

Índice de contenido

1 Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	2
2 Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	3
3 Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.....	4
4 Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....	4
5. Modelo de división funcional front-end / back-end para aplicaciones web.....	5
6. Página web estática – página web dinámica – aplicación web.....	6
7. Componentes de una aplicación web.....	6
8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.....	7
9 Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....	9
10 Características y posibilidades de desarrollo de una plataforma XAMPP.....	9
11 En que casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.....	9
12 IDE más utilizados (características y grado de implantación actual).....	10
13 Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	12
14 Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual)....	12
15. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen,	13
16 Repositorios de software – sistemas de control de versiones para aplicaciones web para aplicaciones sobre los Web: GIT , CVS, Subversion,	15
17 Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-W7ED.....	18
18 Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.....	21

1 Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS

Podríamos definir protocolo de comunicación como las reglas para transmitir información entre dos puntos, entre las partes básicas de un protocolo encontramos el conjunto de símbolos (el conjunto de caracteres), unas reglas usadas para la sincronización de los mensajes construidos a base del conjunto de caracteres y los procedimientos para determinar cuando ocurre un error en la transmisión y como solucionarlo. Algunos de los elementos básicos de los protocolos de comunicación son: un conjunto de caracteres que construyen reglas para la sincronización de los mensajes además de los procedimientos para saber si algún error ha sucedido durante la transmisión, este conjunto de caracteres se creará de tal forma que sea entendible por las personas (caracteres imprimibles), también habrá otro subconjunto para transmitir información de control (caracteres no imprimibles), para que exista comunicación entre ambos puntos del canal estos deben usar la misma configuración. Existen protocolos de bajo nivel que consiste simplemente en que las señales se transmitan llevando la información y los procedimientos de control a los distintos nodos, mientras que el protocolo de red determina el modo y la organización de la información transmitida por el medio físico.

Entre los protocolos de comunicación encontraremos los siguientes:

-IP: Siglas de Internet Protocol o protocolo de internet, se encuentra en la capa de red del modelo OSI, en una red con protocolo IP los datos se envían mediante paquetes, es decir bloques de datos. Este protocolo no dispone de ningún mecanismo para determinar si un paquete alcanza su destino. El protocolo IP posee tres campos para conocer el destinatario del paquete, estos son la dirección IP (dirección del equipo en cuestión), la máscara de subred (permite al protocolo establecer la parte de la IP relacionada con la red) y el campo de pasarela predeterminada hace que el protocolo de internet sepa a qué equipo enviar los paquetes si este no se encuentra en la red de área local.

-TCP: Siglas de Protocolo de Control de Transmisión es uno de los protocolos de la capa de transporte en el modelo TCP/IP, permite administrar datos que vienen desde el nivel más bajo del modelo o van a este, es un protocolo orientado a conexión. Este protocolo permite ordenar de nuevo los paquetes de datos que vienen de un protocolo IP, permite monitorizar el flujo de datos, que los datos se agrupen en conjuntos de tamaño variable y multiplexar los datos (permitir que la información que viene de distintas fuentes puedan circular a la vez), este protocolo se asegura de que no suceda ninguna pérdida de paquetes y que los datos recibidos son consistentes están en orden y completos.

-HTTP: Siglas de Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto), es un protocolo de comunicación que permite transferir información en internet, está orientado a transacciones y sigue el esquema de petición-respuesta entre cliente y servidor, el cliente llamado user agent (agente usuario) realiza una petición mediante un mensaje con un formato hacia el servidor el servidor web le envía una respuesta, los mensajes mandados son mediante texto plano lo que facilita su comprensión, estos mensajes poseen una estructura determinada:

-Línea inicial: para las peticiones lleva un método de petición tras esta la URL y la versión HTTP del cliente termina con un retorno de carro y un salto de línea.

-Cabezeras terminan en una línea en blanco y se componen de metadatos

-Cuerpo del mensaje suelen tener los datos que se intercambian entre el cliente y el servidor,

es opcional.

HTTP posee una serie de metodos de peticion conocidos como verbos, el protocolo es flexible por lo que puede ir añadiendo metodos nuevos.

Podemos encontrar HEAD, pide una respuesta al igual que get pero no devuelve el cuerpo, GET que pide una representacion del recurso especificado, Put que sube o carga un recurso especifico, Post que envia los datos para ser procesados, Delete que elimina el recurso entre otros.

-HTTPS: Siglas de protocolo seguro de transferencia de hipertexto, es una versión más segura de HTTP, la pagina web codifica la sesión con un certificado digital, estas paginas utilizan un cifrado basado en SSL/TLS, el nivel de cifrado depende tanto del servidor como del navegador utilizado por el cliente, las urls de http comienzan por http:// mientras las de HTTPS https:// el puerto por defecto de http es el 80 mientras el de https es el 443.

2 Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.

Una aplicación consta de una parte de servidor y una de cliente, que pueden ser ejecutadas en un mismo sistema en dieferentes. Los usuarios invocan la parte cliente que crea una solicitud para el servicio y es enviada al servidor de la aplicacion que usa TCP/IP, el servidor recibe una solicitud m realiza el servicio y devuelve los resultados como una respuesta, por norma general los servidores pueden tratar más de un cliente realizando peticiones a la vez. Basicamente las tareas se reparten entre los servidores y los clientes estos ultimos realizan una peticion y el servidor provee una respuesta, la separacion cliente-servidor es logica donde el servidor no tiene por que estar en una única máquina mi tan siquiera tiene por que ser un único programa. La red cliente-servidor es una red de comunicación el la que los clientes se conectan a un servidor, en este se centralizan los diversos recursos y aplicaciones. En el modelo cliente-servidor tanto el uno como el otro tienen distintas características:

El cliente:

- Inicia solicitudes
- Recibe respuestas
- Normalmente puede conectarse a varios servidores a la vez
- Suelen interactuar directamente con el usuario(muchas veces mediante interfaz grafica)

El servidor:

- Al iniciarse esperan a la llegada de las solicitudes de los clientes
- Cuando reciben la solicitud la procesan y envian una respuesta al cliente
- Suelen aceptar la conxion de un elevado numero de clientes

Ademas la arquitectura cliente servidor consta de unas características determinadas:

-Cliente y servidor pueden actuar como una unica entidad o como entidades separadas realizando tareas independientes

-Las funciones del cliente y el servidor pueden encontrarse en la misma plataforma o en distintas.

-Los cambios en las plataformas se pueden ralizar de manera independiente.

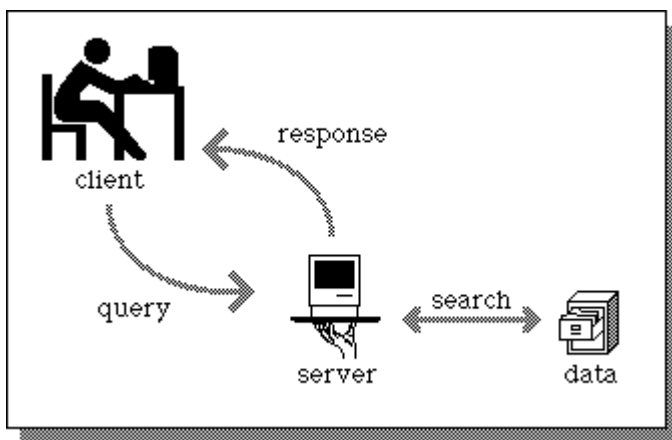
-La relacion entre hardware y software estan basadas en una infraestructura potente, de esta

manera el acceso a los recursos no muestran la complejidad de los distintos tipos de formatos en los datos y en los protocolos

Los servidores pueden ser de dos maneras apátridos(no guarada información alguna entre peticiones) o stateful(puede recordar la información entre peticiones), la interaccion entre cliente y servidor se suele describir usando diagramas de frecuencia.

Este modelo, el modelo Cliente-Servidor actualmente domina el ámbito de la comucicacion ya que descentraliza procesos y recursos.

Mientras que el cliente se limita a pedir un dato o proceso específico el servidor se los da, alguna vez el cliente maneja algún proceso pero necesita recursos del servidor para completarlo



Como podemos observar en la imagen anterior el cliente realiza una petición al servidor este busca entre sus datos y devuelve una respuesta al cliente

3 Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.

HTTP posee un conjunto de métodos de petición para indicar que acción se desea realizar con un recurso en cuestión, aunque pueden ser sustantivos estos metodos son conocidos como HTTP verbs(verbos HTTP), metodos:

-GET: es un metodo con el que el cliente solicita al servidor que le de información identificada en la propia URL, se suelen referir a un documento HTML, a una imagen o incluso a un programa de base de datos. Se puede señalar que get puede mantenerse en cache y en el historial del navegador, nunca se debe usar get cuando se usa informacion importante, los pedidos con get tienen restricciones de tamaño.

-POST: a diferencia del metodo GET usado para recuperar información, Post es usado para enviarla a un servidor web, los pedidos con post no quedan guardados y ademas tampoco tienen una restricción en el tamaño de los datos

-HEAD: este metodo pide una respuesta identica a GET pero en este caso sin el cuerpo de la respuesta, es decir pide solo la cabecera de un recurso como si se pidiera con get, la respuesta al

método GET no debe tener cuerpo

-PUT: reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición

-DELETE: este método borra un recurso en particular

-CONNECT: establece una conexión tunel hacia el servidor indentificado por el recurso.

-OPTIONS: este método se usa para describir las opciones de comunicación para el recurso de destino

-TRACE: este método realiza una prueba de bucle de retorno de mensaje en la ruta al recurso de destino. Se usa con la intención de descubrir fallos o si hay servidores intermedios

-PATCH: aplica modificaciones parciales a uno de los recursos

Cualquier cliente web puede usar estos métodos e incluso se puede configurar un servidor web para que reciba cualquier combinación de estos.

4 Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa

El principal objetivo de este tipo de desarrollos es la separación de las diferentes partes que componen un software o una arquitectura cliente servidor (capa de presentación, negocios y acceso a datos), la capa de presentación es aquella que observa el usuario final, donde vera la página web (su interfaz, su diseño etc) es decir el front-end esta capa captura la información del usuario y pretende ser de fácil uso con el usuario esta capa se comunica con la capa de negocios, en esta capa se encuentran los programas que se ejecutan, se reciben las peticiones de los usuarios y tras el proceso se envían respuestas, se denomina de negocio pues es donde se establecen las reglas a cumplir, esta capa se comunica tanto con la capa de presentación como con la de datos, de la primera recibe solicitudes y presenta resultados y con la de datos para guardar o leer información, la última capa, la de datos es la que se encarga de almacenar y acceder a los datos, esta formada por uno o varios gestores de bases de datos, recibe solicitudes de almacenamiento o recuperación de datos desde la capa de negocio. Las aplicaciones multicapa distribuyen los datos y se comunican en una red de área local o bien sobre internet. Esto proporciona muchas ventajas tales como centralizar la lógica de negocio en un sólo servidor donde un gran número de clientes tiran de él

Todas las capas expuestas con anterioridad pueden encontrarse en un único ordenador aunque no es lo normal es que estén en distintos equipos.

Ahora expondremos las tres capas:

-Capa de presentación: es la capa que visualiza el usuario, nos muestra una interfaz gráfica del recurso solicitado y recoge las interacciones entre usuario e interfaz, se suele encontrar situada en el cliente y se centra en formatear la información enviada al servidor y capturarla. Esta capa recoge la información, manda esta al servidor, recibe los resultados a partir de los cuales generan una presentación y visualizan el contenido

-Capa de negocio: esta capa gestiona las funcionalidades del sistema, aquí se reciben las peticiones del usuario y se mandan respuestas después de procesar la información dada por el cliente, esta capa puede estar programada tanto en un entorno cliente como en un entorno servidor.

-Capa de persistencia: también conocida como capa de datos, esta capa se encarga de acceder a los datos, suelen tener uno o más gestores de bases de datos.

El modelo cliente-servidor lo podremos diferenciar de tres maneras distintas estas son:

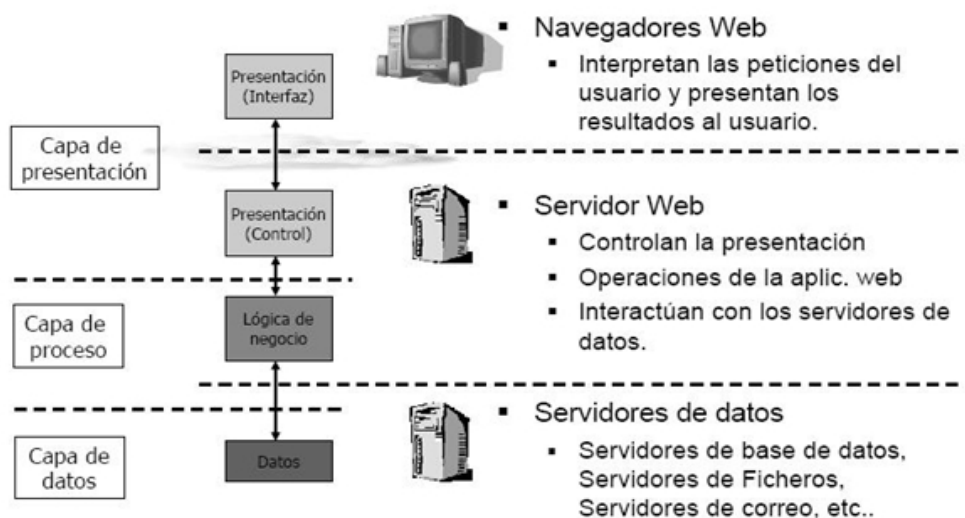
-Dependiendo del tamaño de los componentes: esta se basa en la carga de procesamiento que han de soportar los elementos de la arquitectura web, aquí podríamos hacer dos distinciones:

-Fat client/thin server: el mayor peso de la aplicación recae en el cliente mientras el servidor simplemente administra datos.

-Fat Server/thin client: en este caso el cliente se limita a mostrar la información enviada por el servidor que hace todo el trabajo.

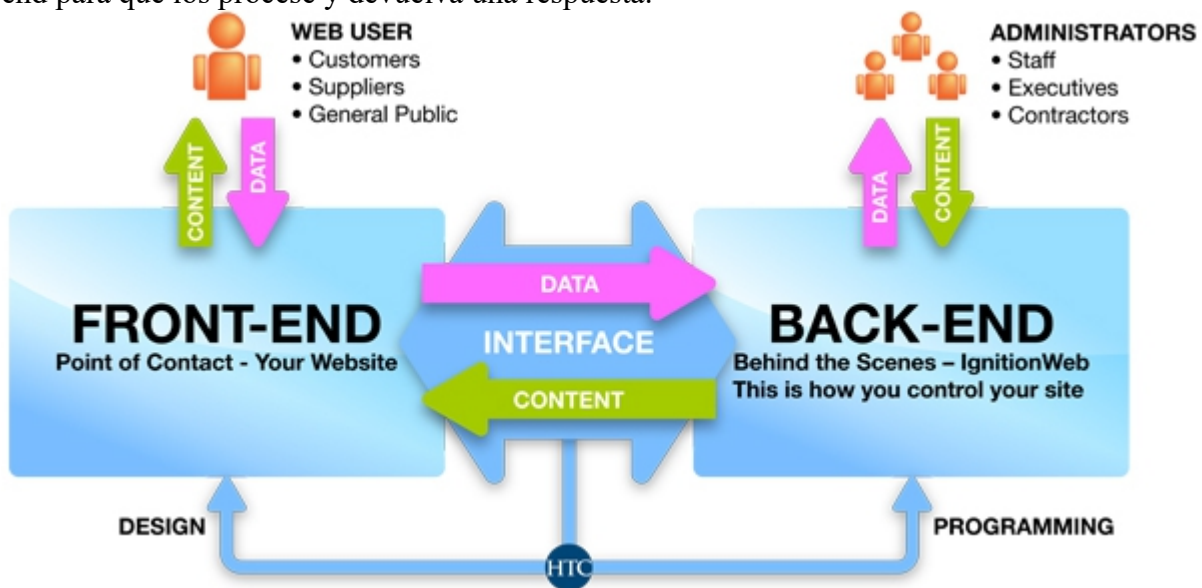
-Si depende de la naturaleza del servicio podemos encontrar servidores de ficheros, donde el cliente busca acceder a datos contenidos en ficheros, servidores de bases de datos basado en la provisión y administración de sistemas gestores bases de datos, también servidores de transacción en el que la información entre el cliente y el servidor se manda en un único mensaje, los servidores que usan objetos son conocidos como servidores de objetos, por último encontramos los llamados servidores web que conforman el modelo WWW (World Wide Web), basada en clientes simples comunicados mediante HTTP a servidores Web.

-Por último si depende del reparto entre el cliente y el servidor se nos permite distribuir las tareas entre cliente y servidor, se suele tener entre dos y tres capas (capa de presentación, capa de negocio, capa de persistencia/datos) si la capa de negocio y datos se juntan será un modelo en dos capas, esta separación se puede ver desde un punto de vista de software como de hardware.



5. Modelo de división funcional front-end / back-end para aplicaciones web.

Podemos agrupar el modelo en torno a su funcionalidad de dos maneras distintas Front-end y Back-end, la primera es la funcionalidad que se comunica directamente con el usuario, y ha de ser amigable con este, en esta funcionalidad encontramos, el diseño de la web, interfaz, como se ve la propia web, entre otros, esta funcionalidad se comunica con el Back-end, aquí encontramos la base de datos, el servidor etc, esta última parte se encarga de procesar los datos recibidos desde el front end, en terminos generales el front-end recoge los datos que introduce el usuario y los envia al back end para que los procese y devuelva una respuesta.



6. Página web estática – página web dinámica – aplicación web

Podemos encontrar diferentes tipos de páginas web y si clasificamos a estas desde una manera funcional (interacción entre el usuario y la interfaz y las respuestas recibidas) podemos encontrar tres tipos:

- Web estática: estas páginas se limitan a mostrarse ante el usuario el cual no puede interactuar con ella de ninguna manera. Lo más normal es que usen solamente el lenguaje HTML. Estas páginas muestran contenido fijo. Características:

- Creada principalmente en XHTML o HTML
- Para modificarla hay que acceder al servidor donde esta alojada la página
- No ofrece bases de datos
- El usuario no puede modificar la página

- Web dinámica: en estas páginas las acciones del cliente se ven reflejadas en una respuesta de la página, cambiando de alguna manera (cambiar, crear, ocultar algún elemento determinado), estas páginas se suelen crear con CSS, HTML, JavaScript entre otros.

- Estas páginas nos ofrecen más posibilidades en cuanto a diseño y desarrollo
- Se actualizan más fácilmente

-Mayor numero de posibilidades(bases de datos, contenido dinamico etc)

En las páginas dinámicas el contenido suele generarse en el momento de visualizarse pudiendo variar por tanto este

Entre las diferencias más importantes entre página estática y dinámica encontramos el hecho de que en las páginas estáticas es más tedioso actualizarlas, el usuario es incapaz de seleccionar, ordenar, modificar etc el contenido o diseño de la página y para cambiarlo se ha de acceder al servidor por otra parte la página web dinámica se puede alterar tanto en diseño como en contenido, se actualiza de manera más sencilla y puede contener funcionalidades tales como bases de datos, foros etc

-Aplicación web:son las más usadas en internet actualmente, producen que al interactuar el usuario se produzca una variación en el recurso recibido, en este tipo se genera un dialogo entre cliente y servidor. Con una aplicación web los usuarios pueden acceder a un servidor web mediante un navegador, una de las mayores bazas de las aplicaciones web es el hecho de que usan el navegador el cual es ligero, es independiente del sistema operativo, es muchísimo más fácil actualizarlas pues no hay que volver a distribuir las a los usuarios, las aplicaciones web ahorran tiempo pues no es necesario descargarlas aunqu esto tiene el lado negativo pues dependen de un tercero

7. Componentes de una aplicación web.

Hay tres componentes principales:

-Cliente(un navegador web):los clientes manejan la logica de presentacion, que controla como los usuarios interactuan con la aplicación, hay casos en los que el cliente valida la entrada que proporciona el usuario. Es usualmente un navegador que interactua con el contenedor web.

-Servidor de aplicaciones web:gestiona la logica empresarial que da soporte a aplicaciones de varios niveles, generalmente esta escrita en java, puede gestionar peticiones de varios clientes remotos, e incluir archivos JSPm servlets java y componentes de Enterprise JavaBeans(EJB), cuando recibe una petición del usuario puede atenderla de manera estática, mediante un fichero como descargar una imagen o dinamicamente ejecutando algún programa, el servidor de aplicaciones se comunica con varios servicios, un servidor de aplicaciones gestiona practicamente casi todas las funciones lógicas de la aplicaciones

-Servidor de bases de datos: proporciona los datos de la aplicación y permite almacenarlos e incluso puede ofrecer logica mediante procedimientos almacenados. Se usa para almacenar, recuperar y administrar los datos de una base de datos, el servidor gestiona las actualizaciones de datos permite el acceso simultaneo de muchos servidores o usuarios web además de garantizar tanto la seguridad como la integridad de los datos

8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.

-Lado del Servidor: los lenguajes en el lado del servidor son aquellos que se ejecutan en el servidor antes de que se envíen las páginas por internet hacia el cliente, estas páginas pueden realizar accesos a bases de datos, conexiones de red, entre otras tareas, entre los lenguajes de este tipo más utilizados encontramos:

-ASP.Net:(Active Service Page):Sucesor de las tecnologías ASP esta desarrollado por microsoft, las páginas creadas con esta tecnología funciona en prácticamente todos los navegadores incluyendo Internet explorer, Safari o Netscape.

Ventajas:

- Controles de usuario y personalizado.
- Orientado a objetos
- Division entre capa de diseño y el código
- El mantenimiento de grandes aplicaciones resulta más sencillo
- Mayor velocidad de respuesta del servidor
- Mayor seguridad

Desventajas:

- Consume muchos recursos
- Tecnología propietaria
- Hospedar un sitio web resulta costoso

-JSP:(Java Server Pages):sirve para crear webs dinámicas, orientado a desarrollar sitios web en java. Es un lenguaje multiplataforma, comparte ventajas parecidas a ASP.NET, tiene un motor de páginas basado en los servlets de java, para que funcione hay que tener un servidor Tomcat instalado.Sus principales características son:

- Código separado de la lógica del programa
- Las páginas se compilan en la primera petición
- Nos deja separar la parte dinámica y estática de la página web
- Los archivos tienen la extensión jsp que puede ser incrustado en código HTML
- En una página JSP podemos incrustar lo siguiente:
 - Código java
 - Directivas para controlar parámetros de servlet
 - Alterar el flujo normal de ejecución de una página

Ventajas:

- Ejecución rápida de servlets
- Crear páginas del lado del servidor
- Multiplataforma
- Código bien estructurado
- Integridad con los módulos de Java
- La parte dinámica está escrita en Java

-PERL:(Practical Extraction and Report Language):soporta tanto los paradigmas de programación orientada a objetos, modular o estructurada, destaca especialmente en el procesamiento de textos y archivos, es un lenguaje basado en scripts y es portable a casi cualquier

plataforma, su punto fuerte esta en las expresiones regulares. Es un lenguaje interpretado

Ventajas:

- Sirve para juntar de forma sencillas varios programas para realizar una determinada tarea
- Es relativamente rápido aunque lento para algunas cosas como programación a bajo nivel
- Esta disponible en bastantes sistemas operativos
- El desarrollo de aplicaciones es bastante rápido

Desventajas:

- Es lento para algunas aplicaciones
- Utiliza muchos recursos

-PHP:(Hiptertext Preprocesor), es un lenguaje de programación independiente de la plataforma y rápido, al recibir el cliente solamente código HTML es compatible con todos los navegadores.

Ventajas:

- Es rápido
- Soporta en cierta manera la orientación a objetos, clases y herencia
- Es multiplataforma
- Capaz de conectarse a muchos gestores de bases de datos
- Puede usar modulos para extender su potencial
- Posee documentación en su pagina oficial que incluye descripcion y ejemplos de cada una de sus funciones
- Es libre
- Gran cantidad de funciones
- No requiere definición de tipos de variables ni manejo detallado de bajo nivel

Desventajas:

- El trabajo lo realiza el servidor y no delega en el cliente, por lo que un alto numero de peticiones podria repercutir negativamente
- La comprensión del texto podria empeorar al mezclar PHP con HTML
- Dificil de modularizar
- Dificil de organizar por capas

-Lado del Cliente: los lenguajes de programación en el lado del cliente son usados para su integración en paginas web, el código escrito en lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, con la programación al lado del cliente se pueden validar algunos de los datos en la máquina cliente antes de que se envíen al servidor, así los usuarios pueden recibir informes de errores inmediatamente, mientras continúan en esa pagina no teniendo que volver atrás. Entre los lenguajes más utilizados encontramos:

-Vbscript:(Visual basic Script) es un lenguaje de script, derivado de visual basic. Es solo compatible con internet explorer, tanto su síntesis como la forma de trabajar esta muy inspirada en Visula basic

-JavaScript: es un lenguaje de scripts compacto basado en objetos, permite realizar aplicaciones de proposito general a traves de la web y aunque no esta diseñado para el desarrollo de grandes aplicaciones es suficiente para la implementacion de aplicaciones web completas o interfaces hacia otras más complejas, es un lenguaje con muchas posibilidades, nos ofrece dos de las capacidades clásicas de los lenguajes orientados a objetos herencia y polimorfismo, pero ademas nos permite la creacion y manipulacion de objetos sencillos y la creacion de metodos y propiedades

para estos objetos, es muy útil para la creación de efectos dinámicos, como diapositivas e imágenes dinámicas, además debido a que se ejecuta en el navegador del cliente sirve para adaptar el aspecto al dispositivo usado, entre sus desventajas se encuentra que al ser ejecutado en el navegador del cliente, tenga una menor seguridad,

9 Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).

Entre los lenguajes utilizados encontramos ASP.net, JSP, PERL y PHP, a continuación veremos algunas de sus características:

-ASP.net: es un framework web de código abierto usado para crear aplicaciones web que usan .NET, ASP.net crea páginas web basadas en HTML5, CSS y javascript, estas pueden escalarse para millones de usuarios, además de estar diseñado para crear sitios web dinámicos puede usarse para aplicaciones web y servicios web XML, las páginas de ASP.net son conocidas como web forms, es decir formularios web, estos están en archivos cuya extensión es ASPX (contiene HTML o XHTML estático y etiquetas conocidas como controles web), ASP.net funciona tanto en el servidor de Microsoft como sobre Apache, Microsoft recomienda el modelo code-behind, es decir colocar el código en un archivo separado o dentro de una etiqueta script especial

-JSP: es una tecnología para crear páginas web dinámicas basadas en HTML y XML, se requiere un servidor compatible con contenedores servlet, la mayor ventaja de JSP es que al usar JAVA es apto para crear clases que manejan muy bien tanto la lógica de negocio como el acceso a datos, también recibe la portabilidad de Java, podríamos considerar JSP como una abstracción de alto nivel de los servlets de JAVA, las JavaServer Pages son pasadas a servlets a tiempo real, cada uno es guardado en caché para después ser reusado hasta que la JSP original se modifica

-Perl: Perl es un lenguaje de programación que toma características de C, de Bourne shell de AWK de sed y de LISP entre otros, su estructura es de bloques, una de sus ventajas es el hecho de que procesa los textos muy bien y no posee las limitaciones de otros lenguajes script, es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código limitados por llaves, subrutinas y estructuras de control a partir de PERL 5 este tomó un modelo orientado a objetos

10 Características y posibilidades de desarrollo de una plataforma XAMPP

Xampp (x: cualquier a: Apache, m: MariaDB, p: PHP, p: Perl) es un paquete de instalación que no depende de la plataforma y es software libre, está formado principalmente por MySQL (un sistema de gestión de bases de datos) ahora cambiada a MariaDB, Apache (un servidor web) y los intérpretes tanto para PHP como para Perl.

Actúa como un servidor web libre, de fácil uso y con la capacidad de interpretar páginas dinámicas, está disponible para Windows, Linux y Mac OS X. Xampp es así diseñado para poderse instalar fácilmente.

Características:

- su licencia es GNU (General Public License)

- solo requiere la descarga y ejecucion de un archivo ademas de una leve configuracion
- es multiplataforma
- incluye otros modulos como OpenSSL y phpMyAdmin
- permite probar las paginas web sin tener que acceder a internet
- es muy útil para elaborar páginas dinamicas pues su entorno permite un extenso análisis de comprobacion y diseño de estas

11 En que casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación

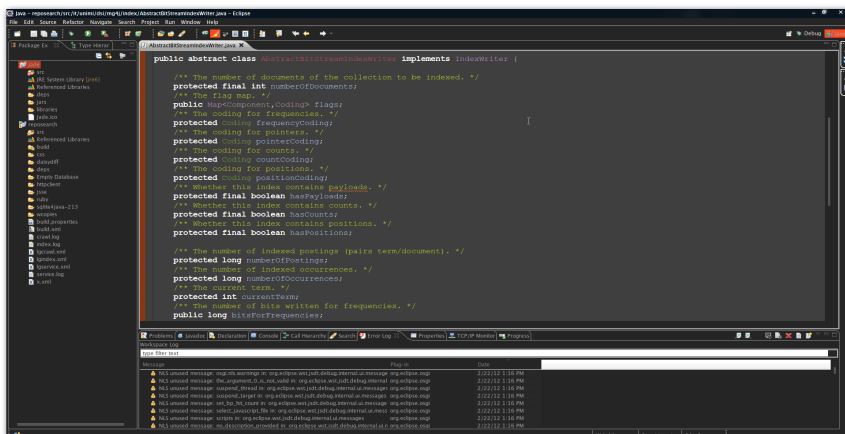
Debemos instalar en el entorno de desarrollo ambos tanto la maquina virtual de java como el JDK, pues alli desarrollaremos las aplicaciones, por otro lado en el entorno de explotación, lugar donde probaremos las aplicaciones nos hara falta unicamente la maquina virtual de java.

12 IDE más utilizados (características y grado de implantación actual).

Entre los IDEs más usados encontramos los siguiente:

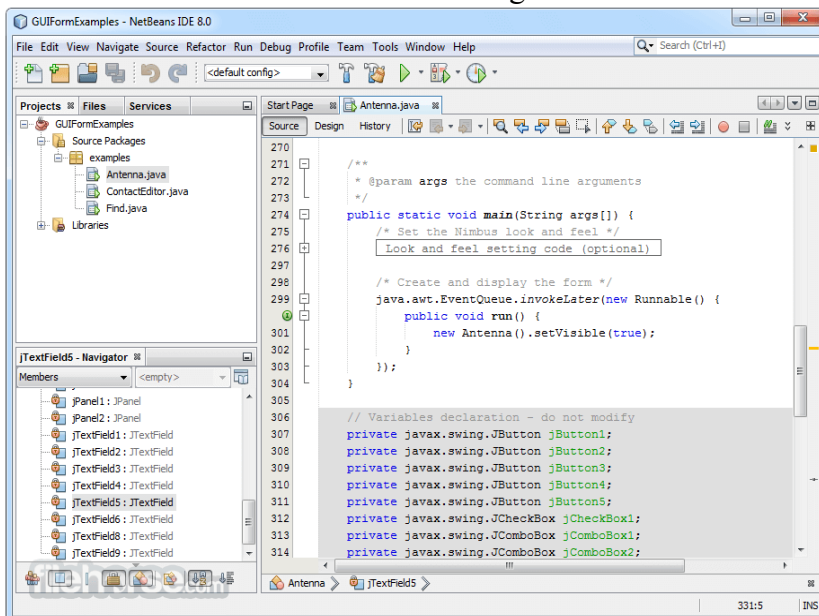
-Eclipse:este ide empezó siendo un proyecto de IBM Canada en noviembre de 2001, se formo como código abierto en 2003, es una plataforma de software formada a partir de un conjuntod ede herramientas de codigo abierto multiplataforma para desarrollar lo conocido como "Aplicaciones de Cliente Enriquecido", todo lo contrario a las aplicaciones "Cliente-liviano" basado en navegadores, esta plataforma a sido usada para desarrollar IDEs.Ahora Eclipse esta siendo desarrollado por la fundación Eclipse.Entre sus características se encuentran:

- la disponibilidad de un editor de texto con un analizador sintactico
- la compilación en tiempo real
- dispone de pruebas unitarias con Junit
- control de version con cVS
- integracion con Ant(un software para automatizar la compilación)
- asistentes para crear proyectos,clases,test...
- refactorizacion



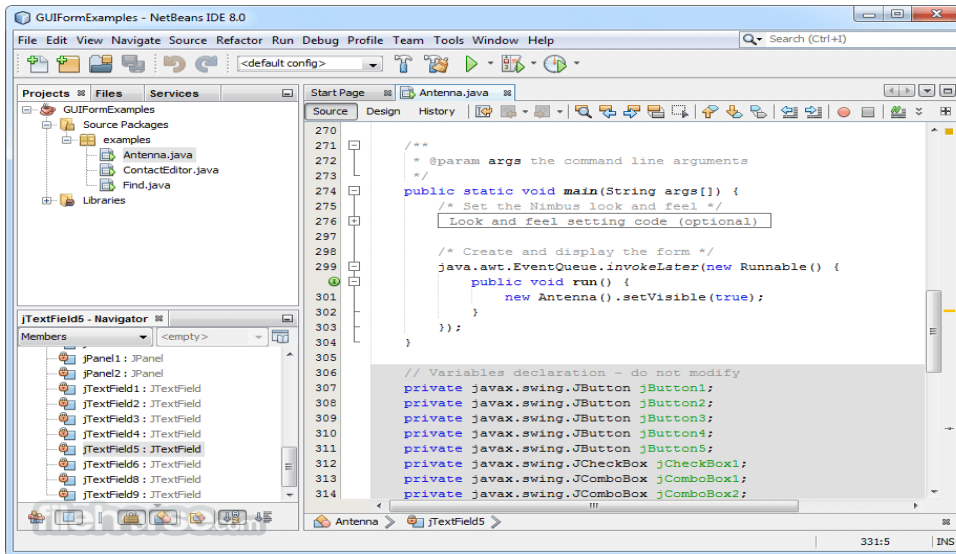
-NetBeans:es un entorno de desarrollo integrado libre, principalmente para el lenguaje Java, tiene muchos modulos para extenderlo, es libre,gratuito y de codigo abierto, netBeans permite que las aplicaciones sean desarrolladas a partir de unos componentes de software llamados módulo, netbeans soporta el desarrollo de todos los tipos de aplicación java(web,J2SE,EJB y aplicaciones moviles), posee las mismas características que enumere antes con Eclipse. Netbeans es modular pues todas las funciones de este IDE son dadas por módulos, cada uno provee una función determinada(soporte para java, control de versiones etc) ademas algunas de sus características son:

- Gestion de interfaz de usuario
- Gestion de configuracion de usuario
- Gestion de almacenamiento
- Gestion de ventanas
- Soporta dialogos paso a paso
- Libreria visual de Netbeans
- Herramientas de desarrollo integrado

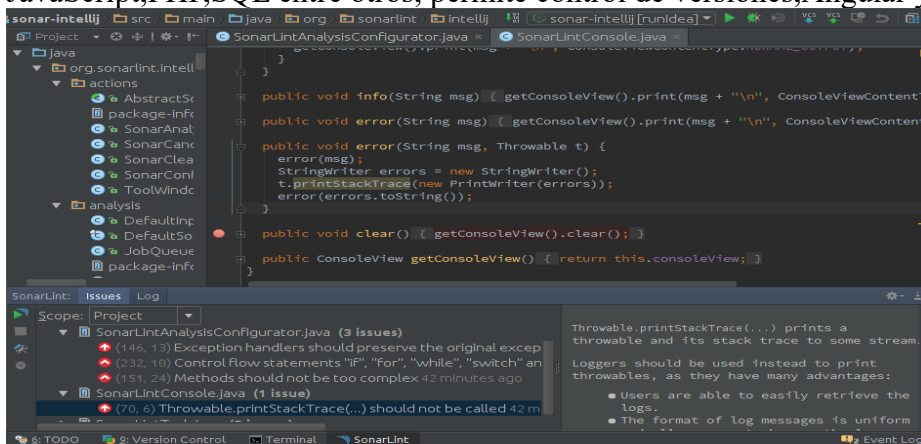


-Jdeveloper: sus primeras versiones(1998) estaban basadas en el entorno Jbuilder de Borland pero a partir la version 9i de 2001 esta basado en java perdiendo la relacion con Jbuilder, jdeveloper paso a ser gratuito en 2005 entre sus principales características encontramos

- Es un entorno gratuito
- Totalmente desarrollado para Java
- Posee diagramas de clase UML
- Funciona en Windows, Linux y Mac OSX
- puede generar codigo repetitivo y arquitectura de aplicaciones preestablecidas
- desarrollo de formularios web
- Unifica el desarrollo de la base de datos con el de la aplicacio



-IntelliJ IDEA: es un IDE desarrollado por JetBrains, no está basado en Eclipse, a partir de su versión 12.1 incluye soporte para Java 8, posee dos ediciones una de ellas comercial ambas permiten desarrollar para Java pero únicamente la versión de pago sirve para HTML, XHTML, CSS, JavaScript, PHP, SQL entre otros, permite control de versiones, Angular y Junit.



13 Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).

Los siguientes servidores son los más usados en estos momentos:

-Apache: es de código abierto, para plataformas Unix, Windows, Macintosh entre otros, es desarrollado y mantenido por usuarios bajo la supervisión de Apache Software Foundation en el proyecto HTTP Server (httpd), la mayoría de vulnerabilidades de la seguridad descubiertas y resueltas solo pueden ser aprovechadas por usuarios locales, no de manera remota.

-Nginx: es un servidor web ligero de alto rendimiento, es software libre aunque posee una versión comercial. Es usado por sitios como WordPress, Netflix entre otros.

-Microsoft IIS: Internet Information Services (IIS) es un servidor web y un conjunto de servicios para Windows, en un principio era parte del optional Pack para Windows NT, aunque luego fue integrado en otros sistemas operativos de Microsoft que ofrecen servicios como

FTP,SMTP,NNTP y HTTP/HTTPS, este servicio convierte el PC en un servidor web, se basa en varios modulos que le ofrecen distintas características.

-Google Web Server:es es el nombre que usa google para sus infraestructuras y servidores, es un servidor personalizado de desarrollo propio usado por google

14 Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).

Entre los navegadores más utilizados encontramos:

-Google Chrome:es uno de los más conocidos y más usados, aparecio en 2008, actualmente posee una cuota del 25%, es intuitivo y sencillo, esta diseñado para ofrecer una mayor seguridad. Entre sus características encontramos:

-Una marca de tiempo(secuencia de caracteres que muestran el momento) de cuando se instalo el navegador

- paginas de error en servidores de google para cuando no se encuentra un servidor
- sugerencias automaticas en la barra de direcciones
- sistema de seguimietno de errores que envía información sobre errores
- asigna a cada tarea un proceso separado
- posee muchas extensiones, temas etc
- puede usar el traductor de google para traducir sitios
- geolocalizacion
- integracion de flash player
- visor pdf
- permite crear diferentes cuentas de usuario

-Mozilla firefox:aparecio en 2003, es un navegador muy estable y el segundo más utilizado, esta desarrollado sin animo de lucro y posee un alto grado de personalización, entre sus características encontramos:

- Es multiplataforma
- Se pueden añadir complementos,temas y plugins
- Se pueden abrir y navegar entre pestañas
- Restaurar la sesión anterior
- Corrector de ortografica
- Sugerencia de busqueda
- Rss
- Diferentes barras de busqueda integradas(google,Yahoo,ebay,RAE etc)
- Bloqueo de ventanas emergentes
- Ayudas para ser más accesible(aumentar la letra etc)
- Proteccion antiphising
- Actualizaciones automaticas
- Proteccion frente a programas espías
- Eliminacion de información privada

-Opera:es un navegador web creado por los noruegos Opera Software, tiene versiones tanto para escritorio, moviles y tablets, los sistemas operativos compatibles son:windows, mac Os X y Linux, entre los sitemas moviles esta Maemo, blackberry, Symbian, Windows phone,Ios y Android ,

características:

- almacena miniaturas de las paginas favoritas
- dispone de una amplia cantidad de extensiones
- permite navegacion privada
- administra contraseñas de paginas visitadas con frecuencia
- administrador de descargas
- disponible en más de 60 idiomas
- tiene un inspector web para ver el código HTML,CSS y JavaScript
- permite usar diferentes buscadores de internet

15. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ...

PHPDoc es una adaptación de JavaDoc para PHP, es el estandar para comentar codigo de PHP, nos permite el uso de generadores externos, PHPDoc esta compuesto por diferentes partes entre ellas: DocBlock; es un comentario al estilo de los de C++, comienza con "/*" y cada vez que hay una línea nueva se añade "*" al comienzo de esta, se permite de manera opcional poner todos los párrafos entre las etiquetas "<p>" y "</p>", si se usan estas etiquetas lo que no este dentro sera ignorado, phpDocumentor tambien permite el formato de javaDoc añadiendo la opción "-j" o "--javadoc", phpDocumentor utiliza diferentes etiquetas., existen ademas otros generadores como Doxygen que ademas de PHP funciona para Java,C entre otros..PHPDocumentor sigue la definición de PHPDoc y sigue los siguientes elementos estructurales:

- Funciones
- Constantes
- Clases
- Interfaces
- Constantes
- Métodos

Ejemplo de DockBlock

```
<?php
/**
 *bloque
 */
```

Algunas de las etiquetas más importantes:

-**@category**: es usada para organizar grupos de paquetes juntos
ejem:

```
/**
 * @category descripcion
 */
```

-**@package**: sirve categorizar los elementos estructurales en subdivisiones lógicas
ejem:

```
/**
 * @package PSR\Documentation\API
```



```
*/
```

-**@license**: indica que licencia es aplicada a los elementos asociados
ejem:

```
/**  
 * @license[<url>][nombre]  
 */
```

-**@copyright**: informa si se aplica algún copyright, suele tener el rango de años y la entidad que aplica el copyright
ejem:

```
/**  
 * @copyright 1998-2015 Grupo A  
 */
```

-**@example**: contiene la ruta a un archivo con un ejemplo de uso de la clase con esta etiqueta, una clase puede tener tantos @example como sea necesario
ejem:

```
/**  
 * @example /ruta/hacia/ejemplo.php  
 */
```

-**@version**: muestra la version actual, debe ser cambiado con cada nueva version
ejem:

```
/**  
 * @version 1.0.2  
 */
```

-**@since**: muestra desde que version esta un determinado elemento
ejem:

```
/**  
 * @since version 1.0.1  
 */
```

-**@author**: muestra el autor de alguno de los elementos
ejem:

```
/**  
 * @author Alejandro  
 */
```

-**@param**: sirve para documentar un parametro de una función o un metodo

```
*@param int $numero
```

-**@return**: documenta el valor que devuelve una función o un método
ejem:

```
/**  
 * @return integer  
 */
```

16 Repositorios de software – sistemas de control de versiones para aplicaciones web para aplicaciones sobre los Web: GIT , CVS, Subversion, ...

Podríamos definir repositorio de software como un lugar donde se almacena software que puede ser recuperado e instalado en un ordenador, muchos editores de software e incluso organizaciones poseen servidores en internet para este fin, los repositorios de software pueden ser tanto para programas en particular como para sistemas operativos completos, que ofrecen un sistema gestor de paquetes con el fin de buscar, manipular e instalar software. En pocas palabras sería un conjunto de servicios web centralizados para gestionar, preservar, organizar y ofrecer acceso a la producción en soporte digital.

Algunas de las características generales de un repositorio son:

- El apoyo a diferentes formatos de archivo
- Escalabilidad, extensibilidad y mantenimiento del sistema
- Acepta estándares de metadatos, además de búsqueda y visualización de estos
- Cumplir los principales protocolos de intercambio de registros de información
- Localización de documentos mediante identificadores persistentes
- Interfaz de búsqueda a texto completo
- Personalización de software

Hay muchos repositorios de software entre ellos:

- Bepress
 - desarrollado por "Berkley Electronic Press"
 - es comercial
 - soporta protocolo OAI-PMH
 - exportación a XML
 - búsqueda de texto completo
- Dspace
 - Desarrollado por el instituto de Massachusetts en el 2002
 - su objetivo es publicar las investigaciones institucionales
 - código abierto y gratuito bajo Licencia Berkeley
 - datos organizados en comunidades, colecciones e ítems
 - permite capturar, describir, buscar, recuperar, distribuir y preservar documentos digitales
 - soporta gran variedad de formatos
 - robusta gestión de usuarios tanto usuarios normales como anónimos o administradores
- Contentdm
 - Software comercial
 - acepta todo tipo de contenido
 - soporte de protocolo OAI-PMH
 - metadatos Dublin Core
 - exportación a xml
- CVS
 - Tiene una arquitectura cliente servidor

- Cliente y servidor pueden estar en la misma o en distintas máquinas
- Puede funcionar en casi todos los SO
- Varios clientes pueden sacar copias del proyecto al mismo tiempo
- Pueden comparar las versiones
- Se puede permitir el acceso de lectura sin necesidad de contraseña
- Puede mantener distintas versiones paralelas de un proyecto
- Por otra parte los archivos en el repositorio CVS no pueden ser renombrados

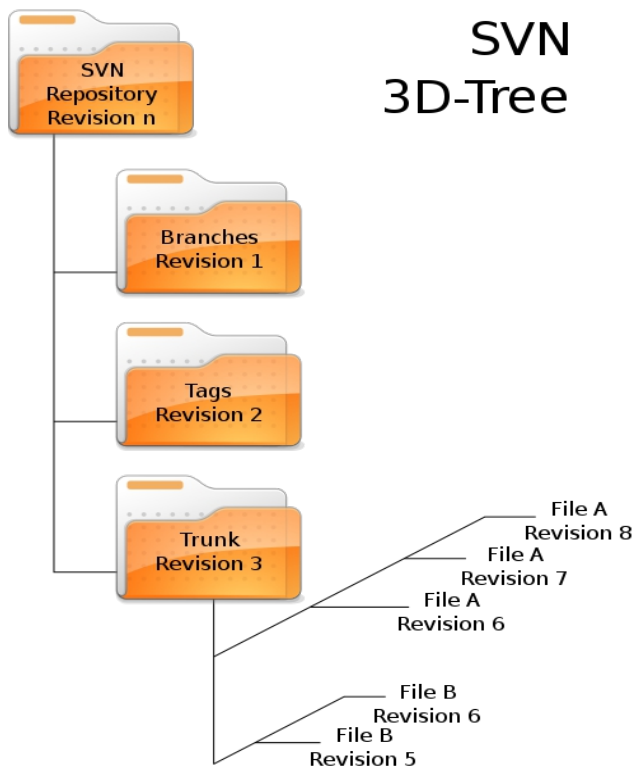
-GIT

- lanzado en 2005
- apoyo al desarrollo no lineal
- herramientas específicas para navegar y visualizar el historial
- gestión distribuida, git da a los programadores una copia local del historial del desarrollo entero y los cambios se propagan entre repositorios locales, los cambios se importan como ramas adicionales
- los almacenes de información pueden publicarse por HTTP, FTP, rsync
- gestión eficiente de proyectos grandes
- todas las versiones previas a un cambio implican la notificación de este cambio
- se deben usar cuatro ramas
 - Master: la rama principal, contiene el repositorio publicado debido a esto debe ser siempre estable
 - Development: esta rama sale del master, es la rama de integración, todas las funcionalidades se implementan en esta rama, cuando se realiza la integración y se corrigen los errores se puede hacer un merge sobre la rama master.
 - Features: cada nueva funcionalidad se tiene que realizar sobre una rama nueva centrada en esa funcionalidad, estas se deben sacar de development, donde se integran las demás funciones
 - Hotfix: los bugs surgidos en producción se tienen que arreglar cuanto antes, una vez arreglados, se hace merge sobre la rama Master, y para evitar que se desactualice un merge de master sobre development

-Subversion:

- Es de código abierto
- Su funcionamiento se asemeja a un sistema de ficheros
- Utiliza la revisión para guardar los cambios en el repositorio
- Entre dos revisiones solo guarda las modificaciones
- Permite crear, copiar y borrar carpetas
- Puede acceder al repositorio a través de la red
- Se sigue el historial de archivos y directorios a través de copias y renombrados
- Se envían solo las diferencias en ambas direcciones
- Puede ser servido mediante apache
- Buen manejo de archivos binarios
- Permite el bloqueo de archivos
- Si se usa junto a Apache permite utilizar todas las opciones de este servidor para autenticar archivos
- Por el lado negativo el manejo de cambio de nombre de archivos no es completo, sino que copia el archivo con el nuevo nombre y borra el antiguo
- no facilita llevar la cuenta de qué cambios se han realizado

Ejemplo de árbol de proyectos en Subversión:



17 Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-W7ED.

Programa/SO	Instalación	Mantenimiento
AHPUSED	SO:Ubuntu Server 16.04.3 IP:192.168.3.115 /24 Cuentas: miadmin /paso alejandro /paso Disco duro:500GB -/var:100GB -/swap:2GB Ram 500mb Programas: - IDE(netbeans) -SGBD(mysql) Otras cosas a tener	

	en cuenta: -abrir ssh	
--	--------------------------	--

Programa/SO	Intalacion	Mantenimiento
AHPW7ED	SO:Windows 7 IP:192.168.3.65 /24 Cuentas: miadmin /paso alejandro /paso Disco duro:500GB Ram: 1GB Programas: - IDE(netbeans) -navegadores (chrome,firefox etc) -cliente sftp -generador HTML -openoffice 4.1.3 -Gimp 2.8.22	

18 Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.

Programa/SO	Intalacion	Mantenimiento
AHPUSEE	SO Ubuntu Server 16.04.3 IP 192.168.3.215/24 dns 192.168.20.20 admin /paso alejandro /paso Disco duro 500GB ram 500 mb	

	Programas: -Navegadores	
--	--	--