

DOCUMENTO DE ESTILO DE CÓDIGO



TANGO APP – ACEPTAR COTIZACIÓN

Grupo 7

Docentes:

- Ing. Judith Meles
- Ing. Cecilia Massano
- Constanza Garnero

Integrantes:

68719 - Albarracín, Gonzalo Nahuel
81907 - Amormino, Julieta
90052 - Cavalleris Malanca, Octavio Félix
83493 - Chavez Insegna Fabrizio Daniel
84282 - Martinoli Marchionni José Ignacio
88208 - Rodriguez Ambard, Frank Jose
77523 - Talavera, María Azul

ÍNDICE

Aspectos de implementación	2
Responsabilidad única	2
Use componentes comunes	2
Agrupe archivos relacionados en carpetas	2
Funciones pequeñas	2
Nombres	3
Identificación	3
No agregues lógica de filtrado y ordenación a pipes	3
Evita el estilo ng-deep	3
Evita el estilo en línea	3
Variables SCSS	4
Diseño Flex	4
Indentación y formato de archivos/métodos/fragmentos de código	4
Cambios de configuración	4
Aspectos de estilo	4
Color de la aplicación	4
Botones - Iconos	5
Selección de cotización - Estilo tarjetas	5
Mensajes	5
Notificaciones que se envían por mail	6
Máscaras	6
Formato pantalla	7

Aspectos de implementación

Para trabajar con **Angular CLI**, nos guiamos de una serie de buenas prácticas detalladas en su documentación propia que detallamos a continuación.

Responsabilidad única

- Defina una sola cosa, como un servicio o componente, por archivo.

Use componentes comunes

- Use componentes comunes como input, button, etc. siempre que sea posible en páginas o ventanas emergentes, etc. Trate de buscar componentes existentes si pueden ser útiles. Usar componentes comunes será útil si se necesita implementar algún cambio a través de la aplicación.
- Tenga cuidado al hacer cambios en estos componentes comunes. Realice cambios en estos solo si realmente son necesarios. Como cualquier cambio en estos componentes/controles impactará otras páginas/ventanas emergentes también. Si es necesario, pida al equipo de UX/PO que revise el diseño solicitado.
- Si se requiere un nuevo componente común que no existe. Planee desarrollar este nuevo componente común en lugar de desarrollarlo en la página misma. Planifique el esfuerzo adecuado para desarrollar tal componente común.

Agrupe archivos relacionados en carpetas

- Si hay múltiples módulos, servicios, etc., intente ponerlos en una carpeta adecuada.

Funciones pequeñas

- Las funciones pequeñas son más fáciles de probar, especialmente cuando hacen una sola cosa y sirven para un solo propósito.
- Las funciones pequeñas son más fáciles de leer, reutilizar y mantener.

Nombres

- Los nombres de las carpetas y archivos deben reflejar claramente su intención.
- Separe los nombres de archivos con puntos.
- Use nombres de tipos convencionales que incluyan `.service`, `.component`, `.pipe`, `.module` y `.directive`.
- Los nombres de tipos no abreviados como `.service` son descriptivos y no ambiguos. Las abreviaturas como `.srv`, `.svc` y `.serv` pueden ser confusas.
- Use camelcase en mayúsculas para los nombres de clases.

Identificación

- Nombra el archivo de forma que sepas al instante lo que contiene y representa. Evite los archivos con varios componentes, varios servicios o una mezcla de ambos. Cree carpetas con el nombre del área que representan.

No agregues lógica de filtrado y ordenación a pipes

- Evita agregar lógica de filtrado u ordenación en pipes personalizados.

Evita el estilo ng-deep

- Evita usar `ng-deep`. Aplicar la pseudo-clase `::ng-deep` a cualquier regla CSS desactiva completamente el encapsulamiento de vistas para esa regla. Cualquier estilo con `::ng-deep` aplicado se convierte en un estilo global.

Evita el estilo en línea

- Evita usar estilo en línea en controles HTML. Donde sea necesario, crear clases SCSS y aplica esas clases en la UI.

Variables SCSS

- Para valores comunes como color, fuente, etc., declara variables comunes y úsalas en toda la aplicación.

Diseño Flex

- Usa diseño flex para estructurar y alinear controles en los componentes.

Indentación y formato de archivos/métodos/fragmentos de código

- Al crear/cambiar un archivo, método o cualquier código, confirma que cada declaración esté escrita con la indentación y el formato correctos.

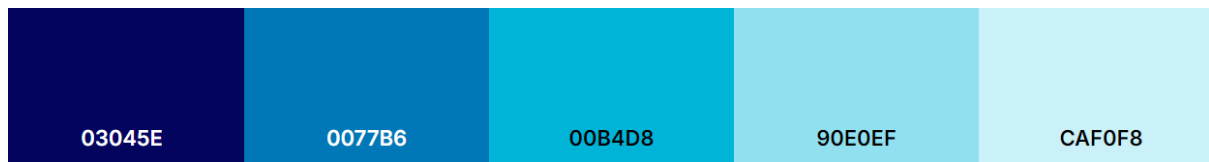
Cambios de configuración

- Usa el siguiente archivo de configuración del entorno para los cambios relacionados con la configuración en node:
`src\config\environment.config.js`

Aspectos de estilo

Color de la aplicación

Se utilizará la siguiente paleta de colores:



Mensajes de error:



Mensajes de advertencia:



Mensajes de confirmación:



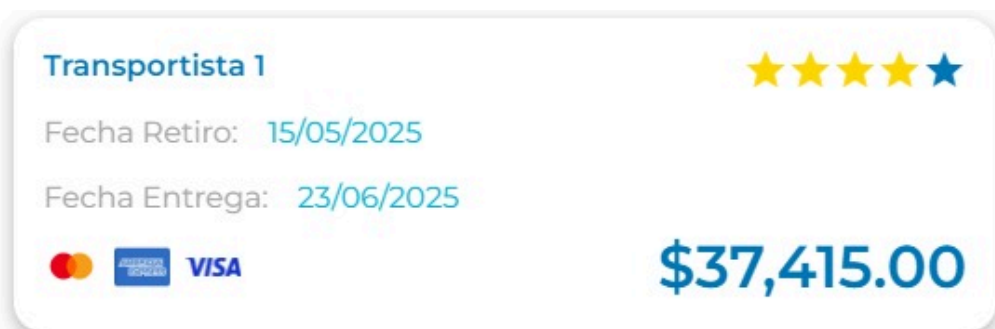
Botones - Iconos

En la medida de lo posible se usarán botones grandes en lugar de iconos para determinar las acciones a realizar.



Selección de cotización - Estilo tarjetas

El sistema muestra todas las cotizaciones en formato tarjeta con los siguientes datos:



En el caso de no estar habilitada se muestra:

Transportista 2

★★★★★

Fecha Retiro: 19/09/2024

Fecha Entrega: 01/01/2025

VISA  

\$34,590.00

Mensajes

Mensaje de confirmación:

Cotización confirmada Pedido #ORD-0226196

Mensaje de advertencia:

Tarjeta con saldo insuficiente.

Mensaje de error:

La tarjeta ingresada no es valida.

Notificaciones por email

Asunto: Cotización confirmada - Pedido #ORD-XXXXXXXXXX

Cuerpo del mail:

¡Cotización Confirmada!

Estimado/a Usuario,

¡Su cotización ha sido aceptada y el servicio de transporte ha sido confirmado!

A continuación, le detallamos la información del envío:

Transportista: <<nombreTransportista>>

Calificación: <<X estrellas>>

Fecha de Retiro: <<fechaRetiro>>

Fecha de Entrega: <<fechaEntrega>>

Forma de Pago: <<formaPago>>

Costo Total: \$costoTotal

Gracias por utilizar nuestros servicios. Si tiene alguna pregunta, no dude en contactarnos.

Este es un correo automático, por favor no responda a este mensaje.

Máscaras

Las máscaras quedan definidas de la siguiente manera:

Fecha de expiración	mm/aa
Número de tarjeta	0000 0000 0000 0000
Fechas	dd/mm/aaaa
CVV	000

dd: día - valor numérico

mm: mes- valor numérico

aa: dos últimos dígitos de año - valor numérico

aaaa: año - valor numérico

0: valores del 0 al 9

Formato pantalla

El texto de los label debe estar arriba de las cajas de texto o combos y deben estar agrupados por comportamiento similar para ayudar a la visualización del usuario, las separaciones se dan por líneas.

Por ejemplo:

Numero de documento*

123123123