



## Iconografía y CSS

<b>Iconografía y CSS</b>	<b>0</b>
<b>Control de versiones</b>	<b>2</b>
<b>Iconografía</b>	<b>2</b>
Estructura de carpetas en Minio	3
Iconos por defecto	4
Iconos personalizados	5
Características que deben cumplir los iconos	5
Cómo subir los iconos en la aplicación	6
Cómo personalizar iconos de subtipo de objeto para Anjana	7
<b>CSS</b>	<b>9</b>
Archivo index.scss	9
Material	10
Ng-zorro	10
Archivos de Material	11
Archivo styles.scss	11
Archivo styles_media.scss	11
Archivo variables.scss	11
Archivo typography.scss	14
Archivo icons.scss	14
Archivos de Ng-zorro	15
Archivo nz-buttons.scss	15
Archivo nz-datatable.scss	15
Archivo nz-form.scss	15
Archivo nz-modal.scss	15
Archivo nz-progress.scss	15
Archivo nz-skeleton.scss	15
Archivo nz-stepper.scss	15
Algunos aspectos generales	15

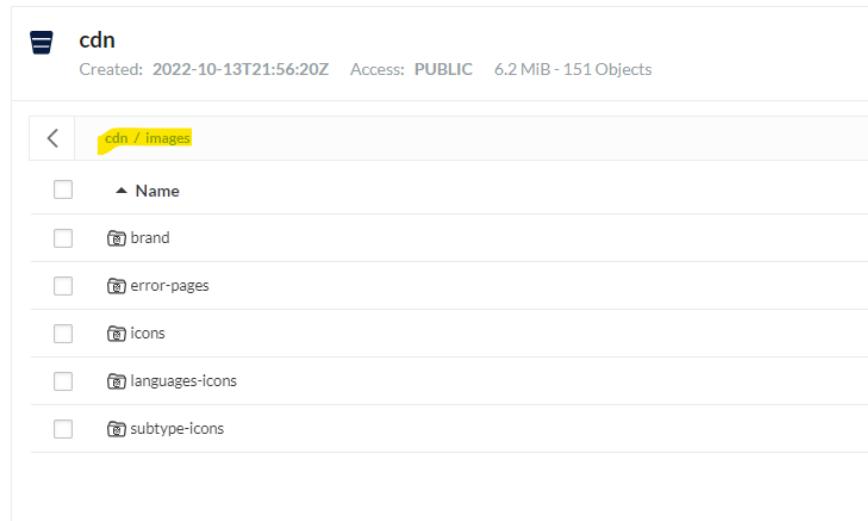
## Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Anjana Producto	Anjana Producto	Creación del documento

# Iconografía

## Estructura de carpetas en Minio

Dentro de la carpeta cdn/images tendremos una clasificación de iconos e imágenes según su finalidad.

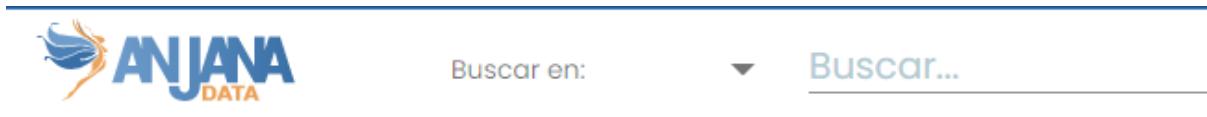


cdn  
Created: 2022-10-13T21:56:20Z Access: PUBLIC 6.2 MiB - 151 Objects

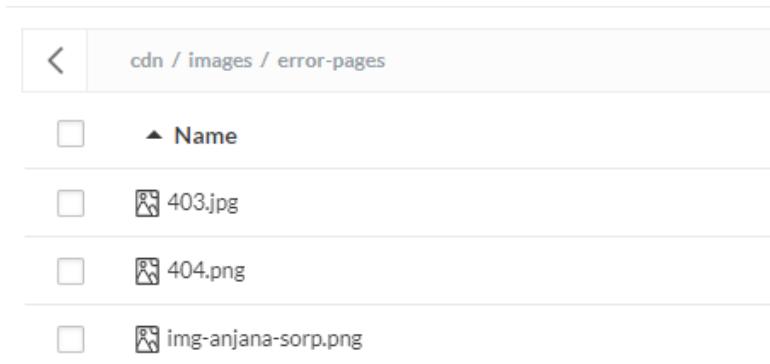
cdn / Images

- ▲ Name
- brand
- error-pages
- icons
- languages-icons
- subtype-icons

En la carpeta **brand** encontraremos imágenes relacionadas con la identidad de marca, logotipos, y la imagen que se mostrará en la cabecera de la aplicación. Ejemplo:



En la carpeta **error-pages** este directorio se encuentran las imágenes que aparecerán en caso de error en la aplicación, por ejemplo errores 403 o 404.



cdn / images / error-pages

- ▲ Name
- 403.jpg
- 404.png
- img-anjana-sorp.png

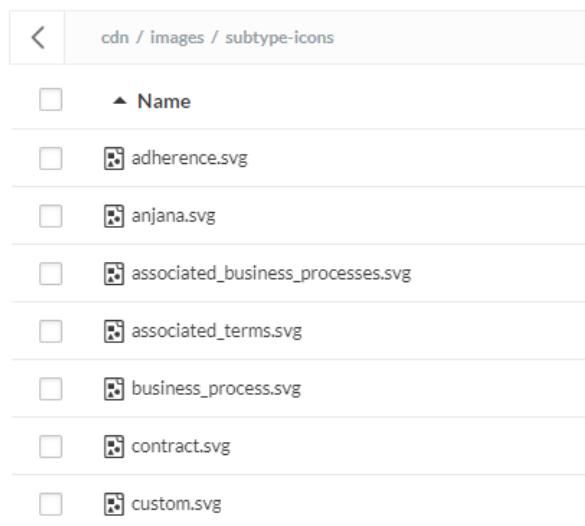
La carpeta **icons** contiene todos los iconos del core de la aplicación, los no personalizables. Ejemplo:



La carpeta **language-icons** Contiene los iconos de idioma que se estarán utilizando en los componentes de traducción. Ejemplo:



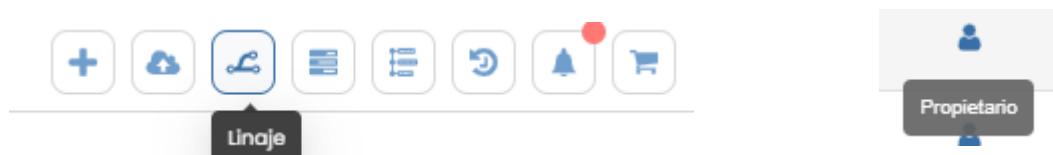
En la carpeta **subtype-icons** están todos los iconos personalizables de entidades y relaciones, en el apartado “Iconos personalizados” de este documento se detallan los requisitos que han de cumplir estos iconos y la forma de incluirlos en la aplicación, para su correcto funcionamiento.



## Iconos por defecto

Son aquellos iconos que vienen por defecto en toda la aplicación y que no pueden ser modificados por el cliente. Para mejorar la experiencia de usuario, Anjana dispone de tooltips que describen el término/acción que representa el ícono.

Algunos ejemplos:



## Iconos personalizados

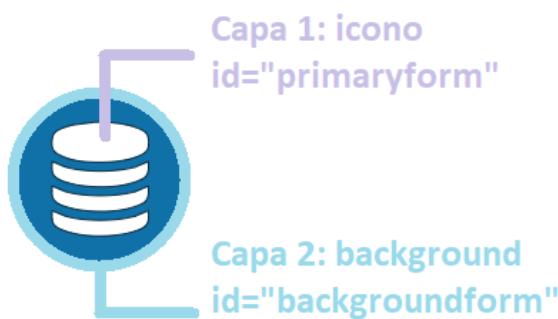
Anjana permite personalizar los iconos que se usan para los diferentes tipos de objetos en Data Catalog y en Business Glossary.



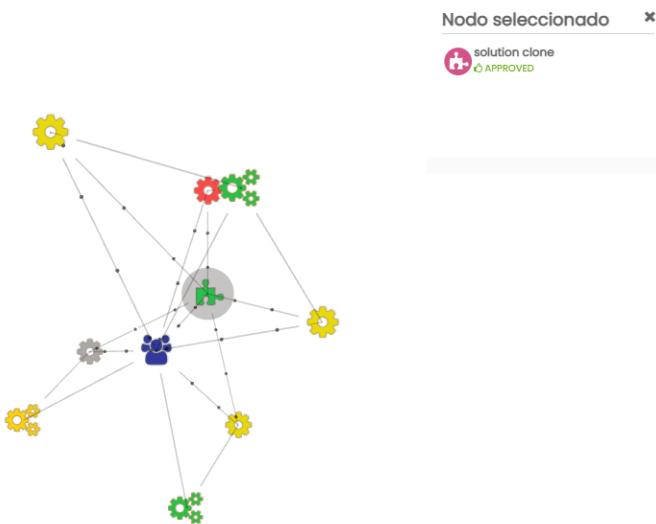
### Características que deben cumplir los iconos

Para que los iconos se vean correctamente tanto en las páginas de linaje (grafo) como en el resto de la aplicación es necesario que los iconos cumplan los siguientes requisitos:

- Formato **SVG**
- Tamaño recomendado **5KB**
- La etiqueta `svg` tiene que tener como `id='svg'`
- El archivo `svg` debe estar compuesto por 2 capas, cada una de ellas con su correspondiente `id`:
  - Icono: el ícono solo debe estar en una única capa, con `id="primaryform"`
  - Fondo: círculo de color que está bajo el ícono, con `id="backgroundform"`



En el grafo, únicamente se visualiza la forma del icono (sin el fondo), y su color depende del estado del objeto al que representa. Para ello hay que comprobar y añadir, en el caso de que no lo tuviera, que el xml de los svg tengan los identificadores esperados como se visualiza en la imagen anterior.

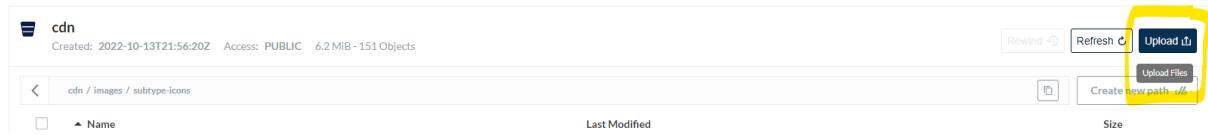


Es preferible que el ícono tenga el mayor cuerpo posible, para ser reconocible en los gráficos de líneaje. Véase de las siguientes opciones, la primera se vería mucho mejor en estas pantallas.



## Cómo subir los iconos en la aplicación

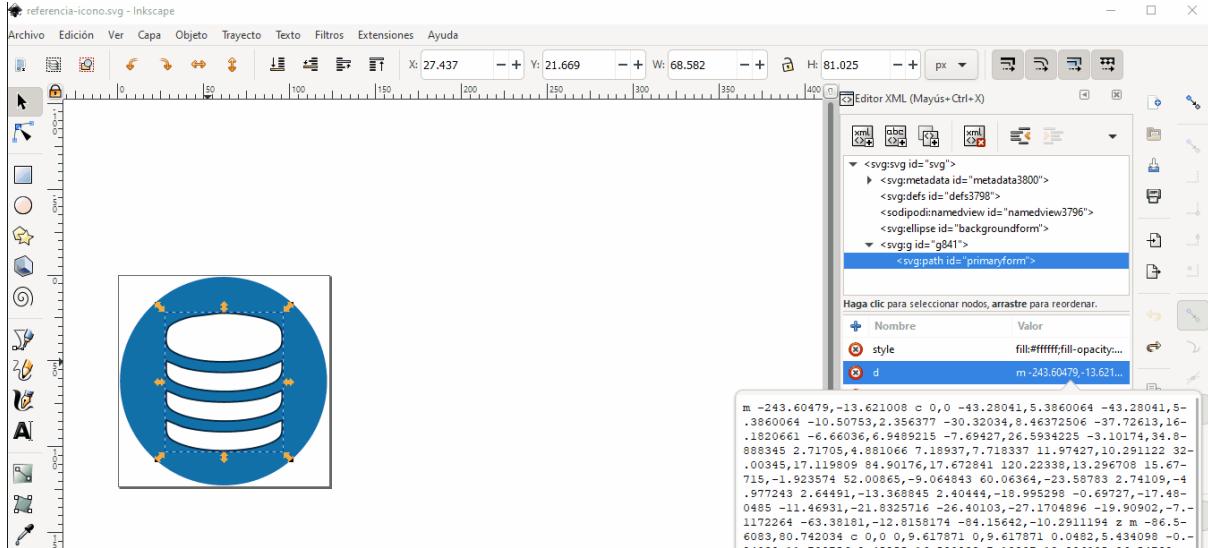
1. Entrar en la web de minio, carpeta `cdn/images/subtype-icons/`
2. Botón upload y seleccionar los iconos escogidos



## Cómo personalizar iconos de subtipo de objeto para Anjana

Para que los iconos se vean correctamente tanto en todas las vistas como en los gráficos de linaje, los svg deben configurarse como se ha explicado previamente. Para ello es necesario utilizar un editor de svg, como puede ser [InkScape](#), una herramienta de software libre que permite editar los svg con la configuración que se necesita:

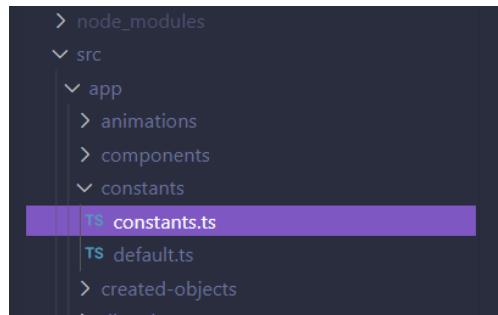
- Abrir [un archivo](#) de subtipo referencia que será modificado para obtener el icono personalizado.
- Abrir en otra pestaña de inkScape el icono que se quiere incluir.
- Clickar la capa donde se encuentra la imagen del icono. **Copiar** el código que contiene D.
- Seleccionar la capa primary form del icono-referencia, y en el apartado D, borrar el código que hubiera y **pegar** el código D que se ha copiado previamente.
- Presionamos enter y se deberá mostrar el nuevo icono.
- Redimensionar y centrar el icono. Si es necesario, se puede cambiar el color del fondo (capa con id 'background form')
- 



**Importante:** a la hora de guardar y subir el fichero al minio, es muy importante que el nombre del fichero sea igual que el subtipo de objeto en minúsculas. Por ejemplo: para el subtipo de entidad REPORT el icono debe llamarse report.svg

En la aplicación tendremos una lista con los nombres de nuestras entidades.

Esta lista se encuentra en el archivo de constantes de la aplicación en la siguiente ruta:



Para modificarla debemos sustituir los iconos de subtipo que trae Anjana por defecto, la lista está ordenada alfabéticamente con las claves de cada ícono, que utilizaremos con el atributo **svgIcon** y la clave del ícono correspondiente

```
<mat-icon svgIcon="solution" class="static-transparent icon-container-option"></mat-icon>
```

```
export const SUBTYPE_ICON_LIST = [
  'adherence',
  'associated_business_processes',
  'associated_terms',
  'business_process',
  'custom',
  'data_quality_rule',
  'dataset_field',
  'dataset',
  'default',
  'dimension_dataset',
  'dimension',
  'document',
  'dq_rule_dataset',
  'dsa',
  'instance',
  'kpi',
  'organizational_unit',
  'policy_dsa',
  'policy',
  'process',
  'related_dataset',
  'related_dq_rules',
  'related_reports',
  'relationship',
  'report_attributes',
  'report_dataset',
  'report',
  'solution',
  'term_dataset',
  'term_dc',
  'term',
  'user'
]
```

**Importante:** Si no incluimos en esta lista un ícono nuevo que incluyamos en la carpeta "subtype-icons", nos lanzará un error avisándonos de que no encuentra el ícono.

# CSS

Actualmente la aplicación está utilizando dos librerías, Material y NgZorro, pero poco a poco se irá dejando de lado Material para mantener la aplicación con NgZorro. Por el momento los estilos de cada una están separados en sus respectivas carpetas:

```
> environments
  ✓ scss
    ✓ material
      ❄ icons.scss
      ❄ index.scss
      ❄ scrollbar.scss
      ❄ styles_media.scss
      ❄ styles.scss
      ❄ typography.scss
      ❄ variables.scss
    ✓ ng-zorro
      {} index.less
      ❄ index.scss
      ❄ nz-buttons.scss
      ❄ nz-data-table.scss
      ❄ nz-form.scss
      ❄ nz-modal.scss
      ❄ nz-progress.scss
      ❄ nz-skeleton.scss
      ❄ nz-stepper.scss
      ❄ reset.scss
      ❄ root.scss
      ❄ styles.scss
```

## Archivo index.scss

Este archivo sirve para importar todos los archivos de estilos y así unificarlos.

```
1  // Mains
2  @import "reset";
3  @import "root";
4
5  // Override libs
6  @import "material/index.scss";
7  @import "ng-zorro/index.scss";
8
9  // Quill textenriched styles
10 @import "~quill/dist/quill.core.css";
11 @import "~quill/dist/quill.bubble.css";
12 @import "~quill/dist/quill.snow.css";
```

Los archivos dentro de las carpetas **roots** y **reset** cuentan con los estilos generales (colores, tipografía, medidas...), en los archivos index.scss de las carpetas material y ng-zorro están los estilos que han sido necesario sobreescribir de las librerías para que se ajustasen a la estética de Anjana y en el último bloque están unos archivos correspondientes al componente de editor de texto que no se debe editar.

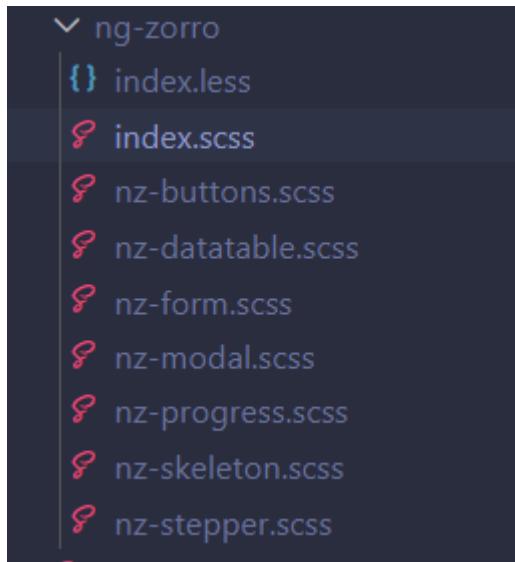
## Material

```
@import "./scrollbar.scss";
@import "./styles_media.scss";
@import "./typography.scss";
@import "./variables.scss";
@import "./icons.scss";
@import "./styles.scss";
```

```
⌄ material
  ⌄ icons.scss
  ⌄ index.scss
  ⌄ scrollbar.scss
  ⌄ styles_media.scss
  ⌄ styles.scss
  ⌄ typography.scss
  ⌄ variables.scss
```

## Ng-zorro

```
@import "./nz-buttons.scss";
@import "./nz-datatable.scss";
@import "./nz-form.scss";
@import "./nz-modal.scss";
@import "./nz-progress.scss";
@import "./nz-skeleton.scss";
@import "./nz-stepper.scss";
```



## Archivos de Material

### Archivo styles.scss

En ambas carpetas de ng-zorro y material, el archivo styles.scss contiene estilos globales para componentes comunes, que apliquen de forma general en toda la app.

### Archivo styles\_media.scss

En este archivo encontraremos media queries en caso de que sean necesarias para implementar un diseño responsive

### Archivo variables.scss

En este archivo tenemos variables globales que reutilizamos en toda la aplicación. En el caso de las variables de la carpeta de material están precedidas por el símbolo del dólar.

```
// STATE
$color-imported: #FF6E9F;
$color-draft: #E8C00D;
$color-disabled: darkgrey;
$color-expired: $color-dark;
$color-waiting: #BD9964;
$color-pending: #E9910E;
$color-rejected: #AA1617;
$color-deprecated: #FA4A4B;
$color-cancelled: #7E0607;
```

Este archivo de variables desaparecerá y será sustituido por las variables de la carpeta de ng-zorro , que están en el archivo scss/ root.scss

En este momento en el archivo de variables de material están clasificadas por colores principales, relacionados con la identidad de marca de Anjana:

```
// Anjana Color Palette

$color-primary: #36699C;
$color-secondary: #FFA500;
$color-tertiary: #4978AA;
$color-dark-blue: #172B4D;
$color-app-prime: #8E8E8E;
$color-dark-app-prime: #757575;
$color-primary_transparent: rgba(54, 105, 156, 0.6);
```

## Support colors

```
// Support colors
$color-error: #F44336;
$color-success: #5AA700;
$color-warning: orange;
$color-info: #40829D;
$green-ok: #00FB00;
```

Estos en realidad deberían estar unificados con los colores de estado, que es lo que se corrige con las nuevas custom properties

```
// STATE
$color-imported: #FF6E9F;
$color-draft: #E8C00D;
$color-disabled: darkgrey;
$color-expired: $color-dark;
$color-waiting: #BD9964;
$color-pending: #E9910E;
$color-rejected: #AA1617;
$color-deprecated: #FA4A4B;
$color-cancelled: #7E0607;
```

Colores de textos y fondos:

```
// More Colors:
$color-lighter-blue: #EAEFF4;
$color-light: #66BAFF;
$color-light-midtone: #36699C;
$color-dark-blue: #37699C;
$color-pure-white: #FFFFFF;
$color-background-light-grey: #EBEBEB;
$color-dark-midtone: #D7D7D7;
$color-light-grey: #D7D7D7;
$color-grey-text: #B3CBD4;
$color-grey-text-secondary: #C4C4C4;
$color-midtone: #8C9BA5;
$color-dark-grey: #757575;
$color-dark: #172B4D;
$color-pure-black: #000000;
$color-lighter-gray: #F1F1F1;
```

Colores para iconos:

```
// Icon colors

$color-icon-pink: #AC0754;
$color-icon-blue: #055583;
$color-icon-green: #35A065;
$color-icon-yellow: #F7DF09;
$color-icon-default: darkgrey;
$color-icon-purple: #rgb(140, 79, 168);
$color-icon-lightblue: #0DA0F5;
$color-icon-dark-grey-1: #rgb(73, 73, 85);
$color-icon-dark-grey-2: #rgb(89, 74, 95);
$color-icon-dark-grey-3: #666666;

// Icon colors by typeObject
$color-icon-relationship: #77D2D8;
$color-icon-term: #7956C3;
$color-icon-dsa: #54C991;
$color-icon-dataset: #1070A8;
$color-icon-dataset_field: #4AA8FF;
$color-icon-process: #FFB100;
$color-icon-instance: #FCC05F;
$color-icon-solution: #D35389;
$color-icon-organizational-unit: darkblue;
$color-icon-user: $color-error;
$color-icon-default: #666666;

// Historic Icon colors
$color-historic-icon-pink: #D70555;
$color-historic-icon-blue: #05A6D7;
$color-historic-icon-green: #16D49B;
$color-historic-icon-yellow: #D7AA05;
$color-historic-icon-default: darkgrey;
```

Otras variables que están repetidas o hay que refactorizar y que hasta que no tengamos completamente sustituido por ng-zorro no podemos eliminarlas:

```
//bar colors
$color-white-bar: ■rgb(245, 245, 245);
$border-bar-label: 1px solid □rgba(5, 5, 5, 0.05);
$shadow-bar-label: 1px 2px 0px □rgba(5, 5, 5, 0.05);
$color-success-bar: ■#5AA700;
$color-light-gradient-success-bar: ■#39C38D;
$color-dark-gradient-success-bar: ■#00B3B4;

$color-warning-bar: ■#FFD20D;
$color-light-gradient-warning-bar: ■#FFA700;
$color-dark-gradient-warning-bar: ■#FFA807;

//Height
$height-header: 3.1rem;

//Width
$width-alphaNavCompact: 3rem;
$width-alphaNav: 3rem;

//view/edit entity colors
$color-green-step: ■#8FCC67;
$color-yellow-step: ■#E2B437;
$color-pink-step: ■rgb(224, 42, 112);
$color-purple-step: ■rgb(127, 80, 155);

//REVIEW/nuevas variables para no perder colores que hay con hexadecimal por la app

$bg-light-gray: ■#F8F8F8;
$bg-custom-attributes: ■#EEF4FB;
$bg-delete-icon-custom-attributes: ■#929292;
$bg-hover-delete-icon: ■#FFDFDF;
$black-text: □#656565;
$subtitle-name-workflows: ■#92B7CB;
$header-border-bottom: □#36699C30;
$workflows-icon-approve-hover: □rgba(90, 167, 0, .2);
$workflows-icon-reject-hover: □rgba(244, 67, 54, .2);      You, 5 months ago • botones
$color-background-edit-form: ■#E0E0E0;
```

## Archivo typography.scss

Este archivo está vacío, se creó por si fuera necesario para darle estilos a las fuentes, pero por el momento, no se está usando.

## Archivo icons.scss

En este archivo tenemos estilos específicos de iconos

## Archivos de Ng-zorro

### Archivo nz-buttons.scss

Contiene estilos para botones

### Archivo nz-datable.scss

Estilos de tablas

### Archivo nz-form.scss

Estilos de formularios

### Archivo nz-modal.scss

Estilos para modales

### Archivo nz-progress.scss

Estilos para barras de progreso

### Archivo nz-skeleton.scss

Estilos de componentes skeleton de precarga

### Archivo nz-stepper.scss

Estilos de procesos por pasos como el de wizard de creación.

## Algunos aspectos generales

Los archivos de estilos sólo se importan la primera vez que carga la app, así optimizamos la carga.

Las variables de material se mantienen hasta que se refactorizan los componentes que dependen de ellas, si necesitamos cambiar colores, por ejemplo, el color primario, hemos de cambiarlo en el archivos de la carpeta de material “variables.scss” y en el de la carpeta de angular “root.scss” para que cambie en todas partes, ya que conviven componentes de ambas librerías.

Una vez todo esté migrado a ng-zorro, sólo habrá que sustituirlos en el el archivo “root.scss”.

Todos los componentes se pueden editar como se quieran creando custom classes dentro del propio scss y html, de ser necesario, para evitar repetir clases con la misma funcionalidad, entonces se crean dentro de la carpeta de estilos globales.