**Teoría sobre la UT1 – Planificación de interfaces gráficas**

1. **Percepción visual**

Se trata del proceso mediante el cual el cerebro capta la información en forma de luz exterior y la transforma en una recreación propia. Al igual que hace los navegadores, cada cerebro hace su propia interpretación de la realidad. Si bien en la mayoría de los casos la interpretación es similar, es probable que haya personas o grupos que tengan una interpretación diferente.

1. **Elementos del diseño: Color, tipografía e iconos**

**Color:** Propiedad que tienen los objetos y que podemos percibir gracias a la luz. Ver tabla de colores HTML: <https://html-color-codes.info/codigos-de-colores-hexadecimales/>

**Tipografía:** El tipo de fuente que se utiliza en un diseño de interfaces. Se pueden utilizar distintos tipos. Ver repositorio de fuentes de Google: <https://fonts.google.com/> Existen muchos más y los veremos más adelante.

1. **Interacción persona-dispositivo**

Es una disciplina que trata de estudiar el intercambio de información entre ordenadores y personas. Tiene varios elementos principales:

* El usuario – Alguien con capacidades y limitaciones para procesar la información.
* El dispositivo y los periféricos.
* El proceso que se lleva a cabo.

1. **Guías de estilo y elementos**

Una **guía de estilos** indica las fuentes, los tamaños de texto y los colores principales y secundarios que aplicaremos a lo largo del sitio o aplicación.

**Elementos:**

* Marca de la empresa o propósito.
* La tipografía.
* Los colores.
* Los iconos.
* El tono (lenguaje).
* Las imágenes.
* El estilo de los controles de formulario.
* Los elementos de la interfaz (listas, tablas, menús…).

**CMS (Gestores de contenido)**: WordPress, Joomla!, Drupal… (Ver Tutoriales – Definiciones Web). Hacen que la máxima preocupación del propietario sea la creación de contenido en vez del desarrollo desde 0 del sitio web. Algunos Gestores de Contenido permiten personalizar en profundidad el código.

**Plataformas de generación automática de sitios web**: Facilitan el desarrollo de páginas web a las personas que no poseen conocimientos. La desventaja es que no proporcionan el control total sobre el código del sitio, aunque cada vez van avanzando más en este campo. Ver ejemplos: Wix, Weebly, Squarespace…

1. **Componentes de una interfaz web.**
   * + Cabecera con metadatos.
     + Un conjunto de componentes. Entre ellos:
       - Textos.
       - Listas.
       - Formularios.
       - Imágenes.
       - Vídeo.
       - Audio.
     + Un layout o disposición de esos componentes (Irá dentro del <body> de la web).
2. **Aplicaciones para desarrollo web**. Definición de IDE. Versiones más extendidas. (Ver Tutoriales – Editores y entornos de trabajo). Aspectos a valorar: multiplataforma, código abierto, gratuito y libre.
   1. **Entornos pesados**:
      * Netbeans.
      * Eclipse.
      * Webstorm.
      * VisualStudio.
      * IntelliJ IDEA.
   2. **Entornos ligeros**:
      1. Visual Studio Code.
      2. Notepad++.
      3. Sublime.
      4. Atom.
      5. Brackets.
3. **Lenguaje de marcas HTML5** (Ver documento con parte práctica).
4. **Mapas de navegación**: Sitemaps – Ver tutoriales con definición y web <https://www.xml-sitemaps.com/>
5. **Maquetación web y elementos de ordenación.**

En los 90 se utilizaban tablas HTML para maquetar el contenido de una página web, sin embargo, esta tecnología ha quedado completamente desfasada. En la actualidad nos apoyamos en CSS3 para realizar la maquetación de tal manera que se adapte a cualquier dispositivo. Uno de los objetivos más importantes que se buscan conseguir hoy en día es que una web sea totalmente adaptable (responsiva) al dispositivo en el que está siendo visualizada.

**Flujo de un documento web**: Lo normal es posicionando los elementos de izquierda a derecha y de arriba a abajo.

Distinción entre **elementos de bloque y elementos inline**:

* **De bloque**: Ocupan todo el ancho de página. El siguiente elemento se posicionará debajo de éste a la izquierda. Ejemplos: <p>, <div>
* **Inline**: No ocupan todo el ancho de página (en principio), por lo que el siguiente elemento se posicionará a la derecha de éste. Ejemplos: <span> <img>

**Elementos flotantes**

El flujo normal de un documento web se puede alterar haciendo que los elementos floten. Para esto hay que aplicar la propiedad adecuada en CSS3 (float), por lo que el elemento se posicionará en el lado que se indique (izquierda o derecha). Todo esto lo veremos más adelante en la UT dedicada a CSS3, pero ahora mismo veremos un ejemplo rápido.

**Posicionamiento absoluto y relativo**

**Absoluto**: el elemento toma como referencia, para posicionarse, la ventana del navegador.

**Relativo**: el elemento se posiciona dependiendo de otros elementos padre con posición relativa.

**Fijado**: el elemento queda fijado en una posición e incluso se mantiene así aunque nos desplacemos hacia abajo en la página.

Tenemos más posibilidades de posicionamiento, pero las veremos en detalle en la UT dedicada a CSS3.

**Dimensiones absolutas y relativas**

**Unidades absolutas**: Se aplican siempre igual, independientemente de la pantalla o resolución. No se suelen usar o deberíamos evitarlas para hacer mejores diseños responsivos.

Ejemplos:

* **px** (píxeles)
* **pt** (puntos)
* **pc** (picas)
* **mm** (milímetros)
* **cm** (centímetros)
* **in** (pulgadas).

**Unidades relativas**: se ajustan a cada tipo de dispositivo. Se suelen utilizar este tipo de unidades.

Ejemplos:

* **ex**: Representa la altura del caracter x en la pantalla. Es relativa al tamaño base del navegador (16 píxeles).
* **em**: Representa la anchura del carácter M en la pantalla. Suele ser el doble de la dimensión ex. Se muestra relativo al elemento padre. Ejemplo: si el elemento body tiene un tamaño de fuente de 16px y un elemento hijo tiene una fuente con tamaño 1.3em, este texto se mostrará de un tamaño un 30% más grande que el del body (20.8px), mientras que si dentro de ese elemento tenemos otro hijo con un *font-size* de 1.3 em, el tamaño de fuente de este objeto sería un 30% más grande que el tamaño de su padre (27.04px). No recomendable con elementos del layout porque si cambiamos el tamaño del elemento padre, se puede trastocar el diseño.
* **rem**: similar a em, pero toma como referencia el elemento raíz del documento (root em, es decir en basado en raíz). Por defecto tiene un tamaño de fuente de 16px. Recomendable en elementos del layout que requieran medidas fijas, ya que, si cambiamos otros tamaños, estos no se van a ver afectados. Es muy útil con media queries.
* **%:** Representa el porcentaje respecto al tamaño total de la pantalla.

Muchos más detalles de las unidades CSS en <https://www.w3schools.com/cssref/css_units.asp> Esto lo veremos en su correspondiente UT.

1. **Iniciación a un entorno de trabajo de diseño de páginas web**: Bootstrap 4 (Ver documento con parte práctica).
2. **Herramientas de desarrolladores en navegadores**.

Son herramientas incorporadas en el navegador que permiten al desarrollador inspeccionar los elementos de una página web para poder tomar decisiones a la hora de modificar su código y aplicar posibles mejoras o correcciones.

1. **Web semántica**:

Se basa en mejorar la web clásica, de tal forma que la estructuración de la información sea mucho más clara y la búsqueda de información sea más precisa.

Hoy en día se da mucha importancia al SEO (Search Engine Optimization) para mejorar el posicionamiento de la web en los buscadores. En esto juega un papel muy importante la implantación de la web semántica, que va a permitir que a cada parte de la página sea posible identificarla con un tipo de contenido.

Otra de las mejoras que supone la implantación de la web semántica es que permite adaptar la web para hacerla accesible a aplicaciones que se dediquen recabar datos.