

1-9-2023

TAREA 1: Marco teórico

Temas de la unidad 1. Información referente a la historia de la programación web.



Jorge Eduardo Escobar Bugarini - 21550317

TECNOLOGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CHIHUAHUA II

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

INGENIERÍA EN SISTEMAS COMPUTACIONALES

PW-10-ISC

PROGRAMACIÓN WEB - 10-11

EVALUACIÓN - 1

DOCENTE: LUIS ARMANDO ACOSTA RAMÍREZ

Índice

Objetivo	1
Introducción	2
1.1 Historia de las aplicaciones web	3
¿Qué es una aplicación web?	3
Historia de la web y su evolución	3
La historia de la web según su versión	7
Web 1.0 o la web de los documentos.....	7
Web 2.0 o la web de la gente	8
Web 3.0 o la web de la información	9
1.2 Arquitectura de las aplicaciones web.	11
Modelo de dos capas	13
Modelo de n-capas.....	13
1.3 Tecnologías para el desarrollo de aplicaciones web.....	15
Tecnologías Front-end / Del lado del cliente.....	15
Tecnologías Back-end / Del lado del servidor.....	16
Frameworks y librerías	17
Sistemas gestores de bases de datos	17
1.4 Planificación de aplicaciones web	18
Selección de Objetivos	18
Selección de usuarios	18
Expectativa de los usuarios.	19
Expectativa de la organización.	19
Fase de Planeación.	19
Fase de Selección de Software.	19
Fase de Selección de Hardware.....	20
Selección del Equipo Adecuado.....	20
1.5 Seguridad en las aplicaciones web	25
¿Qué es la seguridad de las aplicaciones web?	25
¿Cuáles son los mayores o más comunes riesgo en el desarrollo de aplicaciones web?	25
¿Cómo funciona la seguridad en las aplicaciones web?	28
¿Cuáles son las estrategias de seguridad importantes de las aplicaciones web?	29
¿Qué buenas prácticas de seguridad de la aplicación deben esperar las organizaciones de sus proveedores?	30
Conclusión.....	32

Fuentes bibliográficas	34
-------------------------------------	-----------

Índice de Figuras

Figura 1. Diagrama de la evolución de la web	7
Figura 2. Yahoo! Una de las primeras aplicaciones web con popularidad.	9
Figura 3. Arquitectura básica de un servidor web.	11
Figura 4. Modelo de dos capas	13
Figura 5. Modelo de n-capas	14

Objetivo

El propósito de este documento es proporcionar un marco teórico sólido y completo sobre las aplicaciones web, explorando su historia, arquitectura, tecnologías de desarrollo, planificación y seguridad. Se busca ofrecer una comprensión profunda de estos aspectos fundamentales para fomentar un mejor conocimiento de la importancia de las aplicaciones web en la sociedad actual y brindar orientación para aquellos que trabajan en su desarrollo, gestión o toma de decisiones relacionadas con ellas.

Introducción

En la era digital actual, las aplicaciones web desempeñan un papel central en nuestra vida diaria y en las operaciones de las organizaciones de todo el mundo. Estas aplicaciones, que van desde sitios web de noticias hasta complejas plataformas de comercio electrónico y redes sociales, han evolucionado significativamente con el tiempo. Para comprender plenamente su importancia y función, es imprescindible explorar algunos de los aspectos clave que configuran su estructura y desarrollo.

Este estudio profundiza en el apasionante mundo de las aplicaciones web, abordando una serie de cuestiones fundamentales que forman un sólido marco teórico. Exploraremos la historia de las aplicaciones web, desde sus humildes comienzos hasta su complejidad actual. También analizaremos la arquitectura subyacente que lo hace funcionar, las tecnologías que impulsan su desarrollo, la planificación necesaria para crearlo de manera eficiente y, por último, pero no menos importante, se debe garantizar la confidencialidad en un mundo digital cada vez más amenazado.

Este marco teórico servirá como base sólida para una comprensión más profunda de cómo han evolucionado las aplicaciones web, cómo funcionan y su importancia en la sociedad y en el mundo empresarial. Además, proporcionará una descripción general completa de los aspectos importantes que los desarrolladores y los tomadores de decisiones deben considerar al abordar proyectos relacionados con aplicaciones web. A través de este análisis detallado, esperamos arrojar luz sobre el impacto y la relevancia de las aplicaciones web en el mundo moderno y cómo podemos aprovechar al máximo su potencial en constante cambio.

1.1 Historia de las aplicaciones web

¿Qué es una aplicación web?

En la ingeniería de software se denomina aplicación web o software web a aquella herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los correos web, wikis, blogs, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos de aplicaciones web.

Historia de la web y su evolución

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía como interfaz de usuario que tenía que ser instalado por separado en cada computadora personal de cada usuario. El cliente realizaba peticiones a otro programa —el servidor— que le daba respuesta. Una mejora en el servidor, como parte de la aplicación, requería normalmente una mejora de los clientes instalados en cada computadora personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el

navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

Desde que Internet y la Web fueron creados, una cantidad infinita de posibilidades se han abierto, quizás, la más importante y usual sea el acceso de datos e información desde cualquier sitio. Las personas que se dedican al desarrollo de aplicaciones podrían considerar esto como un desafío, ya que los avances tecnológicos de estos últimos años exigen aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web. Sin lugar a dudas, Internet es una de las últimas tecnologías que más rápidamente se está desarrollando para su introducción en los hogares. Las previsiones apuntan a su integración como un electrodoméstico más, con unas capacidades y servicios que evolucionaran rápidamente. Por ese motivo, cada vez más adquieren importancia en las respuestas de los usuarios frente a la implantación de estas nuevas tecnologías.

Gracias al Internet y las conexiones de alta velocidad que lo acompañan, se ha podido mejorar de manera significativa la manera de trabajar de algunas personas al poder hacerlo desde sus hogares, Internet ha permitido a estas personas mayor flexibilidad en términos de horarios y de localización.

La programación web es un término adecuado para describir el proceso general que engloba el diseño y la creación de un sitio web.

Hace algunos años, los sitios web no eran mucho más que folletos digitales. Actualmente los sitios son más grandes y complejos.

Con la introducción de comercio electrónico y las páginas dinámicas, los sitios ya han dejado atrás los folletos y han pasado a ser auténticas aplicaciones de software.

En los años 60. En plena guerra fría, Estados Unidos crea una red exclusivamente militar, con el objetivo de que, en el hipotético caso de un ataque ruso, se pudiera tener acceso a la información militar desde cualquier punto del país.

Esta red se creó en 1969 y se llamó ARPANET. En principio, la red contaba con 4 ordenadores distribuidos entre distintas universidades del país. Dos años después, ya contaba con unos 40 ordenadores conectados. Tanto fue el crecimiento de la red que su sistema de comunicación se quedó obsoleto. Entonces dos investigadores crearon el Protocolo TCP/IP, que se convirtió en el estándar de comunicaciones dentro de las redes informáticas.

- ARPANET siguió creciendo y abriéndose al mundo, y cualquier persona con fines académicos o de investigación podía tener acceso a la red.
- Las funciones militares se desligaron de ARPANET y fueron a parar a MILNET, una nueva red creada por los Estados Unidos.
- La National Science Foundation crea su propia red informática llamada NSFNET, que más tarde absorbe a ARPANET, creando así una gran red con propósitos científicos y académicos.
- El desarrollo de las redes fue abismal, y se crean nuevas redes de libre acceso que más tarde se unen a NSFNET, formando el embrión de lo que hoy conocemos como INTERNET.
- En 1985 la Internet ya era una tecnología establecida, aunque conocida por unos pocos.
- El autor William Gibson hizo una revelación: el término «ciberespacio».

En ese tiempo la red era básicamente textual, así que el autor se basó en los videojuegos. Con el tiempo la palabra «ciberespacio» terminó por ser sinónimo de Internet.

En el Centro Europeo de Investigaciones Nucleares (CERN), Tim Berners Lee dirigía la búsqueda de un sistema de almacenamiento y recuperación de datos. Berners Lee retomó la idea de Ted Nelson (un proyecto llamado «Xanadú») de usar hipervínculos. Robert Caillau quien cooperó con el proyecto, cuenta que en 1990 deciden ponerle un nombre al sistema y lo llamaron World Wide Web (WWW) o telaraña mundial.

A partir de entonces Internet comenzó a crecer más rápido que otro medio de comunicación, convirtiéndose en lo que hoy todos conocemos.

Algunos de los servicios disponibles en Internet aparte de la WEB son el acceso remoto a otras máquinas (SSH y telnet), transferencia de archivos (FTP), correo electrónico (SMTP), conversaciones en línea (IMSN MESSENGER, ICQ, YIM, AOL, jabber), transmisión de archivos (P2P, P2M, descarga directa), etc.

La gran ventaja que presenta la WWW son los hiperenlaces gracias a los cuales la navegación y la búsqueda de información se convierten en un «juego de niños». Cuando se visualiza un documento WWW, el texto que aparece en la pantalla contiene palabras en otro color y para resaltar las palabras clave.

La publicación web o la programación web. - Son términos adecuados para describir el proceso general que engloba el diseño y la creación de un sitio web.

En un principio la web era sencillamente una colección de páginas estáticas, documentos, etc. para su consulta o descarga. El paso inmediatamente posterior en su evolución fue la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese carácter dinámico (es decir, generado a partir de los datos de la petición). Este método fue conocido como CGI («Common Gateway Interface») y definía un mecanismo mediante el que se podía pasar información entre el servidor y ciertos programas externos.

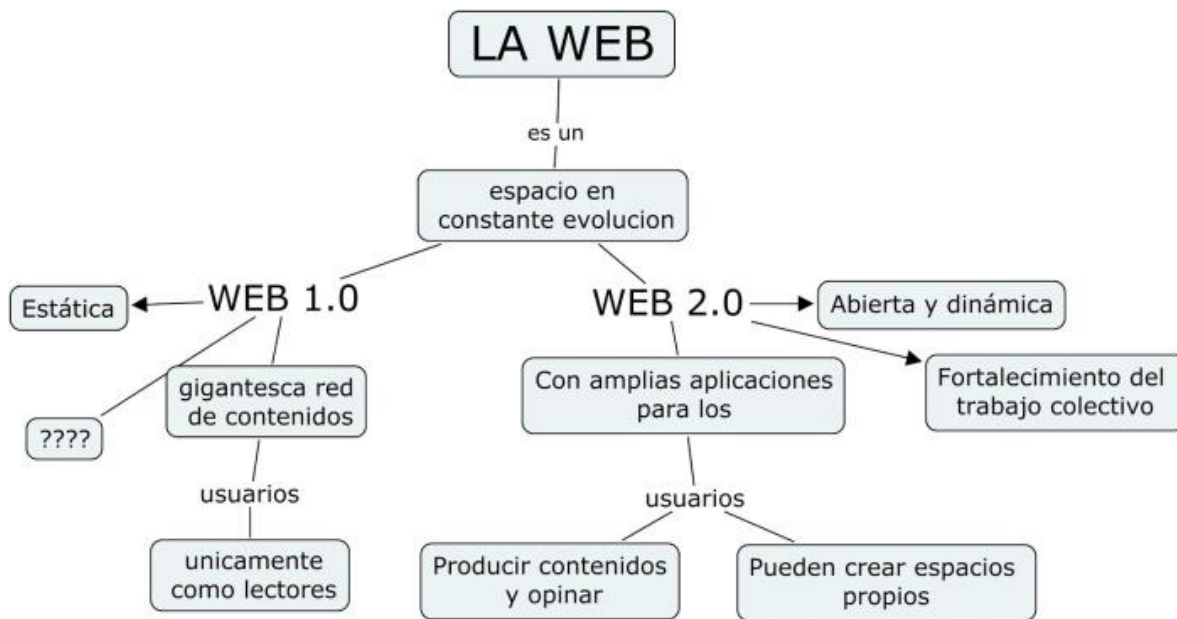


Figura 1. Diagrama de la evolución de la web

La historia de la web según su versión

Web 1.0 o la web de los documentos.

La primera versión de la Web 1.0 inventada por Tim Burners-Lee duró una década a partir de 1990. Esta era de la web se identificó como la "web de sólo lectura". El usuario sólo podía buscar y leer en cualquier contenido proporcionado por los productores de contenidos. La interacción con el usuario era prácticamente nula, ya que todos los sitios web eran estáticos o personales. Este era el requisito de los sitios web en aquella época. Los propietarios debían poner su información a disposición de cualquiera. No había comunicación activa ni intercambio de información entre el consumidor y el productor de la información.

Los principales protocolos web utilizados en la Web 1.0 son HTML, HTTP y URI. También se utilizaron algunos de los protocolos más recientes, como XML, XHTML y CSS. En la Web 1.0 se utilizaban scripts tanto del lado del cliente como del servidor. Tecnologías como ASP, PHP, JSP, CGI y PERL se utilizaban para las secuencias de comandos del lado del servidor, mientras que JavaScript, VBScript y Flash se empleaban para las secuencias de comandos del lado del cliente.

El principal inconveniente de la Web 1.0 era la falta de interacción entre el cliente y el productor de la información. No admitía la comunicación bidireccional y se basaba únicamente en la iniciación del cliente. Además, era lenta y poco ágil. Cada vez que se introducía nueva información en las páginas web, era necesario actualizarlas.

Una de las grandes ideas erróneas de la Web 1.0 era el desconocimiento de los efectos de red. La razón era que la proporción entre escritores y lectores era menor, lo que provocaba que la red fuera lenta. La lentitud de las redes impedía a algunos usuarios acceder a los recursos de la web. Estas deficiencias exigían una nueva versión de la web, que surgió como web 2.0.

Web 2.0 o la web de la gente

Dale Dougherty se refiere a la Web 2.0 como la Web social o la "Web de lectura y escritura". Facilita la interacción entre usuarios y sitios web, que no se limita a la lectura, sino que también permite escribir, modificar y actualizar contenidos. También se facilitó la comunicación entre usuarios, la colaboración y la participación en actividades distribuidas, colmando así las lagunas que se observaban en la Web 1.0. Esto dio lugar a que las transacciones web fueran bidireccionales. La Web 2.0 era mucho más que una nueva versión de la Web 1.0. Tenía un diseño web flexible, reutilización creativa de contenidos y una mayor flexibilidad. Contaba con un diseño web flexible, reutilización creativa, actualizaciones, creación de contenidos y modificación a través de la colaboración.

Figura 2. Yahoo! Una de las primeras aplicaciones web con popularidad.

Estos cambios supusieron un cambio bastante drástico en el panorama de la web. Por ejemplo, surgieron sitios web como YouTube o MySpace, que dependen totalmente de las aportaciones de usuarios de todo el mundo, y otros como Facebook o Twitter, que se basan tanto en las interacciones como en las contribuciones de los usuarios.



Existe la idea de que, si un sitio web se desarrolla con una tecnología determinada, pertenece a la web 2.0. Pero no es así. Pero no es así. El concepto principal de la Web 2.0 es simplemente dar a los usuarios la opción de aportar contenidos e interactuar. Las tecnologías web utilizadas con frecuencia incluyen HTML5, CSS3 y varios marcos de JavaScript como ReactJs, AngularJs, VueJs, etc.

Con estos rápidos avances en la Web, para satisfacer las necesidades de los usuarios en general había limitaciones visibles y potencial para más mejoras. Las cuestiones éticas siempre han sido una preocupación prioritaria. La conectividad y el intercambio de conocimientos entre plataformas seguían siendo muy limitados. Estas preocupaciones dieron lugar a la Web 3.0.

Web 3.0 o la web de la información

La Web 3.0 fue ideada por John Markoff, del New York Times, en 2006. También se la identifica como la "web ejecutable". Siempre ha sido difícil dar una respuesta o definición a la pregunta "¿qué es la web 3.0?", ya que los expertos tienen varias opiniones y enfoques al respecto. La mayoría de los expertos en TI consideran que

la web 3.0 es la web semántica. El concepto de propiedad de los datos ha desaparecido y se ha inclinado hacia la compartición de datos.

Los servicios pueden mostrar distintas vistas de los mismos datos para distintos usuarios, centrándose en contextos personalizados. Estos servicios pueden variar desde navegadores a mundos virtuales, pasando por diversos dispositivos. El objetivo de la web 3.0 es definir datos estructurados para vincularlos e integrarlos de una manera más eficaz que permita su reutilización en distintas aplicaciones, la gestión de datos, el apoyo a la accesibilidad móvil a internet, permitir la colaboración organizada y, lo que es más importante, mejorar la satisfacción del cliente.

Las redes sociales han sido una parte esencial de la sociedad y desempeñan un papel fundamental en la comunidad como plataforma para compartir pensamientos e ideas en lugar de utilizar la web únicamente para enlazar documentos. La web social conecta a personas de todo el mundo, haciendo el mundo más pequeño. La web semántica es una extensión de la web 3.0 que ofrece a la gente la posibilidad de encontrar información a un nivel mucho más profundo en términos de búsqueda y contexto.

La estructura de la información ha cambiado para garantizar la producción de contenidos legibles tanto por humanos como por máquinas. La web 3D permite a las personas vivir en un mundo virtual en el que la interacción alcanza su máximo nivel, ya sea en grupo o individualmente. Según el enfoque de la web centrada en los medios, los motores de búsqueda pueden tomar entradas de medios como audio, vídeo e imágenes y buscar resultados similares.

Algunos de los elementos clave de la web 3.0 son la web social, la web semántica, la web 3D y la web centrada en los medios de comunicación.

1.2 Arquitectura de las aplicaciones web.

Una aplicación web es proporcionada por un servidor web y utilizada por usuarios que se conectan desde cualquier punto vía clientes web (browsers o navegadores).

La arquitectura de un sitio web tiene tres componentes principales:

- Un servidor Web
- Una conexión de red
- Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a

través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.

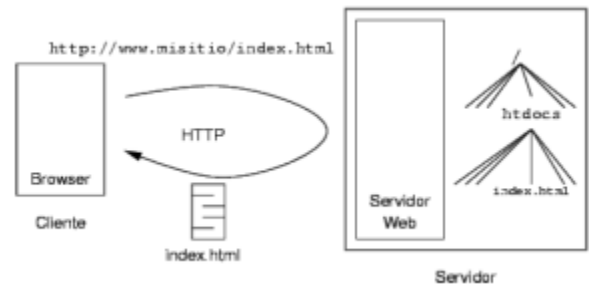


Figura 3. Arquitectura básica de un servidor web.

Las aplicaciones Web están basadas en el modelo Cliente/Servidor que gestionan servidores web, y que utilizan como interfaz páginas web.

Las páginas Web son el componente principal de una aplicación o sitio web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores web. En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor web se rompe, es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente). Cuando el browser ejecuta un script en el cliente, éste no tiene

acceso directo a los recursos del servidor. Hay otros componentes que no son scripts, como los applets (una aplicación especial que se ejecuta dentro de un navegador) o los componentes ActiveX. Los scripts del cliente son por lo general código JavaScript o VBScript, mezclados con código HTML.

La colección de páginas es en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario. El acceso a las páginas está agrupado también en el tiempo (sesión). Los componentes de una aplicación Web son:

1. Lógica de negocio.

- Parte más importante de la aplicación.
- Define los procesos que involucran a la aplicación.
- Conjunto de operaciones requeridas para proveer el servicio.

2. Administración de los datos.

- Manipulación de BD y archivos.

3. Interfaz

- Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
- Funcionalidad accesible a través del navegador.
- Limitada y dirigida por la aplicación.

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas. Una capa representa un elemento que procesa o trata información. Los tipos son:

- Modelo de dos capas: La información atraviesa dos capas entre la interfaz y la administración de los datos.
- Modelo de n-capas: La información atraviesa varias capas, el más habitual es el modelo de tres capas.

Modelo de dos capas

Gran parte de la aplicación corre en el lado del cliente (fat client).

Las capas son:

- Cliente (fat client): La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.
- Servidor: Administra los datos.

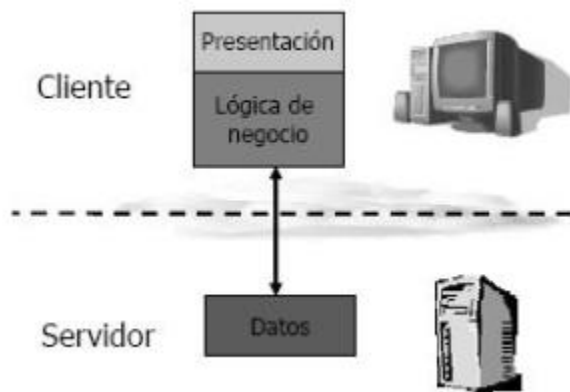


Figura 4. Modelo de dos capas

Las limitaciones de este modelo son.

- Es difícilmente escalable
- Número de conexiones reducida
- Alta carga de la red.
- La flexibilidad es restringida
- La funcionalidad es limitada.

Modelo de n-capas

Está diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de dos capas, introduce una capa intermedia (la capa de proceso) Entre presentación y los datos, los procesos pueden ser manejados de forma separada a la interfaz de usuario y a los datos, esta capa intermedia centraliza la lógica de negocio, haciendo la administración más sencilla, los datos se pueden integrar de múltiples fuentes, las aplicaciones web actuales se ajustan a este modelo.

Las capas de este modelo son:

1. Capa de presentación (parte en el cliente y parte en el servidor)
 - Recoge la información del usuario y la envía al servidor (cliente)
 - Manda información a la capa de proceso para su procesamiento
 - Recibe los resultados de la capa de proceso
 - Generan la presentación

- Visualizan la presentación al usuario (cliente)
2. Capa de proceso (servidor web)
- Recibe la entrada de datos de la capa de presentación
 - Interactúa con la capa de datos para realizar operaciones
 - Manda los resultados procesados a la capa de presentación
3. Capa de datos (servidor de datos)
- Almacenar los datos
 - Recuperar datos
 - Mantener los datos
 - Asegurar la integridad de los datos

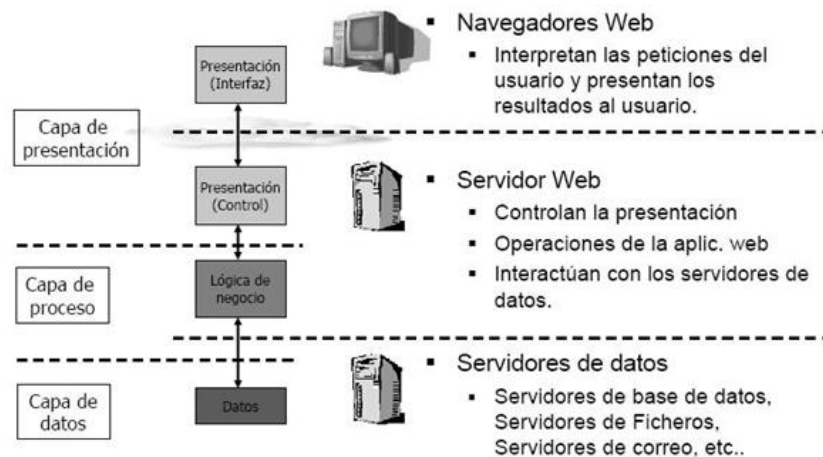


Figura 5. Modelo de n-capas

1.3 Tecnologías para el desarrollo de aplicaciones web

Hay numerosas tecnologías para desarrollo web y, a veces, es complicado para las empresas del sector elegir la mejor opción para sus objetivos. Esto se debe a que el desarrollo web tiene diferentes formas, desde páginas web convencionales hasta CRM, paneles de administración y softwares más complejos.

Tecnologías Front-end / Del lado del cliente

En esta tipología de tecnologías frontend para desarrollo web se enmarcan aquellas que se ocupan de la creación de interfaces de usuario y de establecer comunicaciones con el servidor.

Entre ellas podemos encontrar las siguientes:

JavaScript

Es una de las tecnologías web más usadas para las tecnologías frontend. Con este lenguaje de programación multiplataforma, es posible dar una mayor interactividad y dinamismo a los sitios web. Entre sus distintas funcionalidades, JavaScript permite crear tanto animaciones como objetos, localizar errores en formularios, cambiar elementos web de manera intuitiva, crear cookies...

Además, esta tecnología web también permite desarrollar aplicaciones tan potentes como Facebook o Twitter. Por esto, JavaScript es una de las mejores tecnologías para desarrollo web de la actualidad.

HTML

HTML es una de las tecnologías web, que funciona como un lenguaje de marcado usado para la creación de sitios web. Consiste en un conjunto de códigos cortos, que se clasifican como archivos de textos en las etiquetas. Dicho de otra manera, el texto se guarda en un archivo llamado HTML que se puede encontrar a través de los buscadores. Cada una de las etiquetas generadas tiene diferentes funciones.

De forma breve, HTML sirve para describir el contenido de un sitio web, como la información estructurada de párrafos, imágenes, etc. Por eso, HTML es una de las habilidades indispensables de un desarrollador web.

CSS

CSS es una de las mejores tecnologías para desarrollo web y a la que más recurren los programadores gráficos a la hora de desarrollar sus proyectos. Es un lenguaje funciona como las tecnologías frontend que sirve, sobre todo, para indicar la representación visual de las estructuras HTML. De esta manera, el lenguaje CSS sirve para acotar y trazar el aspecto visual de las etiquetas generadas por el HTML.

Tecnologías Back-end / Del lado del servidor

Las tecnologías back-end son las que se ocupan de la implementación de comportamientos de la web en el servidor. Entre ellas, algunas de las tecnologías para desarrollo web más populares son:

- **PHP:** Una de las mejores tecnologías al ser de uso general y adaptarse al desarrollo web. Se utiliza para la comunicación de un sitio web con un servidor de datos. Gracias a esto, es posible crear contenido dinámico con bases de datos como MySQL y con HTML. Este lenguaje también permite recoger datos de formularios, modificar bases de datos o administrar archivos en el servidor.
- **Python:** El factor más diferenciador de esta tecnología es que además de poseer una gran cantidad de librerías, es completamente gratis. Es una tecnología sencilla pero potente, al usar menos líneas de código que cualquier otro lenguaje de programación. Esto la convierte en una de las tecnologías más usadas.
- **Java EE: Java Enterprise Edition,** se utiliza para el desarrollo y ejecución de software de Java en el lado del servidor para plataformas empresariales de gran nivel.

Frameworks y librerías

Son herramientas y librerías de código ya escrito utilizadas por los/as desarrolladores/as, ya que les evita ejecutar sus tareas rutinarias desde cero. En este caso, los frameworks se utilizan para programar un lenguaje y las librerías para hacerlo más legible.

Destacamos las siguientes tecnologías para este apartado:

- **Angular:** Es uno de los frameworks más utilizados. Usa HTML para realizar UI y es compatible con las últimas versiones de Chrome, Firefox, Edge, iOS y Android.
- **React:** Esta biblioteca destaca por escribirse en código abierto en JavaScript y se utiliza principalmente para construir las interfaces de usuario, admitiendo diferentes idiomas y tecnologías.
- **Vue.JS:** Al igual que React, esta librería está escrita en código abierto. Es un framework escrito con JavaScript. Su característica principal es que se puede adaptar a cualquier proyecto.
- **NodeJS:** Es la tecnología de JavaScript que ha evolucionado para desarrollar toda la parte del servidor o back-end.
- **Laravel:** Tecnología de código abierto, se utiliza para desarrollar de forma notable y simple aplicaciones y servicios web usando PHP.

Sistemas gestores de bases de datos

Son un conjunto de datos que pertenecen a un mismo entorno y son recogidos regularmente para ser utilizados en cualquier momento. Para ello, existen una serie de tecnologías que se utilizan para almacenar, modificar y extraer información de estos datos recopilados.

- **MySQL:** Es una de las bases de datos más utilizadas por los programadores de todo el mundo. Almacenan información y trabajan con tablas que crean nuevas bases de datos para, así, poder hacer cualquier consulta.

- **Mongo BD:** Base de datos que permite un desarrollo escalable, rápido e interactivo. Utiliza código abierto enfocado a documentos.

1.4 Planificación de aplicaciones web

A la hora del desarrollo de un sitio Web se debe comenzar por dar respuesta a un grupo muy concreto de interrogantes, que permitirán aclarar y definir de manera básica las expectativas existentes en torno a la aplicación Web e Interacción que se va a generar.

Dichas preguntas podrían ser las siguientes:

- ¿se necesita realmente el sitio Web?
- ¿para qué se necesita el sitio Web?
- ¿Qué es lo que buscarán los usuarios en el sitio Web?
- ¿Qué imagen se quiere proyectar con el sitio Web a los usuarios?
- ¿Qué servicios interactivos se integrarán a través del sitio Web?
- ¿con qué recursos se cuenta para el desarrollo del sitio Web?
- ¿Cuánto tiempo debe ser invertido en el desarrollo del sitio Web?
- ¿Qué impacto causará el sitio Web en los usuarios?

Selección de Objetivos

Todo proyecto debe contar con objetivos definidos, contra los cuales puede medirse la efectividad de las acciones que se desarrollen. Green, Chilcote y Flick (2003) denominan esta etapa, resumen de objetivos, donde establecen tres puntos primordiales para esta selección, los cuales son definir los usuarios, lo que los usuarios esperan de la visita, y qué se pretende comunicar a través del sitio Web.

También podemos denominarlos: selección de usuarios, expectativas de usuario y expectativas de la organización

Selección de usuarios

Los usuarios o la audiencia se refieren a aquella parte de la población a la que se

pretende alcanzar o influir con el sitio Web; esto está estrechamente vinculado con la naturaleza del sitio que se propone. Toda la información que se obtenga servirá para establecer parte de IOS criterios de diseño de contenido, lenguajes, estética, entre otros.

"El modelado del usuario se basa en definir clases o perfiles de los usuarios y así poder diseñar con el objetivo de satisfacer las necesidades propias de cada grupo de usuarios" (Montero, Y. y Martón, F., 2003).

Expectativa de los usuarios.

La organización puede o no, ser conocida por IOS usuarios en potencia del sitio. Al plantearse la puesta en marcha de un sitio Web es necesario conocer hacia quién va dirigido éste y qué podrían esperar dichos usuarios al visitarlo.

Expectativa de la organización.

Es vital establecer o fijar qué resultados pretenden conseguir la organización con la creación del sitio Web. Con ello se busca definir el propósito del sitio Web, y se deben incluir aspectos que van desde los más generales, relacionados con el público de

destino, hasta IOS más específicos relacionados con la organización.

Morrin y Dickson (2001) indican que se debe decidir si lo que se busca con el sitio Web es: aumentar ventas, ofrecer servicios especializados y conocimientos, publicidad y patrocinio, incentivos (productos gratis para promoción), o para publicitar.

Fase de Planeación.

Todo proyecto posee una fase de planificación. En ella se deben definir cuáles son los requerimientos técnicos para este fin, quiénes serán los miembros del equipo, seleccionar la estructura, planear un sitio bien organizado, y realizar estudios de mercadeo comparativo.

Fase de Selección de Software.

En esta etapa de la metodología se busca determinar los requerimientos en cuanto al software que se utilizará para el desarrollo de la aplicación; es decir, sistemas

operativos, servidor (software), editores para diseño Web, lenguajes de programación, animaciones y otros componentes.

Esta definición dependerá de muchos aspectos como la existencia de licencias de software o los recursos técnicos y hardware

Fase de Selección de Hardware.

Antes de emprender el desarrollo de una aplicación Web, se deben determinar IOS requerimientos de hardware de la aplicación, el equipo servidor, la capacidad de respuesta, terminales, requerimientos de red si la aplicación ha de disponerse de una intranet o extranet.

En el caso que la aplicación vaya a estar publicada en Internet, se debe reservar el dominio y ubicar el servicio de hospedaje de empresas que se especializan en este servicio (Morrison y Dickinson).

Selección del Equipo Adecuado

"El entorno de trabajo en equipos surge de la necesidad de segmentar el proceso de producción en varias funciones diferentes" (Green, Chilcot y Flick, 2003).

Parte de la planificación de un sitio Web consiste en pensar en términos de equipo de trabajo.

Se trata del conjunto de técnicos que estarán a cargo de la concepción, construcción, evaluación y mantenimiento del sitio.

Los miembros del equipo Web serán IOS responsables de todas aquellas estructuras no visibles que hacen a la operatividad de un sitio.

El equipo se basará en la convergencia de saberes y experiencias. Debe estar integrado por técnicos de muy diversas disciplinas, capaces de contemplar IOS distintos aspectos que están en juego.

Inicialmente, la mayoría de las organizaciones tenían a cargo del sitio web un Web master que trabajaba solo, preparaba el servidor, editaba las páginas, hacía el diseño gráfico, y escribía o reunía los textos. Con el tiempo, se ha comprendido la

complejidad de estos proyectos. Es muy importante que el equipo tenga un líder (que puede en un futuro ser el Web master O no) porque formar un equipo Web requiere un gran compromiso personal e institucional.

Los equipos no se construyen de la noche a la mañana, ni a partir de una decisión administrativa O un mandato. Requieren un proceso de consolidación, que pasa por tiempos de búsqueda, acuerdos y adecuación.

Un equipo Web debe estar en constante comunicación tanto con aquellos que le han asignado su misión, como con sus destinatarios (clientes o usuarios). Su éxito se basa, además, en una estrecha colaboración y comunicación entre sus miembros.

Cada miembro del equipo Web debe estar capacitado en su campo de trabajo (la presencia o el nivel de experiencia de ciertos miembros del equipo puede variar dependiendo del proyecto).

Se sugiere una lista de miembros que podrían ser parte de este equipo, de esta lista se extrajeron los siguientes:

Jefe de proyecto: Encargado de la supervisión y control del resto del equipo. Actúa como enlace entre el usuario y IOS demás miembros.

Productor: Este se encarga de establecer el aspecto, percepción y funcionalidad del sitio.

Editor: Ayuda a establecer y mantener el estilo, además de la integridad del sitio. Tiene la responsabilidad de controlar la corrección y el flujo de texto y en algunos casos el contenido multimedia del sitio.

Costos de inversión

A la hora de planificar un sitio Web, se debe realizar un estudio de viabilidad que refleje el costo financiero a invertir, y que abarque tiempo, personal, hospedaje Web, dominio, licencias en caso de necesitarse, y mantenimiento.

Contar con un método de control de tiempo se hace pertinente en el desarrollo Web al igual que en otros proyectos, ya que el tiempo implica dinero. Una técnica para el

control del costo de cierta actividad, es el sistema de agenda, que consiste en dividir cada hora en segmentos de seis minutos llamadas unidades, y es a partir de la cantidad de unidades por actividad que se medirá el costo de ésta.

Beneficios a obtener

La puesta en marcha del sitio no lo es todo, la planificación del desarrollo de un sitio Web debe abarcar aspectos como beneficios a adquirir con el funcionamiento del mismo. Antes de establecer el desarrollo de un sitio Web se debe estimar el tiempo de retorno de la inversión, ya que, de esa forma tangible, como por ejemplo un retorno monetario, o intangible como la mejora de las relaciones interpersonales de la organización.

Fase de Contenido

Básicamente el éxito de un sitio Web se debe a su contenido. El contenido del sitio será probablemente una combinación de la información que actualmente se posee y de la que se tendrá que crear.

Es conveniente presentar un contenido acorde a los objetivos planteados para desarrollar el sitio, y que la información a plasmar cubra las necesidades e intereses del público a quien va dirigido.

Fase de Diseño

El diseño Web supone un trabajo extenso un trabajo extenso y detallado, puesto que abarca no sólo la interacción de múltiples elementos, como tecnología multimedia (audio, sonido, imágenes, animaciones, entre otros); sino, que abarca también su integración con una estructura lógica basada en el propósito del sitio. Representando una labor que va más allá de la simple necesidad e inquietudes de sus potenciales usuarios.

Al momento de diseñar sitios Web, es necesario tener en cuenta aspectos como la usabilidad y accesibilidad.

Usabilidad

Referido al tiempo, esfuerzo y capacidad requerido por el visitante, para que alcance un determinado nivel de adaptación al sistema, esta puede estar vinculada en ocasiones al número de pasos necesarios para lograr determinada actividad, y al nivel de conocimientos que requieran los usuarios para el uso de la aplicación.

Accesibilidad

Un sitio Web se desarrolla con la idea de que sea visto por el mayor número de visitantes, pero hay que tener presente que cada uno de estos puede acceder a la información con equipos, navegadores e idiomas distintos, estos factores pueden impedir que el sitio no pueda ser apreciado en su totalidad.

Romero (1999) señala tres áreas fundamentales para determinar la accesibilidad de un sitio Web, estas son:

Accesibilidad a la computadora: Referida a la capacidad de hardware con la que se debe contar para el uso de la aplicación Web, lo ideal en esta área es que la aplicación

se adapte a la computadora cualquiera que este sea.

Accesibilidad del navegador utilizado: En ocasiones mismos contenidos se visualizan de

distintas maneras en diferentes navegadores Web, se recomienda utilizar un diseño que se pueda visualizar en la mayoría de los navegadores y apegado al estándar.

Accesibilidad del diseño de las páginas Web: Los usuarios pueden tener desactivadas algunas funcionalidades de los navegadores Web, lo que podría incidir en la correcta visualización del contenido del sitio Web.

Fase de programación

Esta etapa corresponde a la utilización de las herramientas de programación Web. En este punto es primordial seleccionar los lenguajes de programación en los que

se desarrollará el sitio Web, la base de datos definir qué contenido del sitio será estático y cuál será dinámico.

Una vez definidos estos criterios se pasa a una etapa denominada Arquitectura de tres fases, la cual expone lo siguiente:

Bases de datos: Se diseña y crea la base de datos.

Programación intermediaria: Programas o códigos que se ejecutarán en el servidor Web. Acá se establecerá la comunicación entre la base de datos y la interfaz.

Interfaz: Programas y códigos que desplegarán el contenido a los usuarios a través del navegador Web. Se refiere aplicaciones que el usuario visualizará y a través de la cual operará con ella.

Fase de Testeo

Se ejecutan pruebas exhaustivas para asegurar el perfecto funcionamiento del sitio Web con usuarios que vayan a utilizar el sitio:

Comprobación en navegadores: El primer paso y también el más importante, es comprobar que las páginas se pueden visualizar de forma correcta en diferentes navegadores.

Detectar los vínculos rotos: Es necesario detectar documentos que existen en el sitio pero que no están conectados a través de vínculos; ellos deben ser reparados o eliminados en caso de no ser necesarios puesto que ocupan espacio en el servidor y tienden a crear confusión.

Comprobar tiempo de descarga: Uno de los puntos clave en el éxito de un sitio Web es su rapidez de descarga, en base a ello se recomienda un tiempo mínimo descarga.

En este punto los administradores deben recibir la capacitación para el desarrollo de sus tareas.

Fase de Mercado y Publicidad.

En esta fase se debe tomar en cuenta los Derechos Internacionales del Autor, por el cual:

Todas las copias realizadas incluyen derecho de autor.

Se anexa en el sitio los créditos de los desarrolladores, o un enlace que le permita a los usuarios contactarlos. Luego se define el dominio del sitio, se verifica que esté disponible para poder registrarlo, y se transfieren los archivos al servidor remoto por FTP, o desde la página del servidor. Se debe verificar que todos los archivos estén en su carpeta correspondiente de la Web.

Después de publicada la Web, su difusión depende de la publicidad que se le pueda dar mediante papelería de la organización, mediante banners en sitios específicos, por medio de buscadores y otros

1.5 Seguridad en las aplicaciones web

¿Qué es la seguridad de las aplicaciones web?

La seguridad de las aplicaciones web es la práctica de proteger los sitios web, las aplicaciones y las API contra los ataques. Es una disciplina amplia, pero sus objetivos en última instancia son mantener el buen funcionamiento de las aplicaciones web y proteger a las empresas del vandalismo cibernético, el robo de datos, la competencia poco ética y otras consecuencias negativas.

La naturaleza global de Internet expone las aplicaciones web y las API a ataques desde muchas ubicaciones y de varios niveles de escala y complejidad. Como tal, la seguridad de las aplicaciones web abarca una diversidad de estrategias y cubre muchas partes de la cadena de suministro del software.

¿Cuáles son los mayores o más comunes riesgo en el desarrollo de aplicaciones web?

Las aplicaciones Web pueden enfrentar varios tipos de ataque, dependiendo de los objetivos del atacante, la naturaleza del trabajo de la organización objetivo y las brechas de seguridad particulares de la aplicación. Los tipos de ataque más comunes incluyen:

- Vulnerabilidades de día cero: son vulnerabilidades desconocidas para los creadores de una aplicación y que, por lo tanto, no tienen una solución disponible. Ahora, atendemos a más de 20.000 cada año. Los ataques buscan explotar estas vulnerabilidades rápidamente y, a menudo, intentan evadir las protecciones establecidas por los proveedores de seguridad.
- Scripting entre sitios (XSS): XSS es una vulnerabilidad que permite que un atacante inyecte scripts del lado del cliente en una página web con el objetivo de acceder a información importante, hacerse pasar por el usuario o engañar al usuario para que revele información importante.
- Inyección de código SQL (SQi): SQi es un método en el cual un atacante se aprovecha de las vulnerabilidades por el modo en el que una base de datos ejecuta consultas de búsqueda. Los atacantes utilizan SQi con el objetivo de conseguir acceso a información no autorizada, modificar o crear nuevos permisos de usuario, o manipular o destruir datos confidenciales.
- Ataques de denegación de servicio (DoS) y ataques de denegación de servicio distribuido (DDoS): mediante una variedad de vectores, los atacantes pueden sobrecargar un servidor determinado o su infraestructura circundante con diferentes tipos de tráfico de ataque. Cuando un servidor ya no es capaz de procesar de manera efectiva las solicitudes entrantes, empieza a funcionar más lento y acaba por negar las solicitudes entrantes de usuarios legítimos.
- Corrupción de memoria: la corrupción de memoria se produce cuando se modifica de forma involuntaria una ubicación en la memoria, que puede acabar causando un comportamiento inesperado en el software. Los agentes maliciosos intentarán detectar y aprovecharse de la corrupción de memoria mediante inyecciones de código o ataques de desbordamiento de búfer.
- Desbordamiento de búfer: El desbordamiento de búfer es una anomalía que se produce cuando el software introduce datos en un espacio definido en la

memoria que se conoce como búfer. Desbordar la capacidad del búfer provoca que las ubicaciones de memoria adyacentes se sobrescriban con datos. Se puede aprovechar este comportamiento para inyectar código malicioso en la memoria, lo que puede generar una vulnerabilidad en la máquina objetivo.

- Falsificación de solicitud en sitios cruzados (CSRF): la falsificación de solicitud en sitios cruzados implica engañar a una víctima para que realice una solicitud que utilice su autenticación o autorización. Al aprovecharse de los privilegios de la cuenta de un usuario, un atacante puede enviar una solicitud que se haga pasar por el usuario. Una vez que la cuenta de un usuario se haya visto comprometida, el atacante tiene la capacidad de filtrar, destruir o modificar datos importantes. Suelen ser un objetivo habitual las cuentas con privilegios altos, como las de administradores o ejecutivos.
- Relleno de credenciales: los atacantes pueden utilizar bots para introducir rápidamente un gran número de combinaciones robadas de nombre de usuario y contraseña en el portal de inicio de sesión de una aplicación web. Si esta práctica permite que el atacante en ruta acceda a la cuenta de un usuario real, puede robar sus datos o realizar compras fraudulentas en su nombre.
- Apropiación de página: los atacantes también pueden utilizar bots para robar contenido de páginas web a gran escala. Pueden utilizar este contenido para obtener una ventaja de precios sobre un competidor, imitar al propietario de la página con fines maliciosos u otros motivos.
- Abuso de API: las API, o Interfaces de Programación de Aplicaciones, son programas informáticos que permiten que dos aplicaciones se comuniquen entre sí. Como cualquier tipo de software, pueden tener vulnerabilidades que permitan a los atacantes en ruta enviar código malicioso a una de las aplicaciones o interceptar datos confidenciales cuando pasan de una aplicación a otra. Se trata de un tipo de ataque cada vez más común a medida que aumenta el uso de las API. La lista OWASP API Top Ten resume

sucintamente los principales riesgos para la seguridad de las API a los que se enfrentan las organizaciones hoy en día.

- API paralelas: los equipos de desarrollo trabajan rápidamente para cumplir los objetivos empresariales, construyendo y publicando con frecuencia las API sin informar a los equipos de seguridad. Estas API desconocidas pueden exponer datos confidenciales de la empresa, operando de forma "paralela", ya que los equipos de seguridad encargados de proteger las API desconocen su existencia.
- Abuso de código de terceros: muchas aplicaciones web modernas utilizan diversas herramientas de terceros; por ejemplo, un sitio de comercio electrónico que utiliza una herramienta de procesamiento de pagos de terceros. Si los atacantes encuentran una vulnerabilidad en una de estas herramientas, pueden ser capaces de comprometerla y robar los datos que procesa, impedir que funcione o utilizarla para inyectar código malicioso en otra parte de la aplicación. Los ataques Magecart, que sustraen los datos de las tarjetas de crédito de los procesadores de pagos, son un ejemplo de este tipo de ataque. Estos ataques también se consideran ataques a la cadena de suministro del navegador.
- Desconfiguraciones de la superficie de ataque: la superficie de ataque de una organización es toda su huella informática susceptible de sufrir ciberataques: servidores, dispositivos, SaaS y activos en la nube accesibles desde Internet. Esta superficie de ataques puede seguir siendo vulnerable a los ataques debido a que ciertos elementos se pasan por alto o están mal configurados.

¿Cómo funciona la seguridad en las aplicaciones web?

Además de preservar la tecnología y las características utilizadas en el desarrollo de aplicaciones, la seguridad en aplicaciones web también establece un alto nivel de protección hacia los servidores y procesos. Además, protege los servicios web, como las API, contra las amenazas en línea.

El aspecto crítico de la seguridad en las aplicaciones web es garantizar que las aplicaciones funcionen de forma segura y sin problemas en todo momento. Para lograr este objetivo, se puede empezar con un testing de seguridad en profundidad.

El testing de seguridad implica descubrir y arreglar todas las vulnerabilidades antes de que los hackers lleguen a ellas. Por ello, es muy recomendable realizar el testing en aplicaciones web durante las etapas del SDLC (ciclo de vida del desarrollo de software), y no después de que la aplicación web haya sido lanzada.

¿Cuáles son las estrategias de seguridad importantes de las aplicaciones web?

Como ya se ha dicho, la seguridad de las aplicaciones web es una disciplina amplia y que cambia constantemente. Como tal, las mejores prácticas de la disciplina cambian conforme surgen nuevos ataques y vulnerabilidades. Sin embargo, el panorama moderno de las amenazas de Internet es lo suficientemente activo como para que ninguna organización pueda arreglárselas sin ciertos servicios de seguridad "tablas apuestas" que se adapten a las necesidades específicas de su empresa:

- Mitigación DDoS: los servicios de mitigación DDoS se sitúan entre un servidor y la Internet pública, utilizando un filtrado especializado y una capacidad de ancho de banda extremadamente alta para impedir que las oleadas de tráfico malicioso desborden el servidor. Estos servicios son importantes porque muchos de los ataques DDoS modernos envían suficiente tráfico malicioso para abrumar incluso a los servidores más resistentes.
- Firewall de aplicaciones web (WAF): que filtran el tráfico que se sabe o sospecha que aprovecha vulnerabilidades de las aplicaciones web. Los WAF son importantes porque las nuevas vulnerabilidades surgen demasiado rápida y silenciosamente para que casi todas las organizaciones las detecten por su cuenta.

- Puerta de enlaces API: que ayudan a identificar las "API paralelas" que se pasan por alto y que bloquean el tráfico que se sabe o se sospecha que aprovecha las vulnerabilidades de las API. También ayudan a gestionar y controlar el tráfico de las API.(Más información sobre la seguridad de la API).
- DNSSEC: un protocolo que garantiza que el tráfico DNS de una aplicación web se dirige de forma segura a los servidores correctos, para que los usuarios no sean interceptados por un atacante en ruta.
- Gestión de certificados de encriptación: en la que un tercero gestiona elementos clave del proceso de encriptación SSL/TLS, como la generación de claves privadas, la renovación de certificados y la revocación de certificados por vulnerabilidades. Esto elimina el riesgo de que esos elementos pasen desapercibidos y que expongan el tráfico privado.
- Gestión de bots: que utiliza el aprendizaje automático y otros métodos de detección especializados para distinguir el tráfico automatizado de los usuarios humanos e impedir que los primeros accedan a una aplicación web.
- Seguridad del lado del cliente: que comprueba las nuevas dependencias de JavaScript de terceros y los cambios de código de terceros, ayudando a las organizaciones a detectar antes las actividades maliciosas.
- Gestión de la superficie de ataques: las herramientas de gestión de la superficie de ataques procesables deben proporcionar un único lugar para mapear su superficie de ataque, identificar posibles riesgos de seguridad y mitigarlos con unos pocos clics.

¿Qué buenas prácticas de seguridad de la aplicación deben esperar las organizaciones de sus proveedores?

Los desarrolladores web pueden diseñar y construir aplicaciones de modo que impidan a los atacantes en ruta acceder a datos privados, acceder fraudulentamente a cuentas de usuario y realizar otras acciones maliciosas. La lista OWASP Top 10

captura los riesgos más comunes para la seguridad de la aplicación que los desarrolladores deben conocer. Las prácticas para prevenir estos riesgos incluyen:

- Exigir la validación de la entrada: bloquear el paso de datos mal formateados a través de los flujos de trabajo de la aplicación ayuda a evitar que entre código malicioso en la aplicación mediante un ataque de inyección.
- Utilizar una encriptación actualizada: almacenar los datos de los usuarios de forma encriptada, además de utilizar HTTPS para encriptar la transmisión del tráfico entrante y saliente, ayuda a evitar que los atacantes roben datos.
- Ofrecer una autenticación fuerte y autorización: incorporar y aplicar controles para contraseñas fuertes, ofrecer opciones de autenticación multifactor, incluyendo claves difíciles, ofrecer opciones de control de acceso y otras prácticas dificultan a los atacantes el acceso fraudulento a cuentas de usuario y moverse lateralmente dentro de u aplicación.
- Seguimiento de las API: existen herramientas para identificar las "API paralelas" que se pasan por alto y que podrían constituir una superficie de ataque, pero la seguridad de las API es más fácil cuando estas nunca se pasan por alto, en primer lugar.
- Documentar los cambios de código: lo que ayuda a los equipos de seguridad y desarrolladores a solucionar antes las vulnerabilidades recién introducidas.

Conclusión

Tras la realización de este trabajo, he podido resaltar y recuperar varias de las nociones que tenía previamente sobre el desarrollo de aplicaciones web y su historia.

Es interesante el ver como evolucionaron las aplicaciones web de ser solamente documentos de texto a ser portales enteros que funcionan de manera similar a una aplicación de escritorio, y es curioso pensar que tales aplicaciones fueron diseñadas principalmente con propósitos distintos con la guerra fría.

También fui capaz de confirmar nuevamente el papel crucial que tiene la arquitectura web en el funcionamiento de las plataformas en línea. Todo esto a que una arquitectura bien diseñada garantiza que las aplicaciones web sean eficientes, escalables y capaces de adaptarse a las demandas cambiantes de los usuarios. Esto no solo mejora la experiencia del usuario, sino que también facilita la expansión y evolución de las aplicaciones con el tiempo, permitiendo que sigan siendo relevantes y efectivas.

Con la investigación también fui capaz de notar que la elección de las tecnologías adecuadas puede marcar una diferencia significativa en términos de velocidad de desarrollo, compatibilidad con diferentes dispositivos y capacidad de respuesta. Estas tecnologías son el lenguaje a través del cual los desarrolladores dan vida a sus ideas, lo que subraya su importancia en la creación de experiencias web innovadoras y efectivas.

La planificación de aplicaciones web es un aspecto a menudo subestimado pero esencial en el proceso de desarrollo. Definir objetivos claros, estrategias de desarrollo y plazos realistas es esencial para mantener un proyecto enfocado y eficiente. La planificación adecuada reduce los riesgos y los costos, alineando la aplicación con las necesidades y expectativas de los usuarios finales y asegurando que el proyecto se desarrolle de manera efectiva y exitosa.

Finalmente, la seguridad en las aplicaciones web no puede subestimarse en un mundo digital donde las amenazas cibernéticas son una constante. Proteger la

información confidencial y salvaguardar la integridad de las aplicaciones es esencial para mantener la confianza de los usuarios y la reputación de una organización. La importancia de la seguridad en las aplicaciones web radica en su capacidad para prevenir brechas de datos, ataques cibernéticos y pérdidas financieras, creando un entorno en línea más seguro y protegido para todos.

Fuentes bibliográficas

1.1. Evolución de las aplicaciones web. (2015, 17 mayo). programación Web.

<https://programacion1class.wordpress.com/1-1-evolucion-de-las-aplicaciones-web/>

colaboradores de Wikipedia. (2023). Aplicación web. *Wikipedia, la enciclopedia libre*.

https://es.wikipedia.org/wiki/Aplicación_web

Kariyawasam, M. (2021, 14 diciembre). The evolution of the Web: How application development changed as a result. *Medium*.

https://medium.com/@mehani_kariyawasam/the-evolution-of-the-web-how-application-development-changed-as-a-result-84354e6349bf

2.1 Arquitectura de las aplicaciones web. (2013, 14 noviembre). Programacion Web.

<https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>

Cloudflare, Inc. (s. f.). *¿Qué es la seguridad en las aplicaciones web?* Cloudflare.

Recuperado 1 de septiembre de 2023, de <https://www.cloudflare.com/es-es/learning/security/what-is-web-application-security/>

Infante, D. C. H., & Infante, D. C. H. (2023). Seguridad en aplicaciones web: qué es, cómo funciona y los mejores servicios. *Tutoriales Hostinger*.

<https://www.hostinger.mx/tutoriales/seguridad-en-aplicaciones-web>

KeepCoding, R. (2023, 11 mayo). 7 tecnologías para desarrollo web | KeepCoding

Bootcamps. *KeepCoding Bootcamps*. <https://keepcoding.io/blog/7-tecnologias-para-desarrollo-web/>

- López Martínez, A. & Universidad Tecnológica del Sureste S.C. (2022). *La planificación de aplicaciones web*. STUDOCU. Recuperado 1 de septiembre de 2023, de <https://www.studocu.com/es-mx/document/universidad-tecnologica-del-sureste-sc/administracion-del-tiempo/241343389-14-planificacion-de-aplicaciones-web/25004152>
- S, C. (2023). Tecnologías para el desarrollo de aplicaciones web | ILERNA. *Blog de ILERNA Online*. <https://www.ilterna.es/blog/fp-online/tecnologias-desarrollo-aplicaciones-web/>