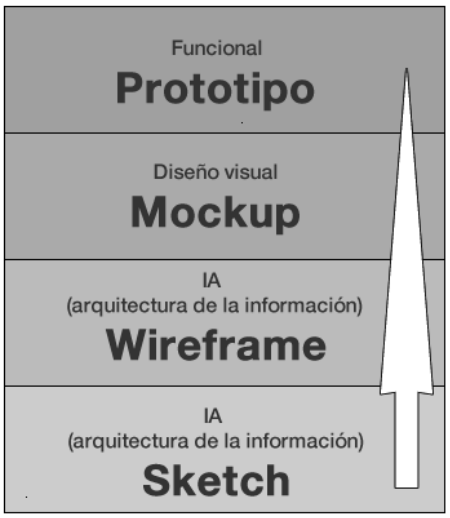
HITO 1 | MIGUEL RODRÍGUEZ POLO | DIW | 2º DAW

CUESTIÓN 1 A) En el desarrollo de productos y/o aplicaciones, ya sean para la web , como otro tipo de aplicaciones informáticas, hay una serie de fases que conviene seguir. De forma general los pasos a seguir serán:

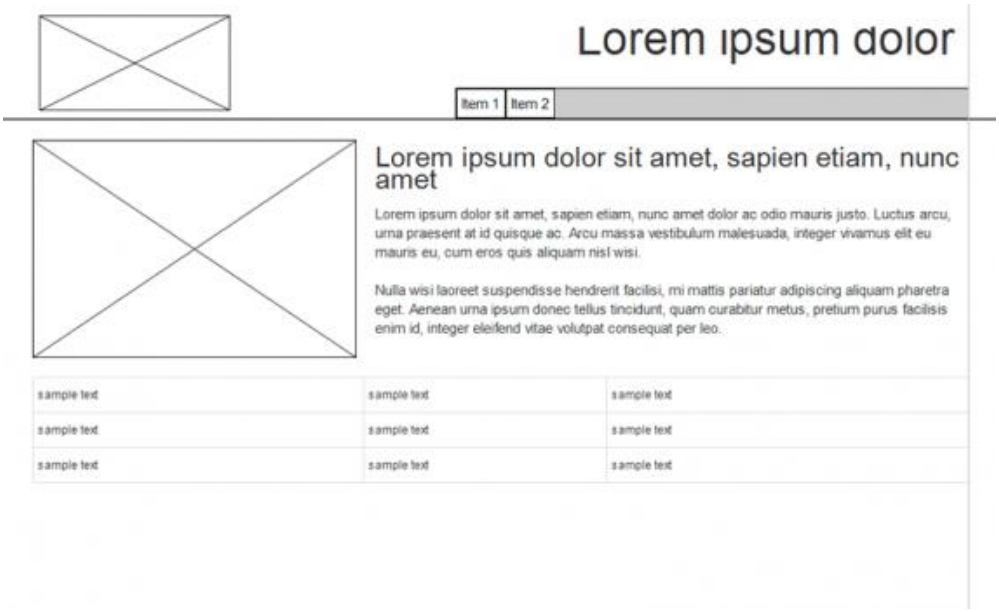
1. Sketch
2. **Wireframe**
3. **Mockup**
4. **Prototipo**



De esta lista, nos interesan principalmente 3: **Wireframe, Mockup y Prototipo.** Darle un vistazo a cada uno seguramente sea la mejor forma de visualizar sus diferencias.

WIREFRAME

Wireframe es una ilustración bidimensional de la interfaz de una página o una aplicación que se centra específicamente en la asignación de espacio y priorización del contenido, las funcionalidades disponibles, y los comportamientos deseados. Por estas razones, los wireframes normalmente carecen de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de contenidos. En otras palabras, se centra en “qué hace la pantalla, no cómo se ve”.

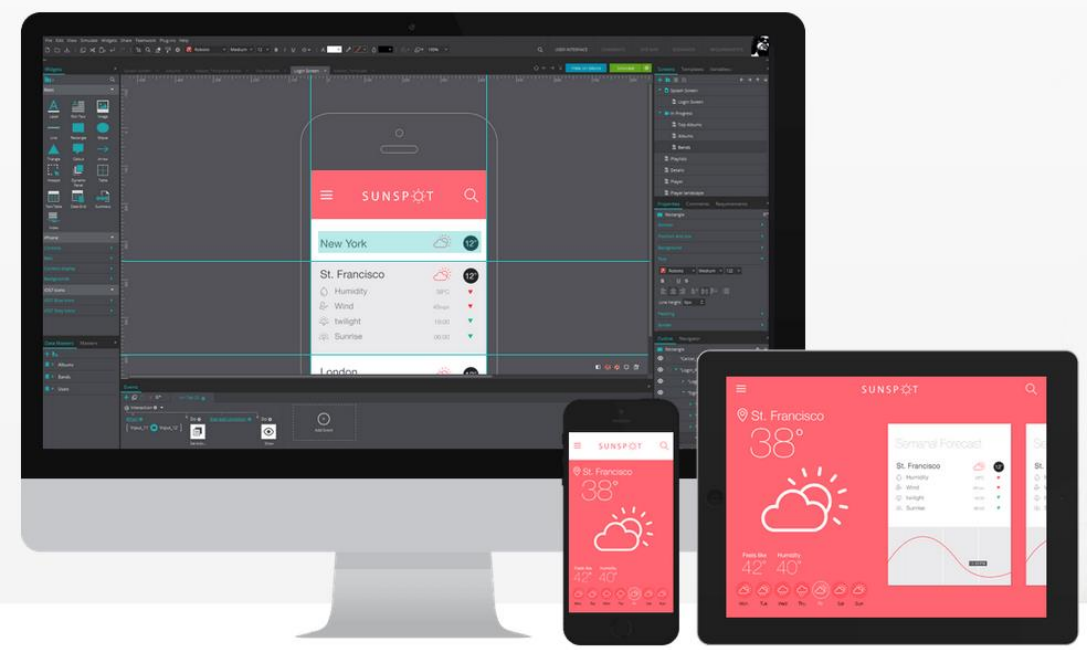


MOCKUP

El Mockup es una representación más avanzada del diseño gráfico y comunicativo de un proyecto. Hay dos escuelas de pensamiento que se diferencian en como conciben la forma de crear un MockUp, aquellos que creen que la maqueta debe representar el producto final exactamente (alta

fidelidad), y aquellos que ven la fase de maqueta como más transitorio y no debe tomar demasiado tiempo (media fidelidad).

Podríamos decir que generalmente es, una composición gráfica completa que ha utilizado el wireframe como plantilla introduciendo todos los elementos gráficos y visuales, convirtiéndose así en un modelo a escala de un producto que se utiliza para demostrar y probar un diseño. El mockup es un medio de representación de la apariencia del producto, y muestra los fundamentos de su funcionalidad. Los MockUp incluyen los detalles visuales, tales como colores, tipografía, etc., y son generalmente estáticas. Al observar un mockup, se debe tener una buena idea de cómo se verá el producto final y una idea aproximada de cómo podría funcionar (incluso si las funciones aún no se han desarrollado).



PROTOTIPO

Un prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

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

- Sistemas de navegación
- Paleta de colores aplicada
- Iconografía
- Experiencia de usuario
- Servicios de ayuda, búsqueda, interacción.
- Otros elementos del proyecto

La creación del prototipo debe llevarse a cabo cuando vamos a evaluar la interacción, y sirve para definir aspectos que no quedan claramente reflejados en un boceto de papel o un wireframe no navegable.

Los prototipos son navegables, por lo que sirven para testear elementos de interacción como estados «encima» de botones, validación de formularios, iconos, o cualquier elemento con el que el usuario interactúe. A través de la creación de prototipos, identificamos y solucionamos problemas UX como pueden ser la transición desde la página principal a los resultados de búsqueda sin recargar al usuario con demasiada información. Nos sirve como modelo del comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos.

En definitiva, cada una de estos conceptos es una de las fases del diseño de interfaces, donde lentamente se va avanzando hasta ir obteniendo un resultado similar al final.

B) El link del diseño es este: Link al diseño

CUESTIÓN 2 A) Cuestion2\index.html Aquí se encuentran los 3 primeros puntos de la cuestión

LA PRECEDENCIA DE ESTILOS

La precedencia de estilos en css, determina la prioridad donde los estilos son cargados en el html. Esto es necesario cuando hay ambigüedades, sobre dos elementos que deben recibir distintos e incompatibles atributos. Para resolver los conflictos, existe una jerarquía para asignar cuál se asigna:

(La lista va de más prioridad a menos prioridad)

1. Anotación `!important`, la cual anula cualquier tipo de conflicto
2. Anotación `style`, incluida en el html base.
3. Media type: Una propiedad aplica a todas las media types, a menos que un media type específico CSS esté definido
4. Definido por el usuario: La mayoría de los navegadores ofrecen opciones extras de accesibilidad.
5. Especificidad del selector: Los selectores definidos, como los id's o las clases.
6. Orden de las reglas: La última regla especificada tiene más prioridad
7. Herencia: Si una propiedad no tiene atributos, heredará los atributos de su padre
8. Definición base de la propiedad: Reglas básicas de CSS para HTML
9. Definición del navegador: La prioridad más baja. Se aplica cuando a un elemento le falta un atributo importante, como el tipo de fuente.

Ejemplos:

```
p{
    background-color: aqua !important;
}
```

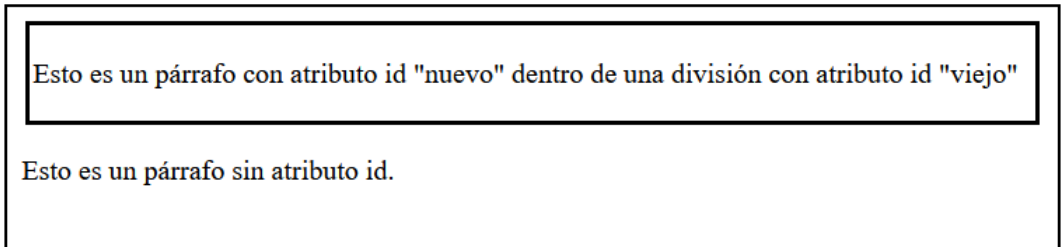
Esto ocurrirá siempre

El ejemplo siguiente muestra también dos selectores con un número de atributos id diferente (2 o 1). Puede comprobarse que se aplica siempre el selector con más atributos id, independientemente del orden en que aparezcan en la hoja de estilo.

```
div#viejo {
    border: black 3px solid;
    margin: 2px;
    padding: 2px;
  }

  div p#nuevo {
    color: red;
  }

  div#viejo p#nuevo {
    color: black;
  }
```



B) Display es un atributo que permite comunicarle al navegador como quieres tratar ese elemento. Se puede convertir en caja, texto, malla, tabla y otras opciones. Si se asigna como none, el elemento no se cargará en pantalla, por lo tanto, no consumirá recursos ni ocupará espacio entre los elementos.

Visiblitiy es un atributo que controla el aspecto visual de un elemento. Puede usarse para dos objetivos: Esconder elementos o plegar tablas. Si se esconde un elemento con él, no se perderá su espacio entre los elementos, a no ser que se utilice position:absolute.

Position es un atributo que controla la posición visual de un elemento, permitiendo mover los elementos o trabajar con ellos de otra forma. Es poco habitual usar este atributo para esconder elementos.

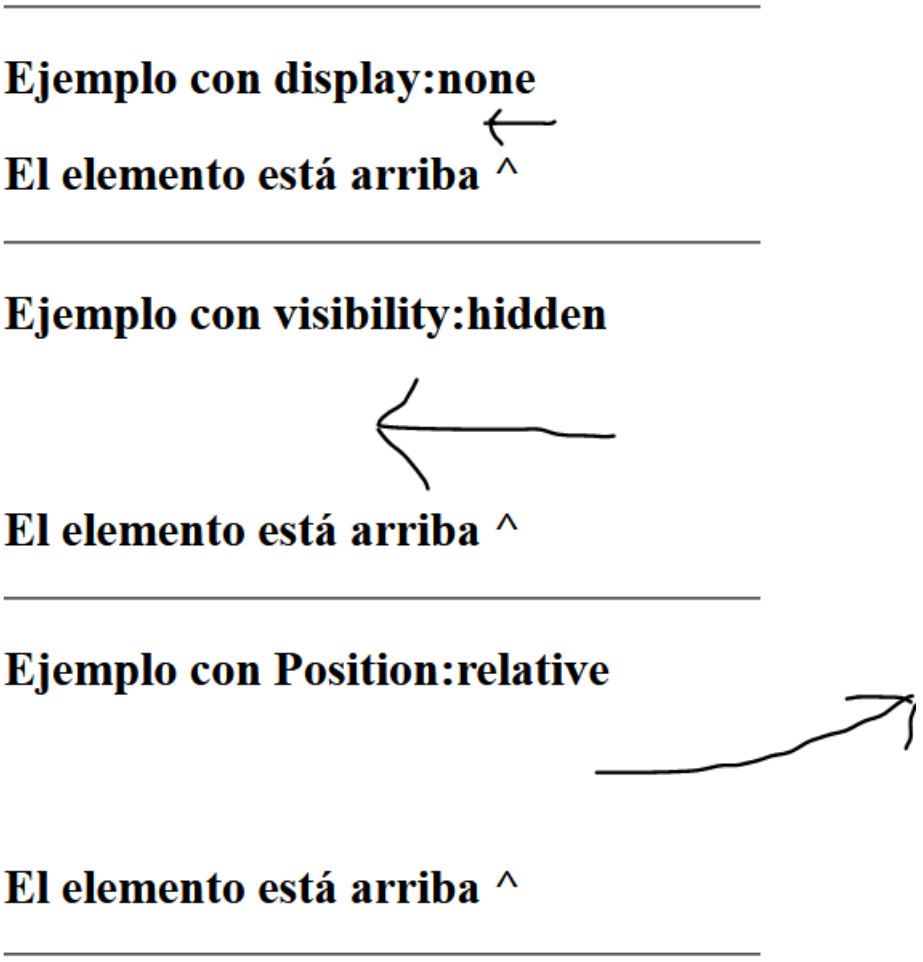
Ejemplos:

```
h1.display{
display: none;
}

h1.visibility{
    visibility:hidden
}

h1.position{
    position: relative;
    top: 400px; left: 400px;
}
```

```
<hr>
<h2>Ejemplo con display:none</h2>
<h1 class="display">H1</h1>
<h2>El elemento está arriba ^</h2>
<hr>
<h2>Ejemplo con visibility:hidden</h2>
<h1 class="visibility">H1</h1>
<h2>El elemento está arriba ^</h2>
<hr>
<h2>Ejemplo con Position:relative</h2>
<h1 class="position">H1</h1>
<h2>El elemento está arriba ^</h2>
<hr>
```



En el caso de Position:Relative, el elemento está fuera de los límites de la página. Sin embargo, su espacio de elemento sigue estando ahí

CUESTIÓN 3 A)

PREPROCESADORES CSS

Los procesadores son herramientas que nos permite escribir un pseudocódigo CSS que después se convertirá en código CSS, el cual usaremos para los estilos principales de nuestros proyectos web.

A la hora de utilizar un preprocesador, la ventaja que tenemos es que transformamos nuestro código CSS, pudiendo facilitar en gran medida el trabajo, ya que implementamos variables, funciones, cálculos, bucles, etcétera.

Todo esto hace que se pueda lograr una maquetación y un diseño bastante completo en nuestras maquetas web, además de forma más rápida.

Algunos de los procesadores más conocidos son Sass, Less, Stylus y PostCSS.

B) Cuestion3\index.html Aquí puedes encontrar el resultado. He usado Less

C) El estilo se aplica con el archivo css que se genera con tras compilar el archivo les. Esto ocurre porque el navegador no sabe interpretar el archivo less. Primero debe ser compilado y transformado en un archivo css.