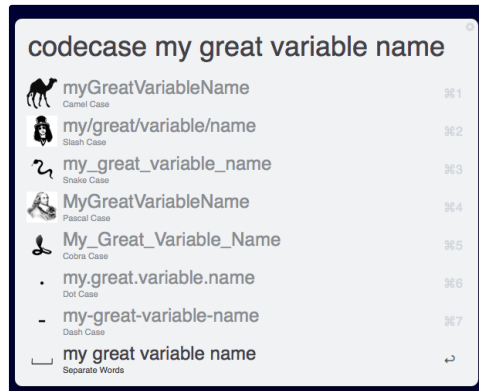


## Maquetación en Nextjs

### General [↗](#)

Piensa que todas **las opciones que describiremos a continuación** y en otros apartados **se pueden utilizar simultáneamente**, no se restringuen entre sí, eso sí, habrá casos en los que por la naturaleza del proyecto sea más útil aplicar una estrategia u otra.

La forma de crear contenidos html, es escribiendolo previamente en jsx, es una notación muy similar al html con sus etiquetas características en **camelCase**.



**Si no tienes ganas de pensar** mucho te dejo esta web donde te realiza una conversión instantánea. [↗ HTML to JSX](#) , para contenido plano es mejor utilizarla sin marcar el tick de creación de clase.

## HTML to JSX Compiler

☐ Create class

Live HTML Editor

```
<div class="awesome" style="border: 1px solid red">
  <label for="name">Enter your name: </label>
  <input type="text" id="name" />
</div>
<p>Enter your HTML here</p>      <div class="box-cajetin-borde-azul" id="cajaClaimPrincipal">
  <div class="subtext-banner">
    <span style="font-family: GutthenBloots; font-size: 40px;"></span>
    Tu hogar protegido con nuestra alarma, ahora a un súper precio.
  </div>
  <div class="subtext-banner-reducido">Contrata ya y <strong>ahorra 240€!</strong> </div>
  <div class="subtext-banner-reducido"><span class="text-banner-bold"></span></div>
  <div class="bandaPromoBanner">
    <span class="txt-caja-bold">INSTALACIÓN GRATUITA. 0€ DE ALTA</span>
  </div>
  <div class="subtext-banner" style="font-size: 16px; color: #ffffff; margin-top: -15px; margin-left: 60px;">
</div>

<div>
  <div className="awesome" style={{border: '1px solid red'}}>
    <label htmlFor="name">Enter your name: </label>
    <input type="text" id="name" />
  </div>
  <p>Enter your HTML here</p>      <div className="box-cajetin-borde-azul" id="cajaClaimPrincipal">
    <div className="subtext-banner">
      <span style={{fontFamily: 'GutthenBloots', fontSize: '40px'}} />
      Tu hogar protegido con nuestra alarma, ahora a un súper precio.
    </div>
    <div className="subtext-banner-reducido">Contrata ya y <strong>ahorra 240€!</strong> </div>
    <div className="subtext-banner-reducido"><span className="text-banner-bold"></span></div>
    <div className="bandaPromoBanner">
      <span className="txt-caja-bold">INSTALACIÓN GRATUITA. 0€ DE ALTA</span>
    </div>
  </div>
```

### Estrategias de maquetación [↗](#)

#### 1.1- Basada en componentes react

Bajo mi punto de vista personal, recomiendo encarecidamente emplear esta estrategia, ya que reutilizan componentes, que han sido **testados** y **cumplen con los requisitos** de: diseño y analítica.

Te ahorrarás escribir css que ya ha sido escrito anteriormente y mucha funcionalidad. Por ejemplo vemos que el siguiente código jsx:

```

<div className="caption">
  <div className="secondary-banner_kicker">
    <Icon id={icon} size={24} color={ICON_COLOR.BLUE} />
    <TextInfo
      fontType={TEXT_INFO_FONT_WEIGHT.BOLD}
      size={TEXT_INFO_SIZE.TITLE7}
      color={TEXT_INFO_COLOR.BLUE}
      text={title}
    />
  </div>
  <TextInfo
    className="secondary-banner_title"
    kind={TEXT_INFO_KIND.HEADING}
    level={TEXT_INFO_HEADING_LEVEL.H2}
    fontType={TEXT_INFO_FONT_WEIGHT.LIGHT}
    size={TEXT_INFO_SIZE.TITLE3}
    text={subTitle}
  />
  <TextInfo
    className="secondary-banner_description"
    kind={TEXT_INFO_KIND.PARAGRAPH}
    size={TEXT_INFO_SIZE.TITLE6}
    text={description}
  />
  <div className="secondary-banner_actions">
    {textButton && linkButton && (
      <CustomLink
        className="secondary-banner_button"
        href={linkButton}
        taggingInfo={taggingInfo}
        title={`Ir a ${textButton}`}
        aria-label={textButton}
      >
        <Button
          text={textButton}
          backgroundColor={BUTTON_COLOR.SECONDARY_INVERSE}
          size={{ w: BUTTON_WIDTH.LARGE }}
        />
      </CustomLink>
    )}
  </div>
</div>

```

Generaría a su vez este código HTML (observamos que genera todas las clases CSS necesarias para su dibujo y aunque no pueda apreciarse, también su funcionalidad para los clicks y enlaces, asegurando sus métricas).

```

<section class="jsx-2151161111 secondary-banner secondary-banner--desktop secondary-banner--color-white secondary-banner--align-left" (rm) => #0
  <div class="jsx-2151161111 photo"></div>
  <div class="jsx-2151161111 caption">
    <span class="jsx-2042618462 typography typography--desktop typography--segment-g typography--variant-caption-bold typography--color-primary typography--align-left typography--font-size-14">
      <strong>Elige la fibra y el 5G que mejor se adapta a ti</strong>
    </span>
    <div class="jsx-2042618462 typography typography--desktop typography--segment-g typography--variant-h3 typography--color-black typography--align-left typography--display-4">
      <h3>YBlock secondary-banner title</h3>
      <p>Elige la fibra y el 5G que mejor se adapta a ti</p>
    </div>
    <p class="jsx-2042618462 typography typography--desktop typography--segment-g typography--variant-body-2 typography--color-black typography--align-left typography--display-block secondary-banner_description">La mejor red fija y móvil de Movistar que usa 100% energía renovable.</p>
  </div>
  <div class="jsx-2151161111 secondary-banner_actions">
    <a href="/particulares/movistar/conectividad" title="Ir a Descubre conectividad" aria-label="Descubre conectividad" class="jsx-1984608051 link link--variant-outline link--color-secondary secondary-banner_button">Descubre conectividad.</a>
    <a href="/particulares/movistar/configurador" title="Ir a Configura MiMovistar" aria-label="Configura MiMovistar" class="jsx-1984608051 link secondary-banner_link">Configura MiMovistar.</a>
  </div>
</div>
</div>
</section>

```

El cual el navegador entenderá a la perfección y dibujará el resultado.

## Paso 1. Conectividad

Elige la fibra y el 5G que mejor se adapta a ti

La mejor red fija y móvil de Movistar que usa 100% energía renovable

Descubre conectividad

[Configura miMovistar](#)

### 1.2.- Maquetación plana/directa

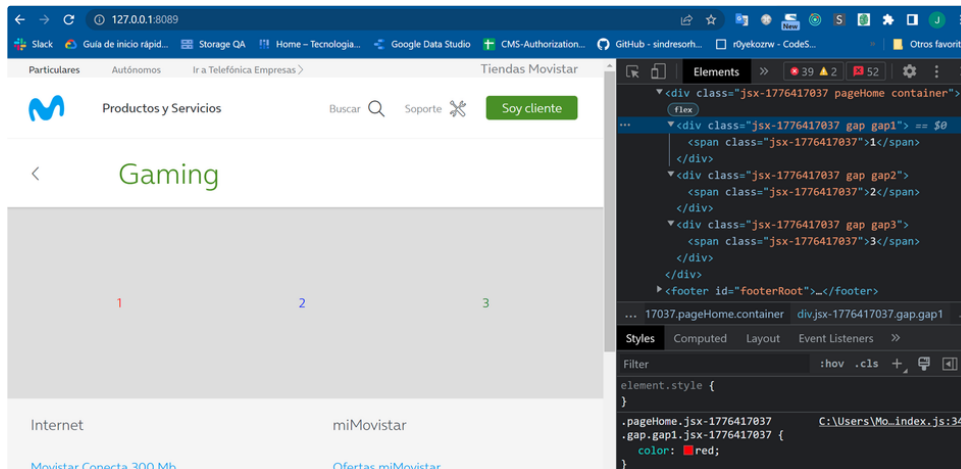
Toda página en nextjs tendrá el siguiente formato:

### 1.2.1.- Estilos jsx

```
const Home = () => {
  return (
    <Header />
    <div className='pageHome container'>
      <div className='gap gap1'>
        <span>1</span>
      </div>
      <div className='gap gap2'>
        <span>2</span>
      </div>
      <div className='gap gap3'>
        <span>3</span>
      </div>
    </div>
    <Footer />
  </style jsx>
  {`
    You, 44 seconds ago | 1 author (You)
    .pageHome{
      display:flex;
      min-height: 250px;
      background-color: var(--color-gp--grey3);
      width:100%;
      align-items: center;
      justify-content: space-around;
      You, 44 seconds ago | 1 author (You)
      .gap{
        &.gap1 { color: red }
        &.gap2 { color: blue }
        &.gap3 { color: green }
      }
    }
  `}
  </style>
);
};
```

Siempre o en la mayoría de los casos deberán estar presentes los componentes <Header /> <Footer />, estos ya harán su trabajo para mostrar la representación correspondiente.

1. El contenido, entre estos dos componentes será el que renderice/dibuje en la parte central de la pantalla.
2. Para asociar estilos a la página puedes optar por varias estrategias la que te presento en la parte izquierda es haciendolo en código que se denomina: estilos en jsx. **Podrás emplear tanto css, como scss**
3. Recuerda que siempre tendrás disponibles la paleta de colores y variables globales de la gerencia. En esta se basan todos los componentes para su unificación de valores.
4. Esto se encuentra en todos los proyectos en la ruta: /styles/globalVars.scss. Y se instancian a nivel de aplicación en pages/\_app.js  
  
*NO DEBES MODIFICAR ESTE CONTENIDO YA QUE SE ACTUALIZA POR DESCARGA PROGRAMÁTICA Y PISARÍA TUS CAMBIOS.*
5. Un ejemplo de consumo de estas propiedades lo podemos ver para background-color: var(--color-gp--grey3)



### 1.2.2.- Estilos css por modulo / página

Este método es más tedioso porque se necesita propagar los estilos por cada nodo del jsx. Para realizarlo se precisaría:

1. Dar un nombre al modulo en nuestro caso se llamará Home, **este nombre debe de coincidir con el nombre de la función exportada** en ese fichero. Podreis ver con como en index.js aparece la linea: "export default function Home ()"
2. **Crear bajo la carpeta styles un fichero asociado** a ese modulo. Recordar el formato: [name].module.css
3. **Importar y enlazar cada nodo con su respectivo style** accediendo a estos como si de un objeto js se tratase

6 minutes ago • Initial commit from Create

le.css'

[name].module.css

```
    t="Generated by create next app" />  
    .ico" />
```

```
    extjs.org">Next.js!</a>
```

```
    n!>
```

```
>pages/index.js</code>
```

```
    ocs" className={styles.card}>  
    2>
```

```
    n about Next.js features and API.</p>
```

```
    earn" className={styles.card}>
```

```
    n interactive course with quizzes!</p>
```

<https://github.com/vercel/next.js/tree/canary/examples>

erplate example Next.js projects.</p>

# ome to Ne

tarted by editing `pages/index`

ion →

formation  
atures and

Learn →

Learn about  
interactive c  
quizzes!

etwork

Performance

Memory

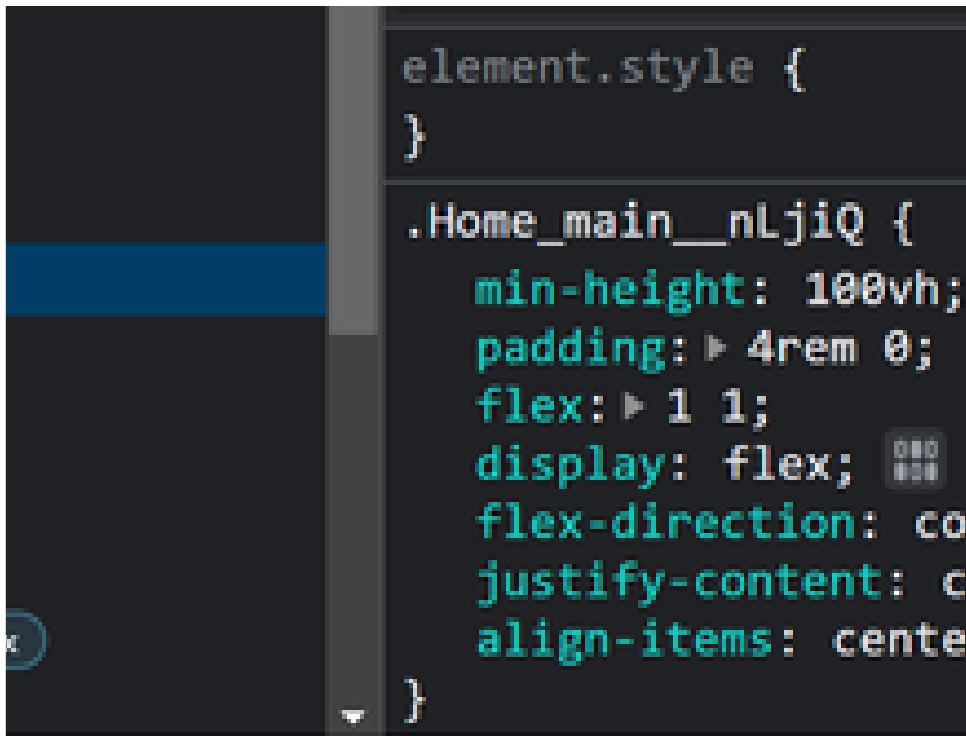
Ap

Styles

Computed

L

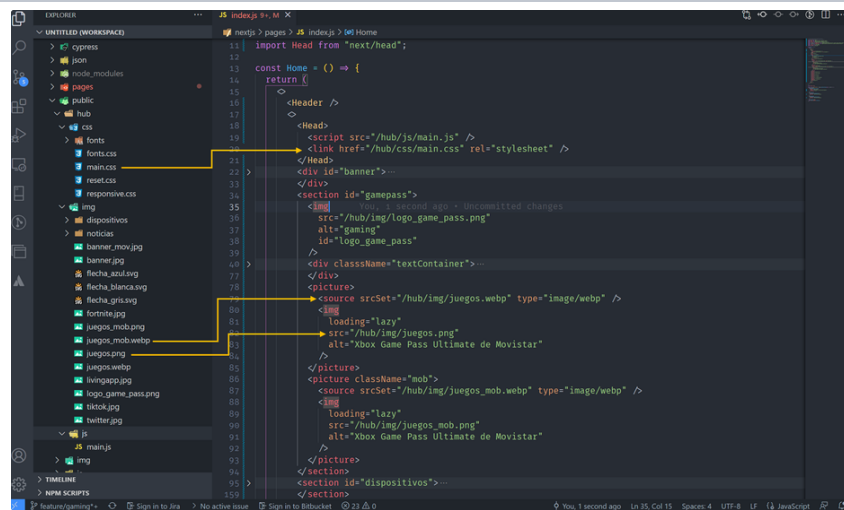
Filter



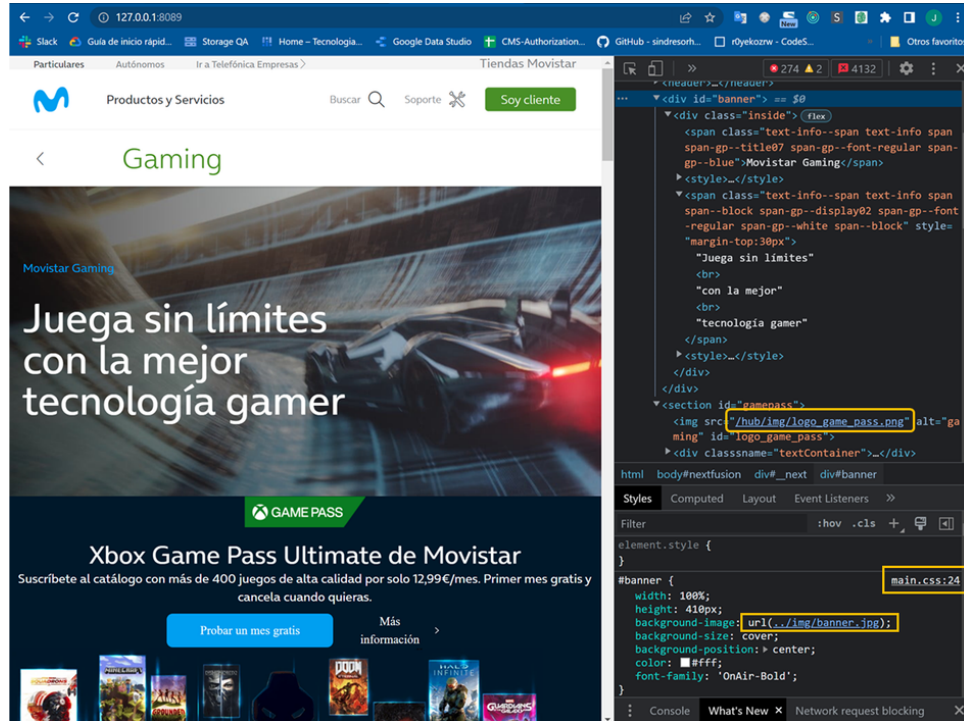
### 1.2.3.- Considerar ficheros css como estáticos de proyecto

Esta es una funcionalidad muy potente, ya que sirve para cualquier estático que se desee disponer de forma interna. En next, existe una carpeta especial llamada **public**, todo lo que se coloque en esa carpeta estará siempre disponible desde ruta absoluta desde **cualquier parte del proyecto**.

En este ejemplo nos estamos importando: una hoja de estilos, y varias imágenes. Incluido un fichero de lógica javascript (hablaremos de esto más adelante)



fichero que se transformará en `hub/index.html`





Inclusión de fichero main.css, junto con inyección de estaticos desde html y css

#### 1.2.4.- Css transversales: (reset.css, base.css o colors)

En caso de disponer de un css que se desee que aplique a nivel de aplicación, en todas las páginas, se podrá añadir en `_app.js`, tal y como tendrás `globalVars.css`

