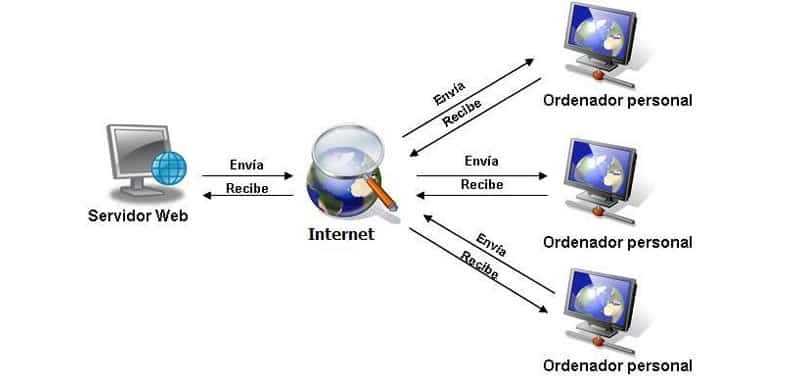
**Exercicis repas Tema 1,**

1. ***Diferencia entre archivo de texto y archivo binario:***  
     
   En general se podria decir que todos los archivos son binarios, sin embargo tienen una diferencia radical. En un archivo de texto cada serie de 8 bits corresponde a un equivalente en el codigo ASCII que puede ser una letra, un numero, un simbolo, etc. (Tambien depende el tipo de codificacion que se use para dicho archivo).  
     
   En cambio un archivo binario, el conjunto de bits tiene un arreglo y una longitud que puede variar de un archivo a otro, en inclusive de una longitud a otra. Dicho esto, podrias crear un archivo mp3 completito si supieras el orden exacto de cada uno de los bits que se necesitan para componerlo.
2. ***¿Qué es el código ASCII? Cuánto ocupa cada caracter:***  
     
   El código ASCII es un estándar para la representación de caracteres en cualquier dispositivo electrónico, haciendo referencia a una codificación fija que asigna a caracteres imprimibles como letras, números y signos de puntuación y a caracteres de control no imprimibles un código concreto, que al pasar por el dispositivo lo traduce y muestra el carácter deseado. Normalmente el código ASCII se extiende a 8 bits (1 byte) añadiendo un bit de control, llamado bit de paridad.
3. ***¿Que es Unicode?***  
     
   Unicode es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de numerosos idiomas y disciplinas técnicas, además de textos clásicos de lenguas muertas.
4. ***Las diferencias entre los siguientes formatos de codificación de caracteres:***  
     
   ***UTF-8:*** Tiene una ventaja cuando los caracteres ASCII son los más frecuentes. En ese caso, la mayoría de los caracteres sólo ocupan un byte cada uno. También es ventajoso que el archivo utf-8 que contiene sólo caracteres ASCII tenga la misma codificación que un archivo ASCII.  
     
   ***UTF-16:*** Es mejor donde no predomina el ASCII, utiliza 2 bytes por carácter principalmente. utf-8 empezará a utilizar 3 o más bytes para los caracteres de orden superior, mientras que UTF-16 se queda en sólo 2 la mayor parte del tiempo.  
     
   ***UTF-32:*** Cubrirá todos los caracteres posibles en 4 bytes cada uno lo que lo hace bastante hinchado.
5. ***¿Que ventajas tienen los archivos binarios? Y los archivos de texto?***  
     
   ***Archivo Binario:*** La principal ventaja es que es posible acceder a un elemento específico dentro del archivo sin la necesidad de procesar todos los elementos anteriores a él, como en el caso de los archivos de texto de acceso secuencial.  
     
   ***Archivo de texto:*** Son ideales para almacenar datos para exportar e importar información en cualquier dispositivo electrónico, su manipulación es más sencilla que la de los archivos binarios, y son directamente modificables, sin tener que acudir a software específico.
6. ***Porque el texto es el formato más versátil para exportar/importar datos?***  
     
   Los archivos de texto sólo son capaces de almacenar texto plano, es decir, texto sin formato alguno. Debido a esta universalidad, se pueden utilizar archivos de texto para que el propio texto sirva para almacenar otros datos, es decir, información que no es texto.
7. ***Que son los lenguajes de marcas y que usos tienen?***  
     
   Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. Se utiliza en las páginas y sitios web y fue creado por el padre de la web, Tim Berners Lee en 1989. El lenguaje HTML es el encargado de definir los contenidos de un sitio web de forma textual y estructurada. Dicho de otra forma, es el encargado de decirle al navegador web cómo debe visualizarse el sitio web.
8. ***¿Cuales son los 3 tipos de lenguajes de marcas que hay?***  
     
   Orientados en la presentación. En ellos los metadatos permiten indicar el formato en el que debe presentarse el texto. Es el caso de RTF, en el que las etiquetas especifican tipos de letra, tamaños de página, colores, etc.  
     
   Orientados a la descripción. En ellos las marcas especiales permiten dar significado al texto pero no indican cómo debe presentarse en pantalla lo mismo. Sería el caso de XML (o de SGML), JSON, Markdown y de las versiones actuales de HTML. En estos lenguajes simplemente se indica el significado del contenido.  
     
   Orientados a procedimientos. Se trata de documentos en los que el texto marcado, se interpreta como órdenes a seguir, y así el archivo en realidad contiene instrucciones a realizar con el texto (girarle, convertirlo en una fórmula, hacer una suma, etc. .). Es el caso de LaTeX o PostScript.
9. ***Explicación de los siguentes lenguajes:***  
     
   ***XML:*** Son las siglas de Extensible Markup Language, que podemos traducir como Lenguaje de Marcas Extensibles, aunque realmente es un meta-lenguaje. Normalmente, los archivos XML contienen información de cualquier tipo, ya sea información sensible o información de cualquier ámbito.  
     
   ***JSON:*** JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. JSON es de fácil lectura y escritura para los usuarios. JSON es fácil de analizar y generar por parte de las máquinas. JSON se basa en un subconjunto del lenguaje de programación JavaScript, Estándar ECMA-262 3a Edición - Diciembre de 1999.  
     
   ***HTML:*** El Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos.
10. ***El funcionamiento de un servidor WEB:***  
      
    El servidor web recibe la petición HTTP y la procesa a través de su servidor HTTP. Una vez que su servidor HTTP acepta la solicitud, buscará en los archivos del servidor para obtener los datos pertinentes. Después, el servidor web devuelve los archivos del sitio al navegador web que envió la solicitud.  
      
    
11. ***¿Qué diferencia hay entre un lenguaje de script de servidor y un lenguaje de script de cliente?***  
      
    El client-side rendering, también renderizado del lado del cliente, programación del lado del cliente o client-side scripting, es una técnica a la que recurren los desarrolladores web principalmente para realizar proyectos con contenido dinámico. En este caso, el servidor no ejecuta ni procesa los scripts; lo hace el navegador de acceso. Para ello, los scripts se insertan en el documento HTML o XHTML o se escriben en un archivo separado que se vincula al documento.  
      
    Cuando el usuario trata de acceder a una web con scripts del lado del cliente, el servidor web envía el documento HTML y los scripts al navegador. Este los ejecuta y presenta el resultado final. Los scripts del lado del cliente también pueden contener instrucciones concretas para el navegador web sobre cómo debe reaccionar ante determinadas acciones del usuario, por ejemplo, si este pulsa un botón. A menudo, el cliente no tiene que volver a establecer contacto con el servidor web.