



Universidad Tecnológica Nacional

Ingeniería en Sistemas de la Información

PRÁCTICO N° 6 - Requerimientos Ágiles

Cátedra de Ingeniería de Software

Grupo N°9

Curso: 4K4

Integrantes:

- | | |
|--------------------------|---------------|
| • Acevedo Hernán | Legajo: 72380 |
| • Aresu Barella, Bruno | Legajo: 73446 |
| • Bardin, Tomás | Legajo: 72783 |
| • Boné, Nicolás | Legajo: 72624 |
| • D'uva, Micaela | Legajo: 74300 |
| • Frattin Juan | Legajo: 71939 |
| • Molina, Matias | Legajo: 75477 |
| • Pereira Duarte, Martin | Legajo: 76367 |

Docentes:

- Covaro, Laura Ines
- Robles, Joaquín Leonel
- Crespo, María Mickaela

Fecha de entrega: 27/04/2021

BUENAS PRÁCTICAS IMPLEMENTADAS EN EL PRÁCTICO N°6

Angular

Responsabilidades

Funciones pequeñas: se deben limitar a un tamaño no mayor a de 75 líneas, ya que son más fáciles de probar, promueven la reutilización, y son más fáciles de mantener.

Estructura de archivos

Utilizar el acceso directo `hero.component.ts|html|css|spec` para representar esos varios archivos. Permite más facilidad de leer las estructuras de archivos, y las hace más concisas.

Nombrado

Utilizar nombres claros, separando palabras con puntos y guiones. Debe aplicarse tanto para métodos como archivos, servicios, variables, etc.

Typescript

Sintaxis

- **Identificadores**: los nombres de los identificadores de datos deben utilizar únicamente caracteres ASCII, el guión bajo (_) y la barra invertida (\). Utilizaremos la siguiente convención para el nombrado de los identificadores:
 - **UpperCamelCase** → Emplearemos este formato para Clases, Interfaces, Tipos de datos, Decoradores y Parámetros de Tipo.
 - **lowerCamelCase** → Emplearemos este formato para variables, parámetros, funciones y propiedades.
 - **CONSTANT_CASE** → Emplearemos este formato para variables globales.
 - **#ident** → Nunca emplearemos identificadores privados
- **Estilo de Nombrado**: para el nombrado en sí de los diferentes identificadores, más allá del formato de mayúsculas y minúsculas, vamos a seguir las siguientes reglas.
 - Omitiremos palabras que sean obvias debido a la declaración del tipo de variable que sea. Por ejemplo, no llamaremos a algo declarado como String, **nombreString**.
 - Omitiremos detalles irrelevantes. Por ejemplo, si tenemos una variable auto, no la nombraríamos como `autoExtravaganteDeSuperLujo`, pues todo lo que está después de auto es irrelevante.
 - Omitiremos palabras que sean claras debido al contexto en el que se encuentren las variables. Por ejemplo, en una clase Auto, no buscamos declarar sus atributos como `nombreAuto`, simplemente nombre, porque se conoce el contexto del mismo.
 - Omitiremos palabras que se puedan aplicar a cualquier identificador, como `data`, `manager`, `engine`, `object`, etc.
- **Nombres Descriptivos**: buscamos que los nombres de los identificadores que utilizaremos sean descriptivos y claros para cualquier nuevo lector. Evitaremos

utilizar abreviaciones ambiguas o poco familiares para lectores fuera del proyecto, y no utilizaremos abreviaciones que quiten letras que se encuentren dispersas por las palabras.

HTML

Sintaxis

- Comentarios en una única línea
- Utilice etiquetas en minúscula de manera consistente.
- Asegúrese de cerrar todas las etiquetas HTML de manera apropiada.

CSS

- Comentarios en una única línea
- Utilice etiquetas en minúscula de manera consistente.
- Asegúrese de cerrar todas las etiquetas HTML de manera apropiada.

Javascript

Sintaxis

- Utilice === en lugar de ==
- No utilizar "eval"
- No dejar sentencia del tipo "debugging" tales como console.log en el código de ambiente de producción

Consistencia

- Evitaremos patrones que son conocidos por causar problemas, especialmente para usuarios nuevos al lenguaje.
- Buscaremos que el código sea mantenible a largo plazo.

Bibliografía

- <https://google.github.io/styleguide/tsguide.html>
 - <https://angular.io/guide/styleguide>
 - <https://testing.googleblog.com/2017/10/code-health-identifiernamingpostforworl.html>
 - <https://code.tutsplus.com/tutorials/30-css-best-practices-for-beginners--net-6741>
 - <https://code.tutsplus.com/tutorials/24-javascript-best-practices-for-beginners--net-539>
- 9