

Rebuild a website using a CSS Framework

Vuetify es un framework que combina la potencia del popular VueJs con la estética de Material Design. Permite acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes “listos para usar”.



Los elementos que he usado para recrear en la mayor medida posible el layout dado, son varios, lo que más he intentado utilizar son grids.

`<v-flex>` que nos permite seleccionar las columnas (de 12 posibles) que queramos para posicionar nuestros elementos y además es responsive.

```
<v-flex xs12 sm6 >
```

Name

Al comenzar a recrear la página, lo primero que tenemos que crear es el Toolbar. Elemento que también es responsive, nos sirve para navegar por dentro de nuestra página web. `<v-toolbar>`

```
<v-toolbar class="grey lighten-2" flat height = "120">
```

Con **class="grey lighten-2"** lo que estoy haciendo es seleccionar el color que quiero que tenga de fondo. **"Flat"** remueve la sombra que tiene por defecto y con **height**, evidentemente, le asigno una altura determinada.

Freeze

[HOME](#)

[ABOUT](#)

[BLOG](#)

[CONTACT](#)

Después también tenemos “<router-link to="/">”

Esto nos sirve para poder insertar enlaces en los elementos que queramos, dentro de nuestra página. En mi caso, necesitaba poner que si el user hace click en el logo de “Freeze” dentro del Toolbar, nos lleve al Home.

```
<router-link to="/">
  <v-img v-for="image in image" :key="image.image">
    
  </v-img>
</router-link>
```

Esta parte, está dentro de otro elemento que hemos usado bastante, que es <v-layout>

Nos sirve para separar contenido, en nuestro caso es útil porque por un lado tenemos el logo y por el otro, una especie de menú creado por Links.

```
<v-layout justify-space-around>
  <router-link to="/">
    <v-img v-for="image in image" :key="image.image">
      
    </v-img>
  </router-link>
</v-layout>
```

Ahora pasamos a lo que sería el **Body**, el elemento que más he usado es

<v-container>

Nos ha servido para poder centrar el contenido y delimitar los elementos dentro del mismo, sin esto, todos lo que se añada a <v-layout> empezará desde la izquierda sin margen alguno.

```
<v-layout row wrap>
  <v-flex class="backCol">
    <v-container>
      <h1 class=" mt-2 title brown--text text--lighten-1">NEW PRODUCT</h1>
      <h1 class=" mt-2 change-font pink--text">The Twist of Healthy Yogurt</h1>
      <p class=" maxWid body-1">This website template has been designed by freewebsitetemplates.com for you, for
      </p>
      <v-spacer></v-spacer>
      <v-layout class="imageSth" justify-space-between>
        <v-img v-for="image in image" :key="image.image">
          
        </v-img>
      </v-layout>
    </v-container>
  </v-flex>
</v-layout>
```

NEW PRODUCT

The Twist of Healthy Yogurt

This website template has been designed by freewebsitetemplates.com for you, for free. You can replace all this text with your own text.



y así quedaría sin `<v-container>`, la diferencia es notable.

NEW PRODUCT

The Twist of Healthy Yogurt

This website template has been designed by freewebsitetemplates.com for you, for free. You can replace all this text with your own text.



Para terminar con los elementos más notables de la página, podríamos decir que `<v-avatar>` es uno de los más útiles, ante la dificultad de seleccionar un solo icono de una misma foto, he decidido, utilizar este elemento. Usado para controlar el tamaño y border radius de imágenes responsive, es escalable.

```
<router-link to="About">
  <v-avatar class="blockIcons">
    
  </v-avatar>
  <v-avatar class="blockIcons">
    
  </v-avatar>
  <v-avatar class="blockIcons">
    
  </v-avatar>
  <v-avatar class="blockIcons">
    
  </v-avatar>
</router-link>
```

creamos un avatar para cada una de nuestras fotos, con el mismo tamaño, como hemos puesto **Router**, todas tienen un link hacia la página de Contact.



PRODUCTS



OUR STORY



FLAVORS



OUR LOCATION

Problemas encontrados

Los principales problemas que he encontrado son a la hora de cambiar la configuración por defecto.

He tenido que buscar durante largo rato en internet, al final la solución más óptima y rápida que he encontrado es modificar la variable de stylus.

Lo primero que tenemos que hacer es ->

```
npm i stylus stylus-loader style-loader css-loader --save-dev
```

Para finalizar, aplicamos estilo CSS, como por ejemplo, para cambiar la font-family y se le pasa como class="NombreClase".

```
<style lang="stylus" scoped>
.iconText00 {
  font-family: "Lato", Sans, serif;
  font-size: 30px;
  font-weight: normal;
}
</style>
```

BOOTSTRAP VS VUETIFY



VS



En mi opinión, Bootstrap 4 es bastante mejor para este tipo de proyectos donde tienes que recrear una pequeña página web.

La principal diferencia, en mi opinión es que la curva de aprendizaje con Vuetify es mayor, necesitas mucho más tiempo para poder encontrar cosas que con Bootstrap sería muy rápido, la gran cantidad de tutoriales que hay y documentación, con Vuetify no los tienes.

Lo mejor que tiene vuetify es que la estructura que tiene es muy intuitiva y está muy bien pensado si tienes que añadir grandes cantidades de elementos similares a la vez, tienes for....

```
<v-img v-for="image in image" :key="image.image">
  
</v-img>
```

```
<script>
export default {
  data () {
    return {
      image: [
        {logo: "/logo.png"},
        {back: "/bg-home.jpg"},
        {yogurt: "/yogurt.jpg"}
      ]
    }
  }
}
</script>
```