



**GUIA TRABAJOS PRÁCTICOS**

**LABORATORIO DE**

**COMPUTACIÓN II**

Alumno: PIZARRO Rodrigo

**CICLO 2018**

# **TRABAJO PRACTICO N1**

## **Instalación y características de los Sistemas Operativos**

### **Preguntas:**

#### **1. ¿Qué es un SO?**

Un Sistema Operativo es un software que se inicia junto con el ordenador y que es el encargado de gestionar todos los recursos de la computadora. Básicamente permite la comunicación entre el usuario y la computadora. Sobre él se montan el resto de las aplicaciones necesarias. Algunos ejemplos son Windows, Mac, Linux (distintas distribuciones).

#### **2. ¿Qué diferencia existe entre los requisitos mínimos y los recomendados de hardware especificados por los SO?**

Los requisitos mínimos, son los que si no se cumplen, no se va a poder ejecutar correctamente el sistema operativo. Los recomendados, son los que si se cumplen, se asegura que el sistema operativo va a andar correctamente.

#### **3. ¿Qué es la BIOS? ¿Qué función cumple?**

La función de la BIOS es iniciar y probar el hardware del sistema y cargar un gestor de arranque. También funciona como una abstracción del sistema operativo hacia el hardware controlando algunos parámetros del hardware como la frecuencia de la memoria RAM o del microprocesador.

Otra característica es que la memoria de la BIOS es solo de lectura, ya que solo puede ser modificada por una actualización del fabricante.

#### **4. ¿Qué sucede al formatear un disco duro? ¿Qué es el formato rápido?**

Formatear un disco duro consiste en restablecer a este a su estado original para ser reutilizado o reescrito con una nueva información.

Al realizar esta operación se elimina toda la información contenida en el.

Hay dos tipos de formato al formatear un disco duro, el formato rápido y el completo.

El formato rápido no elimina físicamente los archivos y el completo sí. El primero de ellos no borra ni el sistema de archivos ni la tabla de particiones.

Esto tiene varias implicaciones de cara al uso futuro del dispositivo. Cuando se formatea el disco con un formato completo, todos tus archivos desaparecen definitivamente. No se podrán recuperar de forma posterior, por lo que se debe asegurar que se haya salvado lo fundamental antes de dar el paso (backup).

#### **5. ¿Qué es una partición? ¿Qué ventajas tiene particionar un disco duro? ¿Qué son las particiones primarias y secundarias?**

Una partición es la división de espacio que se le asigna a un disco duro.

Un disco duro puede tener varias particiones, que son de cierto modo independientes entre sí, dado que cada una de ellas tiene su propio sistema de archivos o formato, y el SO lo reconoce como un disco independiente (aunque físicamente sea uno).

Una ventaja de hacer esto es que se puede formatear una partición sin necesidad de formatear todo el disco.

#### TIPOS DE PARTICIONES:

- Partición Primaria: Se encarga de iniciar la carga del SO
- Partición Extendida o Secundaria: Almacenan información

**6. ¿Qué diferencia existe entre la instalación mínima, típica o estándar y personalizada de un programa? Explique ventajas y desventajas.**

El Software independientemente del tipo que sea:

- Sistema operativo (Ej: Windows, Linux,...).
- Drivers (Ej: Chipset de placa base, Tarjeta gráfica,...).

Entre los tipos de instalaciones de software más comunes están:

Mínima: Instala los archivos mínimos que se necesitan para poder ejecutar la aplicación, su mayor ventaja es que ocupa poco espacio en disco duro (Actualmente con la capacidad de los discos duros no tiene mucho sentido utilizar este tipo de instalación salvo alguna excepción como por ejemplo los Juegos que en los más actuales pueden ocupar varios gigabytes de espacio), sin embargo tiene una desventaja bastante clara si necesitas utilizar alguna función que esté en el disco del Software el programa nos pedirá dicho disco para poder usar dicha función con lo cual si es un programa que utilizamos asiduamente necesitaríamos tener el CD de instalación a mano por si lo pide.

Típica: Instala la mayoría de archivos que se necesitan para poder ejecutar la aplicación, ocupa más espacio que la anterior pero

normalmente no requiere el CD de instalación salvo que se utilice alguna función que no esté instalada por defecto.

Completa: (También denominada Full): Instala todos (o al menos la gran mayoría) de archivos que se necesitan para poder ejecutar la aplicación, ocupa más espacio que las anteriores pero normalmente no requiere el CD de instalación salvo que se utilice alguna función que no esté instalada por defecto (Una excepción suelen ser los juegos de ordenador/PC que aunque se haga una instalación completa suelen requerir el CD/DVD del Juego en cuestión).

Personalizada (A veces también denominada Avanzada o Custom): Permite al usuario elegir los programas que se instalarán, con diferencia es la mejor opción ya que permite al usuario elegir las aplicaciones que necesita realmente.

## 7. ¿Qué es una máquina virtual? ¿Qué ventajas posee?

Es un software que simula a un computadora y puede ejecutar programas.

Se puede configurar para que los procesos que ejecutan están limitados por los recursos proporcionados.

Habitualmente se usa para ejecutar sistemas operativos y probarlos antes de instalarlos o para probar alguna característica.

### VENTAJAS:

- Probar sistemas operativos
- Usar software que no está disponible en nuestro SO
- Copias de Seguridad y Clonación
- Crear entornos de prueba

## **TRABAJO PRACTICO N2**

### **Drivers**

#### **1. ¿Qué es un driver o controlador de dispositivo? ¿Para qué sirve?**

Un driver o controlador de dispositivo para equipos con sistema operativo Windows, es un programa cuya finalidad es relacionar el sistema operativo con los dispositivos hardware (tarjeta gráfica, tarjeta de sonido, módem, wifi, etc.) y periféricos (impresora, escáner, cámara fotográfica, cámara de vídeo, etc) de nuestro equipo.

Los drivers sirven para que el Sistema Operativo reconozca y permita trabajar con los diferentes dispositivos. Existen programas que en muchos casos nos van a permitir conocer los dispositivos que tenemos montados en nuestro equipo.

Los fabricantes de dispositivos como Discos duros, Tarjetas de sonido, Tarjetas de vídeo, Modem, Tarjetas de red, Tarjetas de Tv, etc., ponen a disposición del usuario los drivers que pueden ser descargados de su web oficial y posteriormente instalados en el equipo.

#### **2. ¿Por qué es recomendable mantener los drivers actualizados?**

Los fabricantes de hardware suelen actualizar los drivers para otorgar más funcionalidades, mejorar el rendimiento o aumentar las medidas de seguridad con la finalidad de corregir errores o vulnerabilidades que podrían comprometer el equipo de los usuarios.

#### **3. ¿Qué son los dispositivos PnP o “plug and play”?**

Plug-and-Play o PnP es la tecnología o cualquier avance que permite a un dispositivo informático ser conectado a una

computadora sin tener que configurar, mediante jumpers o software específico proporcionado por el fabricante, ni proporcionar parámetros a sus controladores. Para que sea posible, el sistema operativo con el que funciona el ordenador debe tener soporte para dicho dispositivo.

#### **4. En Windows:**

El usuario puede:

- Comprobar si el funcionamiento de los dispositivos es correcto
- Actualizar el software de controlador o Driver
- Modificar o establecer propiedades a un dispositivo
- Habilitar o deshabilitar dispositivos

#### **5. En Linux:**

- Los drivers son módulos del Kernel
- Con el comando lsmod
- Desde el visor de hardware del entorno gráfico

## **TRABAJO PRACTICO N3**

### **Herramientas de desarrollo**

#### **1. ¿Qué son las herramientas de desarrollo? ¿Qué tipo de herramientas existen?**

Una herramienta de desarrollo de software es un programa informático que usa un programador para crear, depurar, gestionar o mantener un programa .

Existen diversos tipos de herramientas con fines muy diversos, a continuación se listan algunas seguida de un ejemplo:

- Sistema de seguimiento de errores: Bugzilla
- Generador de código: Make
- Conversor de código: JTest
- Compilador: gcc

#### **2. ¿Qué significa IDE? De ejemplos de IDE's para desarrollo web ampliamente usados.**

Un entorno de desarrollo integrado, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen autocompletado inteligente de código (IntelliSense). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse; otros no, tales como SharpDevelop y Lazarus.

El límite entre un IDE y otras partes del entorno de desarrollo de software más amplio no está bien definido. Muchas veces, a los efectos de simplificar la construcción de la interfaz gráfica de usuario (GUI, por sus siglas en inglés) se integran un sistema controlador de versión y varias herramientas. Muchos IDE modernos también cuentan con un navegador de clases, un buscador de objetos y un diagrama de jerarquía de clases, para su uso con el desarrollo de software orientado a objetos.

### **3. ¿Qué herramientas suele integrar un entorno de programación? Describa cada una.**

- IDE

Es un software que proporciona servicios integrales para facilitarle al programador el desarrollo de software.

- Sistema de Control de Versiones

Permiten gestionar los archivos de un proyecto (y sus versiones) y que sus integrantes puedan acceder remotamente a ellos.

- Bug Tracker

Para asegurar una buena calidad en los proyectos de dimensión considerable es importante utilizar un Bug Tracker para realizar el seguimiento de los defectos o Bugs que surgen y son detectados.

- Pruebas de desempeño, carga, stress

Load Test: Simular la realidad a la que está expuesto el sistema cuando esté en producción.

Stress Test: Simula más carga de la esperada.

Endurance o Resistencia: Desempeño del sistema después de una carga duradera por un periodo de tiempo largo.

#### **4. ¿Qué es un sistema de gestión de versiones o VCS? ¿Qué ventajas tiene?**

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente dando lugar a los sistemas de control de código fuente o SCM . Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etc.

#### **5. ¿Qué es Git? ¿Qué servidores se pueden utilizar para ello?**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Servidores: GitHub – GitLAB

#### **6. ¿Por qué es importante la documentación de Código?**

Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, punto por punto, de

forma que no sólo los ordenadores sepan qué hacer, sino que además los humanos entiendan qué están haciendo y por qué.

Porque entre lo que tiene que hacer un programa y cómo lo hace hay una distancia impresionante: todas las horas que el programador ha dedicado a pergeñar una solución y escribirla en el lenguaje que corresponda para que el ordenador la ejecuta ciegamente.

Documentar un programa no es sólo un acto de buen hacer del programador por aquello de dejar la obra rematada. Es además una necesidad que sólo se aprecia en su debida magnitud cuando hay errores que reparar o hay que extender el programa con nuevas capacidades o adaptarlo a un nuevo escenario. Hay dos reglas que no se deben olvidar nunca:

- Todos los programas tienen errores y descubrirlos sólo es cuestión de tiempo y de que el programa tenga éxito y se utilice frecuentemente.
- Todos los programas sufren modificaciones a lo largo de su vida, al menos todos aquellos que tienen éxito.

Por una u otra razón, todo programa que tenga éxito será modificado en el futuro, bien por el programador original, bien por otro programador que le sustituya. Pensando en esta revisión de código es por lo que es importante que el programa se entienda: para poder repararlo y modificarlo.

## 7. ¿Qué es doxygen?

Doxygen es un generador de documentación para C++, C, Java, Objective-C, Python, IDL (versiones Corba y Microsoft), VHDL y en cierta medida para PHP, C# y D. Dado que es fácilmente adaptable, funciona en la mayoría de sistemas Unix así como en Windows y Mac OS X.

## 8. A) git clone

- B) git fetch**
- C) git add**
- D) git commit**
- E) git push**

## **TRABAJO PRACTICO N4**

### **Prototipado**

#### **1. ¿Qué es la UI (interfaz de usuario)? ¿Quién lo hace? ¿Por qué es importante?**

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora.

Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una plantilla de pantalla que constituye la base del prototipo de la interfaz de usuario.

Un ingeniero de software es el encargado de diseñar la interfaz de usuario con la aplicación de un proceso iterativo que sigue principios de diseño predefinidos.

Es muy importante porque si el software es difícil de usar, fuerza al usuario a cometer errores, o si frustra sus esfuerzos para alcanzar las metas, entonces no le gustará, sin que importe el poder computacional que tenga, el contenido que entregue o las funciones que ofrezca.

La interfaz tiene que estar bien hecha porque moldea la percepción que el usuario tiene del software.

#### **3. ¿Qué factores se deben considerar para realizar el diseño de una UI?**

Se deben tener en cuenta capacidades físicas y mentales de las personas que utilizarán el software.

- Las personas tienen una memoria limitada a corto plazo
- Cuando el sistema falla, y emite avisos y mensajes genera estrés en el usuario
- Las personas poseen un amplio rango de capacidades físicas

- Las personas poseen diferentes preferencias de interacción

#### 4. ¿Cuáles son los principios de diseño? Explique.

- Familiaridad del usuario: Utilizar términos que el usuario conozca
- Uniformidad: Operaciones comparables se deben activar de la misma forma
- Mínimo Sorpresa: El comportamiento del sistema debe ser esperable
- Recuperabilidad: Mecanismos de recuperación de errores
- Guía de Usuario: Retroalimentación significativa frente a errores. Asesoramiento y características.
- Diversidad de Usuarios: Interacción para diferentes tipos de usuario. (Casuales, Potenciales)

#### 5. ¿Cuáles son los estilos de interacción? ¿Qué ventajas y desventajas tienen? De ejemplos de aplicaciones.

Estilo de Interacción	Ventajas	Desventajas	Aplicación
Manipulación Directa	Interacción rápida Intuitiva Fácil de Aprender	Difícil de Implementar Sólo adecuada para sistemas visuales	Videojuegos Sistemas CAD
Selección de menús	Evita errores de usuario Requiere teclear poco	Lenta para usuarios experimentados Compleja si hay muchas opciones	Sistemas de propósito general
Rellenado de Formularios	Introducción de datos sencilla	Ocupa espacio en pantalla Opciones del usuario no se ajustan a los campos	Control de Stock Préstamos personales
Lenguaje de Comandos	Poderoso y Flexible	Difícil de aprender Gestión pobre de errores	Sistemas operativos Sistemas de comandos y control
Lenguaje Natural	Accesible a usuarios casuales Fácil de ampliar	Requiere más tecleo Los sistemas de comprensión de LN no son fiables	Sistemas de recuperación de información

#### 6. ¿Qué es el modelo de Seeheim?

El modelo de Seeheim es un patrón de arquitectura de software introducido en 1983 para estructurar la interfaz hombre-máquina en un software interactivo. Seeheim es una ciudad alemana, donde llevó a cabo en noviembre de 1983 un taller organizado por Eurographics sobre "Sistemas de Gestión de usuario Interfaz de usuario" (Sistemas de Gestión de la interfaz).

Modelo de Seeheim es un patrón de diseño :

- Resumen: no especifica cómo hacer las diferentes partes y sus interconexiones utilizando construcciones disponibles en lenguajes de programación.
- Alto nivel: se aplica al software de análisis en un grano macroscópica, de nivel de módulo, la biblioteca o proceso. Hay modelos abstractos de nivel inferior para el análisis de nivel de grano microscópica de la función o clase.

## 7. ¿Cuáles son las pautas de Shneiderman para la utilización del color?

- Limitar el número de colores utilizados y ser conservador en la forma de utilizarlos
- Utilizar un cambio de color para mostrar un cambio en el estado del sistema
- Utilizar el código de colores para apoyar la tarea que los usuarios están tratando de llevar a cabo
- Utilizar el código de colores de una forma consciente y uniforme
- Ser cuidadoso al utilizar pares de colores

## 8. ¿Qué factores se deben tener en cuenta para presentar mensajes al usuario?

- Contexto: Generar mensajes relacionados con la actividad actual del usuario.
- Experiencia: Proporcionar 2 tipos de mensajes: Cortos y concisos, largos y significativos.

- Nivel de Habilidad: El mensaje se debe adaptar a diferentes clases de usuarios.
- Estilo: Los mensajes deben ser: positivos y activos, formales, concisos, uniformes y constructivos. Nunca: negativos, pasivos, insultantes.
- Cultura: Adaptado a la cultura del país donde se utilizará el sistema.

## 9. ¿Qué es la usabilidad? Explique los atributos de usabilidad.

Medición cualitativa de la facilidad y eficiencia con la que un humano emplea las funciones y características que ofrece el producto de alta tecnología.

- Aprendizaje: Tiempo que tarda un usuario nuevo en ser productivo con el sistema
- Velocidad de Funcionamiento: Respuesta del sistema a las operaciones de trabajo del usuario
- Robustez: Tolerancia del sistema a errores de usuario.
- Recuperación: Cómo se recupera el sistema a los errores del usuario
- Adaptación: El sistema se adapta a único modelo de trabajo

## 10. Evaluación de interfaz

- Cuestionarios de la opinión de los usuarios sobre la interfaz
- Observación de los usuarios cuando usan el sistema
- Videos del uso típico del sistema

Incluir código en el software que recopila información de los recursos utilizados y errores comunes

## 11. ¿Qué es un Sketch?

Pensemos en el Sketch como el primer boceto que se realiza sobre el proyecto digital que queremos realizar. Son aquellos trazos (primeras pinceladas) sobre una hoja de papel, un tablero o –incluso– en una servilleta.

El Sketch refleja las ideas generales sobre el proyecto: ¿Qué áreas de contenidos y servicio queremos presentar en el proyecto? ¿Dónde estará la zona de navegación? ¿Dónde se cargan los sistemas de ayuda para usuarios? ¿Se agregaron servicios de redes sociales? Entre otras preguntas.

El Sketch no tiene un trabajo conceptual muy extenso. Aquí prima la creatividad, la experiencia y el deseo del diseñador.

## 11. ¿Qué es un Wireframe?

El siguiente paso en el diseño de un proyecto digital (al menos desde su componente gráfico y arquitectura de información) es la creación de los wireframe.

Para comprender el concepto, pensemos en una representación básica (en escala de grises) del proyecto, donde se definen con mayor precisión:

- Zonas de contenido
- Uso de elementos HTML (marcado semántico)
- Servicios de navegación y ayuda
- Flujos de navegación (cómo se conectan las unidades de información)

## 12. ¿Qué es un Mockup?

El Mockup es una representación más avanzada del diseño gráfico y comunicativo (desde una visión de navegación y AI) del proyecto. Aquí se integran elementos con mayor detalle, pues visualiza una aproximación de:

- Contenidos (pueden ser textos dummy)
- Paleta de colores, tomando como referente lo institucional, misional y el público objetivo del proyecto
- Declaraciones CSS

- Dimensiones de áreas de contenido y servicios
- Iconografía

Es de anotar que el mockup incluye los elementos del sketch y el wireframe. Cada uno es una evolución del anterior.

### 13. ¿Qué es un prototipo?

El prototipo es una representación de alto detalle sobre proyecto digital. En ella se puede identificar y operar:

- Sistemas de navegación
- Paleta de colores aplicada
- Iconografía
- Aplicación de las declaraciones CSS
- Experiencia de usuario
- Servicios de ayuda, búsqueda, interacción, entre otros

Este producto es el paso previo al desarrollo y presentación final del proyecto. El prototipo sirve para identificar, a partir de pruebas de usuario (beta-tester), las dificultades del proyecto.

## **TRABAJO PRÁCTICO N6**

### **Aplicaciones Web**

#### **1. ¿Qué es una aplicación web? De ejemplos de aplicaciones web.**

Software almacenado en una computadora o servidor web, que los usuarios pueden utilizar a través de Internet o de una Intranet, con un navegador web, para obtener los servicios que ofrezca.

Ejemplo: Webmails, Wikis, Tiendas en Línea, Ofimática, Dibujo, Juegos Online

#### **2. ¿Qué ventajas tienen las aplicaciones web? ¿Cómo se clasifican según el tipo de acceso?**

- Facilidad de acceso (sólo se necesita navegador web)
- Independencia de sistema operativo
- Utilizable desde cualquier plataforma (celular, tablet, PC)
- Facilidad de actualización y mantenimiento (sin redistribuir o instalar software)
- Bajo consumo de recursos

#### **Clasificación:**

- Públicas: Tiendas virtuales, diarios digitales, portales de internet, etc.
- Restringidas: Intranets, que ofrecen servicios para mejorar gestiones internas de una empresa (control hs de personal, gestión de proyectos o tareas), etc.

#### **3. ¿Qué diferencias existen entre las aplicaciones web y las páginas web?**

Una página web es un documento estático al que se accede a través de un navegador (la información sólo se puede leer, no interactuar con ella).

Una aplicación web es una página web especial, que tiene información sobre la que se puede interactuar e incluso cambiar.

#### **4) ¿Qué son las páginas web estáticas y dinámicas?**

Una aplicación Web es un conjunto de páginas Web estáticas y dinámicas.

- Web Estática: No cambia cuando un usuario la solicita. El servidor Web envía la página al navegador Web solicitante sin modificarla.
- Web Dinámica: Contiene elementos que permiten la comunicación activa entre el usuario y la aplicación, y cuyo contenido se genera a partir de lo que el usuario introduce.

#### **5) ¿Qué es la web 2.0?**

La Web 2.0 no es más que la evolución de la Web en la que los usuarios dejan de ser usuarios pasivos para convertirse en usuarios activos, que participan y contribuyen en el contenido de la red siendo capaces de crear, dar soporte y formar parte de sociedades y/o comunidades tanto a nivel local como global; que se informan, comunican y generan conocimiento y contenido.

La Web 2.0 es un concepto que se creó en 2003 y se refiere al fenómeno social surgido a partir del desarrollo de diversas aplicaciones en Internet. El término establece una distinción entre la primera época de la Web (donde el usuario era básicamente un sujeto pasivo que recibía la información o la publicaba, sin que existieran demasiadas posibilidades para que se generará la

interacción) y la revolución que supuso el auge de los blogs, las redes sociales y otras herramientas relacionadas.

## 6) ¿Qué lenguajes se utilizan generalmente para la programación web?

El mejor lenguaje de programación para empezar depende del tipo de proyectos que se quiera hacer.

- Programador de Servidores o Back-end: Python, Ruby, PHP, Java, .Net (o JavaScript con nodeJS). Conocimientos de bases de datos y de administración de sistemas.
- Programador de clientes o Front-end: HTML, CSS, Javascript. Conocimientos de diseño.
- Programador móvil: Objective C, Swift, Java (para Android). HTML/CSS para sitios web móviles. Conocimientos sobre servidores.
- Programador 3D o de videojuegos: C/C++, OpenGL, Animación. Conocimientos de diseño y artísticos.
- Programador de alto rendimiento: C/ C++, Java, conocimientos en matemáticas y análisis cuantitativo.

## 7) ¿Qué significan las siglas HTML? ¿Qué es HTML5? ¿Qué características tiene? Explique cada una.

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.<sup>1</sup>

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

### CARACTERÍSTICAS:

- Semántica: lo que le permite describir con mayor precisión cuál es su contenido.
- Conectividad: lo que le permite comunicarse con el servidor de formas nuevas e innovadoras.
- Desconectado y almacenamiento: permite a páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.
- Multimedia: permite hacer vídeo y audio de ciudadanos de primera clase en la Web abierta.
- Gráficos y efectos 2D/3D: permite una gama mucho más amplia de opciones de presentación.

- Rendimiento e Integración: proporcionar una mayor optimización de la velocidad y un mejor uso del hardware del equipo.
- Dispositivo de Acceso: admite el uso de varios dispositivos de entrada y salida.
- Styling: deja a los autores escribir temas más sofisticados.

#### **8) ¿Qué es una hoja de estilo o CSS?**

Hojas de Estilo en Cascada (CSS) es un lenguaje de estilo de hojas usado para describir la presentación de las páginas web escrito en HTML o XML.

CSS permite la separación del contenido del documento de la presentación del documento (disposición, colores, fuentes, etcétera).

#### **9) ¿Cómo se pueden agregar estilos en las páginas web? ¿Qué ventajas y desventajas tiene cada método?**

- Estilo en Línea
- Incrustando una Hoja de Estilo
- Enlazando una hoja de Estilo Externa

#### **10) Buenas prácticas se deben seguir a la hora de diseñar una página web**

En HTML:

- Cerrar las etiquetas
- Nombre de las etiquetas siempre en minúscula
- Poner comillas a todos los atributos

- Validar el código
- Indentar el código correctamente
- Tener en cuenta la accesibilidad (alt, title para img-label para inputs)
- No utilizar etiquetas deprecadas
- Usar ficheros externos para CSS y JavaScript

En css:

- Comprobar el diseño en varios navegadores
- Depuración (Ej. firebug)
- Comentar el código
- Ordenar y hacer el código fácil de leer
- Usar sistema común de nombrado
- Evitar tamaños absolutos en fuentes o elementos
- Utilizar notación de colores en hexadecimal
- Ordenar los elementos según pertenezcan a cabecera, contenido principal o pie de página

**11) Haga un listado de las etiquetas de HTML que ha empleado en los diversos ejercicios, y explique los atributos y la funcionalidad.**

- Header: Especifica un encabezado de un documento o sección.
- Nav: Define un conjunto de enlaces de navegación.
- Aside: Define un contenido que se coloca como una barra lateral.
- Section: Define una sección de un documento. (Ejemplo, introducción, contenido, contacto)
- Article: Específica contenido independiente. Debe tener sentido por sí mismo. (Ejemplo mensaje foro, entrada blog, artículo periódico)
- Footer: Especifica un pie de página de un documento o sección



## **TRABAJO PRACTICO N7**

### **Orientación a eventos**

**1) ¿Qué es la programación dirigida por eventos? De ejemplos de lenguajes orientados a eventos.**

La programación dirigida por eventos es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurrán en el sistema, definidos por el usuario o que ellos mismos provoquen.

Algunos lenguajes:

- JavaScript
- ActionScript
- Visual Basic
- C#
- Visual C++

**2) Explique qué es un evento. ¿Qué tipos de eventos hay?**

Es todo aquello que puede ser detectado y que pueda producir una reacción.

Ej. Hacer click en un botón, mover el puntero sobre un enlace, seleccionar un elemento de una lista desplegable.

#### **TIPOS DE EVENTOS:**

Eventos de Usuario

- Eventos del mouse
- Eventos del teclado

Eventos del navegador

- Manipulación de las ventanas del navegador
- Comunicación con el servidor

Eventos generados desde código

- Cambios en el contenido, aspecto o estructura
- Especificación DOM

### **3) ¿Qué es el event handler?**

Una función o método que contiene instrucciones de programa que se ejecutan en respuesta a un evento. Un manejador de eventos típicamente es una rutina de software que procesa acciones tales como pulsaciones de teclas y movimientos del ratón. Con los sitios Web, los controladores de eventos hacen que el contenido Web sea dinámico. JavaScript es un método común de scripting de manejadores de eventos para contenido web.

### **4) ¿Qué es JavaScript? ¿Qué características tiene?**

Es un lenguaje de scripting multiplataforma y orientado a objetos. Se usa principalmente del lado del cliente implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

- Orientado a Objetos
- Lenguaje Interpretado
- Tipado dinámico

### **5) ¿Qué herramientas permiten debuguear Javascript?**

Las herramientas de desarrollo de los navegadores cada vez son más potentes y van mejorando día a día. El primer gran debugger de JavaScript fue Firebug para Firefox, que hace unos 4 o 5 años no tenía ni la más mínima competencia, pero actualmente tenemos las

herramientas de desarrollo de Chrome y las de Internet Explorer (que han mejorado mucho en su versión número 11) como buena y completa competencia.

Las APIs de la consola han evolucionado y encontramos cosas tan interesantes como tablas de datos en la consola, algo que ayuda mucho a visualizar, sobre todo, respuestas por parte de un servidor.

## **TRABAJO PRACTICO N8**

### **Aplicaciones móviles**

**1) ¿Qué es una aplicación móvil? De ejemplos de aplicaciones móviles.**

Son aplicaciones diseñadas para ser ejecutadas en teléfonos inteligentes, tablets y otros dispositivos móviles y que permite al usuario efectuar una tarea, facilitando la gestión o actividad a desarrollar.

Ej: Whatsapp, Telegram, Twitter, Opera mini

**2) ¿Qué lenguajes se utilizan para programar las principales plataformas móviles? (Android, iOS y Windows Mobile)**

#### **APPS ANDROID**

- Lenguaje: Java o Kotlin + Librerías Android
- Android Studio
- SO: Windows, Linux o Mac
- Emuladores de Dispositivos

#### **APPS IOS**

- Lenguaje: ObjectiveC - Swift
- XCode
- SO: Mac
- Emuladores de Dispositivos
- Para probarlo en dispositivo, pagar licencia de desarrollador

**3) ¿Qué diferencias hay entre aplicaciones nativas, web e híbridas? ¿Cuándo es recomendable utilizar una u otra?**

Las Aplicaciones Nativas son aquellas que se desarrollan para un determinado sistema operativo en el móvil. Se crean

exclusivamente para teléfonos móviles, debiéndose crear una para iOS, otra para Android y otra para Windows Phone.

VENTAJAS	DESVENTAJAS
Acceso completo al dispositivo, en software y hardware	Diferentes lenguajes de programación y habilidades según el sistema operativo
Mejor experiencia de usuario	Costos y tiempos de desarrollo altos
Visualización desde las tiendas de apps e integración con wearables	No SEO, no aparecería bien posicionada en búsquedas de Google en caso de que no tuviera web optimizada para móvil

En el caso de las Web App o Webs Mobile Friendly son más sencillas de desarrollar y se manejan desde un ordenador. Los desarrolladores no usan diferentes lenguajes de programación; por el contrario, las desarrollan en lenguajes conocidos como el Java Script o el HTML.

VENTAJAS	DESVENTAJAS
Código de programación reutilizable	La conexión a internet es 100% necesaria
Desarrollo más sencillo y de menor costo	Acceso limitado al hardware del dispositivo
No necesita instalación	Al no necesitar instalación pierde visibilidad en las tiendas.

Las aplicaciones híbridas, las cuales se desarrollan utilizando tecnologías web como HTML, JavaScript (no confundir con Java) y CSS, y que, normalmente, se ejecutarán en el navegador nativo del sistema, por lo tanto, aunque dependerá del framework utilizado, no pueden acceder a tantas funcionalidades del hardware del dispositivo ni a las librerías del sistema. Además de tener normalmente, un peor diseño y un rendimiento más bajo.

**5) ¿Qué es Cordova? ¿Qué otras alternativas existen como esta?**

Es un framework para desarrollar aplicaciones nativas dentro de un smartphone. Cuenta con muchas APIs de diversos dispositivos móviles. Permite desarrollar con HTML, JavaScript y CSS,

Otra alternativa es Bootstrap que es un framework web desarrollado por Twitter para crear interfaces que se adapten al tamaño de pantalla de diversos dispositivos.

Está basado en HTML5 y CSS3 por lo que es compatible con la mayoría de los navegadores web.

## 6) ¿Qué significa que un sitio sea responsive?

Diseño web adaptable, donde la apariencia de las páginas web se adapta al dispositivo que se esté utilizando para visualizarla.