**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Sistema Web de Búsqueda de Cursos utilizando Web Scraping  
  
Guía de estilo de codificación**

**Versión <1.0>**

Integrantes:

Fajardo Astete, Manuel Alexandro 17200107

Gonzales Orosco, Jhon Anderson 17200276

Romaní Diaz, René Angel 15200224

Campos Apaza, Leyla Gabriela 17200261

**Historia de Revisión**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 13/11/2021 | 1.0 | Guía de estilo de codificación | Grupo 2 |

Contenido

[1. Sobre la Guía de estilo de codificación 4](#_Toc87919106)

[2. Angular 5](#_Toc87919107)

[2.1. Estructura de un proyecto 5](#_Toc87919108)

[2.1.1. Directorio media 5](#_Toc87919109)

[2.1.2. Módulo core 5](#_Toc87919110)

[2.1.3. Módulo data 5](#_Toc87919111)

[2.1.4. Directorio layout 7](#_Toc87919112)

[2.1.5. Directorio modules 7](#_Toc87919113)

[2.1.6. Módulo shared 7](#_Toc87919114)

[2.1.7. Directorio styles 7](#_Toc87919115)

[2.2. Usar alias para las importaciones 8](#_Toc87919116)

[3. Typescript 9](#_Toc87919117)

[3.1. Nombres 9](#_Toc87919118)

[3.2. null y undefined 9](#_Toc87919119)

[3.3. Comentarios 9](#_Toc87919120)

[3.4. Strings 9](#_Toc87919121)

[3.5. Estilos 9](#_Toc87919122)

[4. Referencias 11](#_Toc87919123)

# Sobre la Guía de estilo de codificación

Este documento se centra en las guías de estilo para el desarrollo de aplicaciones web mediante el framework Angular y Typescript.

En principio, este documento se ha escrito para describir las convenciones en la codificación del proyecto Edutec.

Este documento se basa en el documento de Nandakumar R. titulado “Angular coding standards” y el proyecto Angular Folder Structure.

# Angular

## Estructura de un proyecto

Se recomienda utilizar la siguiente estructura de directorios, la cual es flexible para proyectos grandes o pequeños.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Directorio media

Este directorio se utiliza para almacenar archivos de apoyo para la aplicación. Cosas como documentación de requisitos, esquemas de texto, etc.

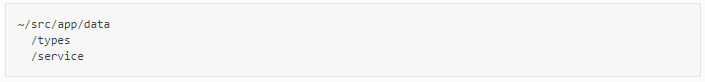
### Módulo core

Este módulo es para clases utilizadas por app.module. Los recursos que siempre se cargan, como los guards, interceptors y los servicios de nivel de aplicación, como ThemeService.

### Módulo data

Este módulo es un directorio de nivel superior y contiene los tipos (modelos/entidades) y servicios (repositorios) de los datos consumidos por la aplicación.

Se debe crear los siguientes subdirectorios:



El directorio types contiene los archivos de definición de clases para estructuras de datos. Una estructura de datos de ejemplo:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

El directorio de servicios contiene los servicios para obtener datos. Los archivos de servicio no son necesariamente una coincidencia 1: 1 con los archivos de tipos. Un ejemplo de archivo de servicio:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si su aplicación consume datos de más de una fuente, entonces el directorio de datos debe reestructurarse para contener subdirectorios para cada fuente de datos. No cree varios módulos para cada fuente de datos:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

### Directorio layout

Este directorio contiene uno o más componentes que actúan como un layout o son parte de uno, como un header, navbar, footer, etc.

### Directorio modules

El directorio modules contiene una colección de módulos que son independientes entre sí. Esto permite que Angular cargue solo el módulo que requiere para mostrar la solicitud, lo que ahorra ancho de banda y acelera toda la aplicación. Para lograr esto cada módulo debe tener su propio enrutamiento.

### Módulo shared

Este módulo contiene clases y recursos que se utilizan en más de un módulo cargado dinámicamente. Al cargar siempre con la aplicación, los componentes compartidos están listos cuando un módulo los solicita.

### Directorio styles

Este directorio se utiliza para almacenar hojas de estilo (css, scss) para la aplicación. Puede contener temas, Bootstrap, Angular Material y cualquier otro estilo.

## Usar alias para las importaciones

Poner un alias en las carpetas de nuestra aplicación y entornos nos permitirá implementar importaciones limpias que serán consistentes en toda nuestra aplicación.

Considere la situación hipotética, pero habitual. Estamos trabajando en un componente que se encuentra a tres carpetas de profundidad en una función A y queremos importar el servicio desde el módulo core que se encuentra a dos carpetas de profundidad. Esto daría lugar a que la declaración de importación se parezca a:



Y lo que es aún peor, cada vez que queramos cambiar la ubicación de cualquiera de esos dos archivos, nuestra declaración de importación se romperá. Compare eso con la siguiente importación:



# Typescript

## Nombres

* Usar PascalCase para nombrar los types.
* No usar “I” como prefijo de nombres de interfaces.
* Usar PascalCase para valores enum.
* Usar camelCase para nombrar funciones.

## null y undefined

* Evitar en lo posible el uso de null, en su lugar usar undefined.

## Comentarios

* Usar el estilo de JSDoc para comentar funciones, interfaces, enumeraciones y clases.

## Strings

* Usar comillas simples para los strings.

## Estilos

* Usar funciones de flecha sobre funciones anónimas.
* Solo usar paréntesis en los parámetros de funciones de flecha cuando sea necesario.
* Por ejemplo: (x) => x + x es incorrecto, pero lo siguiente es correcto:

Imagen que contiene Texto

Descripción generada automáticamente

* Siempre rodee los cuerpos de un bucle y condicionales con llaves. Las declaraciones en la misma línea pueden omitir llaves.
* Usar un espacio para las comas, punto y coma, paréntesis. Por ejemplo:

Texto

Descripción generada automáticamente

* Utilice una sola declaración por variable:



* Utilice 2 espacios para la identación.

# Referencias

<https://angular.io/guide/styleguide>

<https://www.perfomatix.com/wp-content/uploads/2020/04/Perfomatix_Angular-Coding-Standards.pdf>

<https://angular-folder-structure.readthedocs.io/en/latest/overview.html>