleja la forma en la que se resuelven los problemas”.(p. 33). Este mismo autor también indica que el desarrollo incremental es mejor que el modelo en cascada debido a que “… resulta más barato y fácil de realizar cambios en el software conforme éste se diseña.” (p. 33).

### Ingeniería de software orientada a reutilización

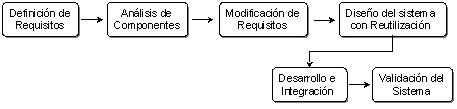
Este modelo de desarrollo de sistemas se asemeja bastante a otros modelos existentes, con la diferencia de que aboga por la reutilización de componentes, de forma que no exista duplicidad. Sommerville (2011) es claro cuando indica que:

La ingeniería de software orientada a la reutilización tiene la clara ventaja de reducir la cantidad de software a desarrollar y, por lo tanto, la de disminuir costos y riesgos; por lo general, también conduce a entregas más rápidas de software. Sin embargo, son inevitables los compromisos de requerimientos y esto conducirá hacia un sistema que no cubra las necesidades reales de los usuarios. (p. 36).

En la siguiente imagen se aprecia la ingeniería de software orientada a reutilización, nótese que existe una fase denominada diseño de sistemas con reutilización, donde se contempla poder reutilizar código, diseños, flujos de trabajo entre otros.

Imagen 3

Ingeniería de software orientada a la reutilización



Fuente : Ingeniería de Software, Novena Edición Pearson 2011

## Plugin

Un plugin es una extensión o complemento desarrollado para solventar una necesidad específica dentro de una tecnología determinada. Es generalmente utilizado en los entornos de desarrollo integrado como Eclipse o Visual Studio y además en navegadores o agentes de usuario más populares entre los que figuran Google Chrome y Mozilla Firefox. El objetivo de estos complementos es extender la funcionalidad o proveer más herramientas de las que están comúnmente disponibles a fin de realizar tareas específicas. En la actualidad existe gran diversidad de estas extensiones que permiten por ejemplo buscar y reemplazar texto, convertir texto de minúscula a mayúscula y viceversa, encontrar texto con contenido similar entre muchas otras cosas.

En la siguiente imagen se muestran algunos ejemplos de extensiones para el navegador Firefox.

Imagen 4

Extensiones disponibles para Firefox



Fuente: https://addons.mozilla.org/es/firefox/

## Prototipo

Sommerville (2011) define prototipo como “Una versión inicial de un sistema de software que se usa para demostrar conceptos, tratar opciones de diseño y encontrar más sobre el problema y posibles soluciones.” (p. 45). Partiendo de la definición anterior se observa que el prototipo es una herramienta de suma importancia pues permite tener una mejor percepción del problema a resolver y de esta forma encontrar soluciones a problemas de diseño e incluso encontrar en una etapa temprana alguna complicación en el futuro. Según afirma Somerville (2011) “Los prototipos no tienen que ser ejecutables para ser útiles” (p. 46).

## Tecnologías de información y comunicación

Son un conjunto de normas que agrupan elementos de infraestructura que permiten hacer más ágil y eficiente la comunicación entre los seres humanos, facultándolos para que puedan realizar sus actividades diarias de forma oportuna. Dentro de este modelo se sitúan las personas, los elementos de infraestructura tales como servidores, redes inalámbricas y telefonía entre otros elementos cuyo objetivo común es poner al alcance de los seres humanos mecanismos para que se comuniquen mejor y por ende mejorar las relaciones humanas.

## Software

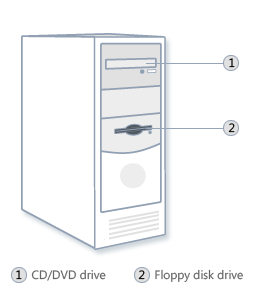
Es un término genérico que hace referencia a el conjunto de programas que forman parte de una computadora. Pese a ser un anglicismo, el diccionario de la Real Academia de la Lengua Española (2001) define el software como el “Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora”. Basado en esta definición, se puede interpretar que el software son aquellos componentes intangibles de una computadora. Para Tanenbaum (1998) “El programa de sistema más fundamental es el sistema operativo que controla todos los recursos de la computadora”. Partiendo de la definición que brinda el autor, se reconoce que el sistema operativo dentro de una computadora es el sistema más importante ya que sus tareas están relacionadas con el manejo de los recursos y con permitir que nuevas aplicaciones sean instaladas y ejecutadas.

## Hardware

Se entiende por hardware todos los elementos tecnológicos que pueden ser percibidos por medio de los sentidos, tal es el caso de los servidores físicos, computadoras portátiles, impresoras, teclados, monitores dispositivos de red entre otros. El hardware tiene un rol fundamental en la tecnología pues habilita al usuario a realizar mejor sus tareas.

Imagen 5

Ejemplo de hardware



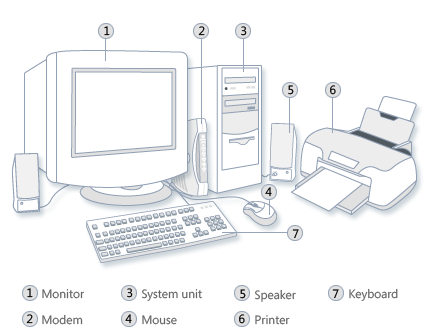
Fuente: Microsoft http://goo.gl/8BrfXK

## Computadora

Una computadora es una máquina que tiene como función básica capturar datos de entrada, procesarlos y emitir una salida. Stroustrup (2009) afirma que “Las computadoras son construidas por personas para que sean usadas por personas” (p. 21), luego indica que “Una computadora es una herramienta bastante genérica; puede ser usada para un rango inimaginable de tareas. Hace que un programa sea útil para alguien” (p. 21). En la imagen siguiente se puede apreciar una computadora ordinaria que fácilmente se encuentra en un hogar.

Imagen 6

Computadora ordinaria



Fuente: Microsoft http://goo.gl/8BrfXK

## Aplicaciones de software

Son programas informáticos desarrollados con el objetivo de ayudar a resolver un problema del negocio por medio de una computadora y que están desarrollados en una tecnología específica. Estas aplicaciones de software pueden ser desarrolladas a lo interno de las organizaciones o puede ser subcontratadas.

## Usuario malicioso

Es un usuario experto de la tecnología pero que busca la forma de darle un uso inapropiado a la misma, con el objetivo de descubrir alguna vulnerabilidad existente y comprometer la seguridad del sistema, alterar o robar información considerada sensitiva para la organización y comprometer a otros usuarios del sistema.

## Hacker

Según Erickson (2008) el término hacker se refiere a aquella persona que tiene un modo de pensar distinto al de la mayoría cuando indica que “La esencia de hacking consiste en la búsqueda de usos no previstos o pasados por alto por un usuario normal”. Tomando como base la definición que el autor brinda, se puede determinar que un hacker no es un criminal que comete delitos informáticos, si no es más bien una persona que tiene un modo de pensar distinto al convencional y que aprovecha esta capacidad para alcanzar grandes logros.

## Activo

Un activo son aquellos componentes que tienen un valor para la organización y que la pérdida del mismo o el deterioro causa una interrupción de las operaciones e impide que una organización pueda alcanzar sus metas. En una época como la actual la información de una empresa es quizá el activo más importante y por ende es necesario implementar controles para proteger el acceso no autorizado a la información . Según afirma Paul (2011) “Los activos pueden ser tangibles e intangibles en su naturaleza”, luego el mismo autor indica que “Los activos tangibles son aquellos que pueden ser percibidos por los sentidos”.(p. 16).

## Vulnerabilidad

Paul (2011) define el concepto de vulnerabilidad como “Una debilidad que puede ser ejecutada accidentalmente o explotada intencionalmente por un atacante, resultando en la brecha de la política de seguridad”. Como lo indica el autor, una vulnerabilidad está asociada al concepto de debilidad o falla que es causada de forma accidental o intencional. Cuando se habla de vulnerabilidades en el código de fuente se habla por lo tanto a fallas introducidas por los desarrolladores durante el proceso de desarrollo del software y que son aprovechadas por usuarios maliciosos para comprometer al sistema y a la organización.

## Ataque informático

El autor y experto en seguridad Paul(2011) define el concepto de ataque al mencionar que

Los agentes de riesgo pueden intencionalmente causar que un riesgo se materialice o que las amenazas puedan ocurrir como resultado de un error de usuario o que haya sido descubierto accidentalmente. Cuando los agente de amenaza de forma activa o intencional causan que un riesgo se materialice, se le conoce como un ataque. (p. 18).

En la definición anterior se aprecia como un ataque está relacionado a una actividad que se hace de forma intencional y que tendrá efectos adversos en las organizaciones son víctimas de éstos ataques.

## Probabilidad

En términos de riesgo, la probabilidad es la oportunidad de que una amenaza particular ocurra. Paul(2011) afirma que “La probabilidad se expresa como un percentil, no obstante algunas veces se utilizan técnicas meramente heurísticas por lo cual algunas organizaciones utilizan categorías como medio alto y bajo” (p. 19).

## Impacto

Es el resultado de una amenaza materializada, la cual puede tener efectos adversos para una empresa como lo son la interrupción temporal de los servicios, pérdidas económicas, robo de información sensitiva.

## Factor de exposición

Paul (2011) define el término de factor de exposición como “… la oportunidad de una amenaza de causar pérdidas” (p.18). De ésta forma este parámetro juega un papel muy importante en el momento de calificar el riesgo.

## Controles

Un control es un mecanismo para mitigar una amenaza, el cual puede ser técnico, administrativo o físico. En términos de seguridad de aplicaciones Paul(2011) indica que algunos ejemplos de controles lo constituyen “…validación de datos de entrada, control del código fuente, controles de acceso controlados y supervisados” (p. 19).

## Políticas de seguridad

Paul (2011) se refiere a una política de seguridad cuando expresa que “…Es el instrumento por medio del cual los activos digitales que requieren protección pueden ser identificados”. (p. 26). Según la definición anterior se observa que el objetivo de una política de seguridad es responder a la pregunta de qué es lo que se quiere proteger y las consecuencias de no protegerlos. Bajo esta perspectiva se puede concluir que identificando los activos más críticos de la organización es vital.

## Crimen cibernético

Se refiere a acciones poco éticas y criminales dirigidas hacia una entidad u organización a fin, con el objetivo de causar una interrupción de las operaciones, generalmente asociadas con pérdidas económicas, robo de información sensitiva y afectación de la imagen, sin dejar de lado las repercusiones legales en las cuales las empresas pueden verse inmersas luego de un ataque cibernético.

## Microsoft

Empresa norteamericana dedicada al desarrollo y comercialización de aplicaciones de software y sistemas operativos. La empresa fue fundada en 1985 por Bill Gates y Paul Allen. La empresa Microsoft desarrolla y mantiene la plataforma de desarrollo denominada Microsoft .NET, la cual permite que ingenieros en computación puedan crear software escalable.

## Hewlett Packard

Es una empresa estadounidense la cual también se conoce bajo el acrónimo de HP, fue fundada en 1939. La empresa ha sido un gigante en el mercado informático creando computadoras personales, servidores, impresoras y dispositivos móviles entre otros. HP también participa activamente en el mercado de seguridad de aplicaciones.

## HP Fortify

Es una herramienta que permite hacer análisis estático de código fuente, brindándole al usuario un reporte detallado de los problemas de seguridad que tiene la aplicación antes de que dicha aplicación sea publicada en un ambiente de producción. Una vez que se han direccionado las vulnerabilidades mostradas por la herramienta, se proceden con las distintas etapas del ciclo de vida del desarrollo de software. El objetivo principal de la herramienta es servir de ayuda a mitigar riesgos asociados al diseño y a la programación de la aplicación.

## IBM

IBM es el acrónimo de International Business Machines Corporación, una empresa norteamericana la cual provee productos de software y hardware para la industria informática. Fue fundada en el año 1911 y ha sido una empresa muy importante en la evolución de la computación moderna.

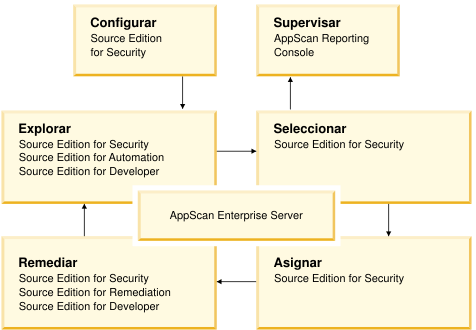
## IBM Security AppScan

Es una herramienta creada por la empresa IBM, según el sitio oficial [[2]](#footnote-2) del producto lo definen como “IBM Security AppScan mejora la seguridad de las aplicaciones Web y aplicaciones móviles, mejora los programas de seguridad y fortalece los cumplimientos de las normativas”.

La imagen siguiente muestra el flujo de trabajo que ejecuta la herramienta para determinar los problemas de seguridad en el código fuente.

Imagen 7

Flujo de trabajo de IBM AppScan



Fuente: <http://goo.gl/rJPlVj>

## Checkmarx

Es una empresa basada en Tel Aviv, Israel la cual se dedica a proveer herramientas para realizar análisis de código fuente a fin de encontrar vulnerabilidades y que éstas se corrijan en etapas tempranas. En términos de análisis estático de código, Checkmarx ofrece el producto denominado Checkmarx Suite el cual brinda extensiones para diversos entornos de desarrollo integrados.

## Instituto Ponemon

Es un instituto fundado en el año 2002 por el doctor Larry Ponemon y es considerado como una de las instituciones más importantes dedicado a temas como lo son privacidad, protección de datos y políticas de seguridad.

## Security Innovation

Es una empresa estadounidense enfocada en ayudarle a las organizaciones a construir y mantener software seguro, dicha empresa se dedica a la seguridad del software brindando una gama de productos en ésta materia.

## Entorno de desarrollo Integrado

Un entorno de desarrollo integrado o comúnmente denominado IDE por sus siglas en inglés, es una aplicación de software desarrollada para que ingenieros en sistemas puedan a su vez escribir aplicaciones de software. Como su nombre lo refleja, es integrado y los usuarios encuentran en la herramienta un sin fin de módulos pequeños que cumplen tareas particulares entre las que se identifican el poder contar las líneas de código, hacer reingeniería, buscar ocurrencias, dar formato al código fuente, agregar componentes de terceros entre otros.

## Visual Studio .Net

Visual Studio .Net es un entorno integrado de desarrollo creado por Microsoft por medio del cual los desarrolladores pueden escribir aplicaciones empresariales distribuidas y escalables bajo diversos lenguajes de programación. La versión más reciente de éste ambiente integrado es Visual Studio .Net 2014.

## .NET Framework

Thai y Lam (2003) cuando hacen referencia a la plataforma .NET Framework indican que:

.NET Framework es una plataforma de desarrollo que provee una nueva interface de programación de servicios Windows y que integra un número de tecnologías que emergieron de Microsoft durante la década de 1990. Microsoft anunció la iniciativa .NET en Julio de 2000. En Abril de 2003, la versión 1.1 integral de .NET Framework fue liberada. (p. 6).

Según los mismos autores, la plataforma .NET consta de cuatro grupos de productos separados entre los que se destacan :

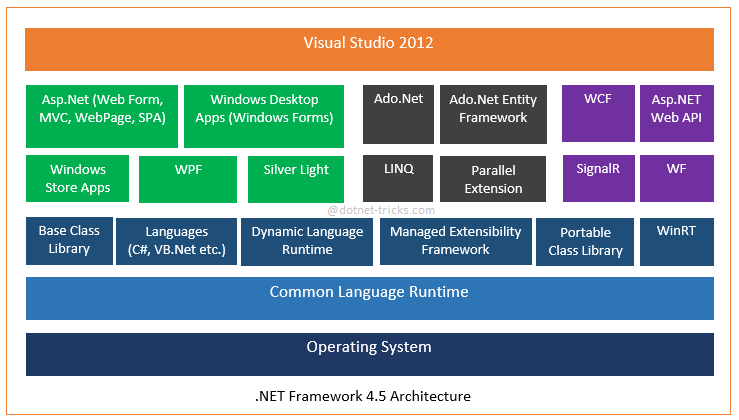
1. Herramientas de desarrollo y librerías.
2. Servicios Web.
3. Servidores especializados.
4. Dispositivos.

En vista de que es una plataforma de desarrollo, incluye herramientas predefinidas para tareas habituales y provee un modelo extensible para que la industria cree sus propios productos utilizando los fundamentos básicos previstos por la tecnología.

La imagen siguiente muestra una hoja de ruta de la plataforma Microsoft .NET 4.5.

Imagen 8

Arquitectura de Microsoft .NET 4.5



Fuente : DotNet-Tricks http://goo.gl/UwZi8J

### Lenguaje de programación C#

C# (léase C Sharp) es un lenguaje de programación de alto nivel, de propósito general y orientado a objetos que permite desarrollar múltiples tipos de aplicaciones entre las que se destacan aplicaciones de escritorio, aplicaciones Web, aplicaciones móviles entre otras. La sintaxis y semántica de este lenguaje de programación hace que sea similar a otros lenguajes de propósito general como lo es Java.

### Eclipse

Eclipse es un entorno integrado de desarrollo (IDE), el cual es soportado por la fundación Eclipse. Actualmente en el mercado se encuentran diferentes versiones de ésta herramienta entre las que figuran Luna, Kepler, Indigo, Helios, Galileo, Juno entre otras.

# Referencias Bibliográficas

File, G. (2011). *The SWOT Analysis Using Strengths to overcome Weakness Using Opportunities to overcome Threats.* Kindle Edition. Recuperado de Amazon.com

Hazzard, K., Bock, J. (2013). *Metaprogramming in .NET.* Shelter Island, New York: Manning Publications Co.

Ioannou, C. (2012). *SWOFT Analysis An easy to understand guide.* Kindle Edition. Recuperado de Amazon.com

Kendall, K., Kendall, J. (2011). *Análisis y Diseño de Sistemas.* Naucalpan de Juárez, Estado de México : Prentice Hall.

Lee, C. (2011). *How Do You Cite an E-Book (e.g., Kindle Book)?*. Recuperado de http://blog.apastyle.org/apastyle/2011/06/how-do-you-cite-an-e-book.html

Osherove, R. (2009). *The art of Unit Testing with Examples in .NET.* Greenwich, CT: Manning Publications Co.

Paul, M. (2011). *Official (ISC)2 Guide to the CSSLP*. New York, New York: CRC Press Taylor & Francis Group.

Sommerville, Ian. (2011). *Ingeniería de Software*. Naucalpan de Juárez, Estado de México: Addison Wesley.

Stroustrup, B. (2009). *Programming Principles and Practice Using C++.* Boston, MA : Addison Wesley.

Tanenbaum, A. (2010). *Computer Networks, Fift Edition.* Pretince Hall.

1. https://www.securityinnovation.com/company/news-and-events/press-releases/safelight-acquisition.html [↑](#footnote-ref-1)
2. http://www-03.ibm.com/software/products/en/appscan [↑](#footnote-ref-2)