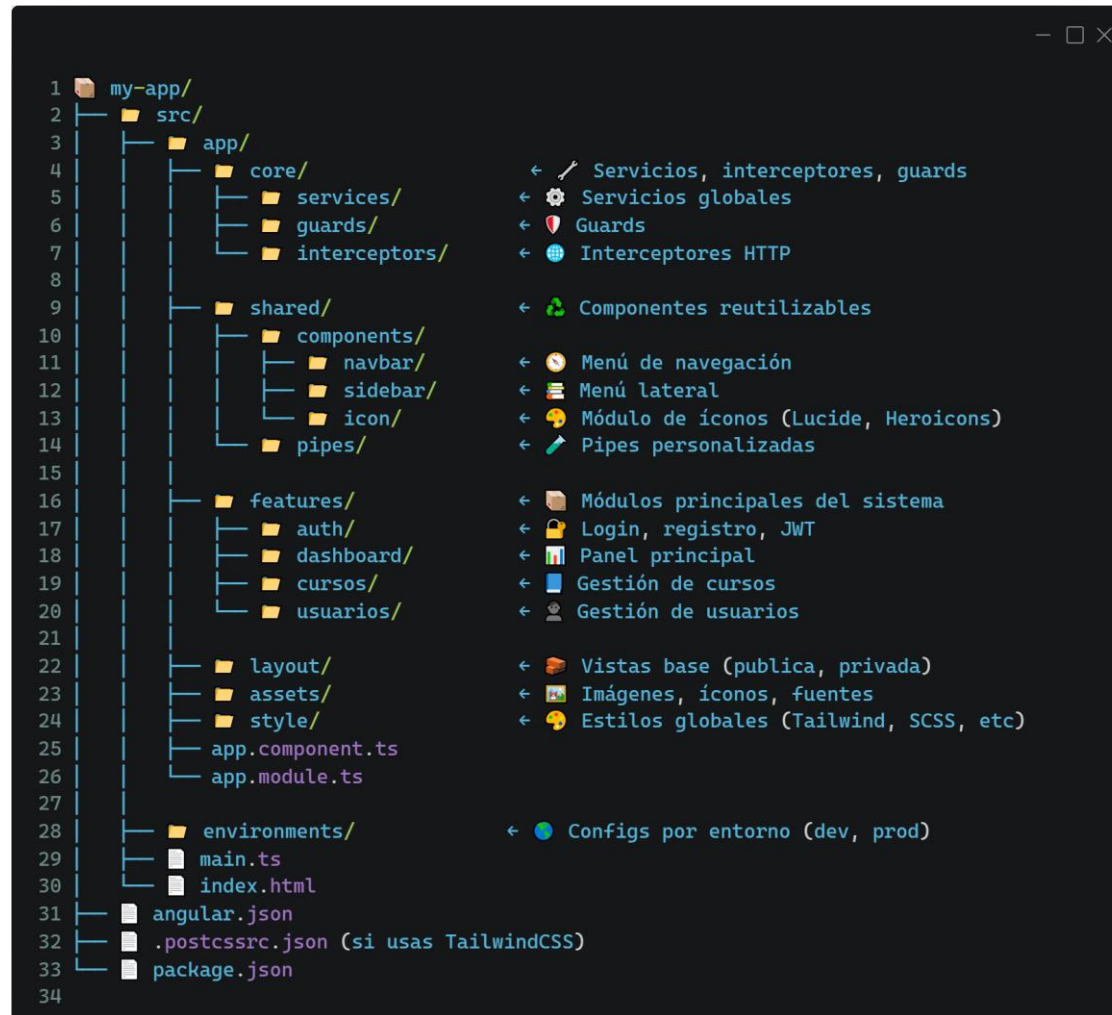


ESTRUCTURA DE UN PROYECTO EN ANGULAR



1. ESTRUCTURA DE CARPETAS:

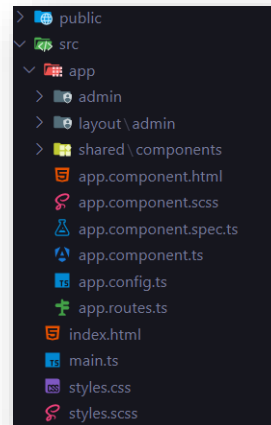
La organización de las carpetas sugiere una separación clara de responsabilidades:

ADMIN: Contiene los componentes y módulos específicos de la sección de administración. Dentro de esta carpeta, se encuentran funcionalidades como dashboard, goals, recent, saleslist, y starred. Esto indica que la aplicación permite a los usuarios gestionar información relacionada con estos aspectos.

LAYOUT/ADMIN: Parece definir la estructura visual general de la sección de administración. La carpeta layout-admin probablemente contiene el componente principal del layout, y las carpetas header-admin y sidebar sugieren la presencia de una barra de encabezado y una barra lateral para la navegación dentro de la administración.

SHARED/COMPONENTS: Contiene componentes reutilizables como chart, left-filters, plataforma, revenue, right-side, top-sales, y user-top. Esto promueve la modularidad y la reutilización de código en diferentes partes de la aplicación.

SRC/APP: Es la carpeta principal de la aplicación Angular, conteniendo los módulos, componentes, servicios y otros archivos fuente.



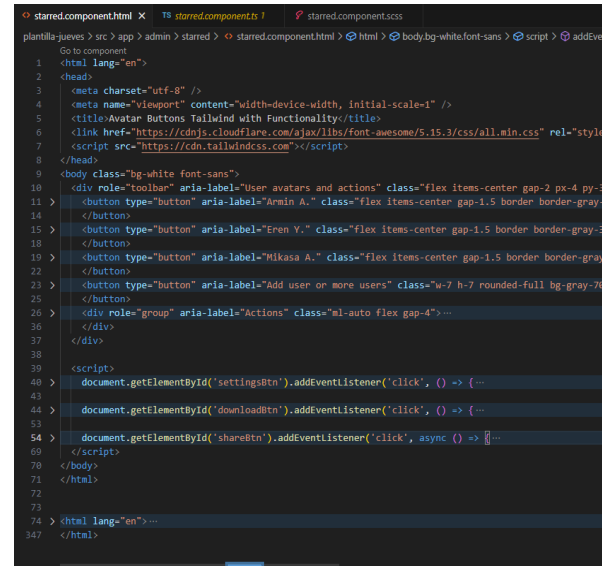
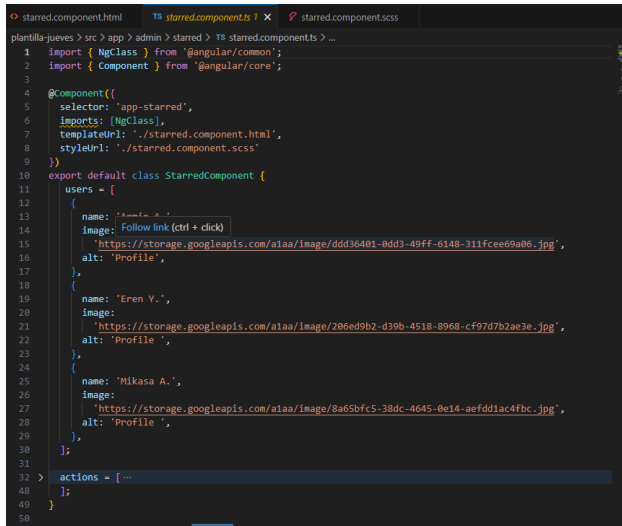
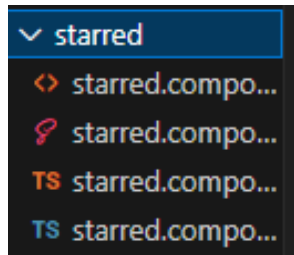
2. **ARCHIVO DE RUTAS (APP.ROUTES.TS):** Este archivo define la navegación dentro de la aplicación:

La ruta principal ("/") carga el componente LayoutAdminComponent, lo que confirma que esta es la estructura base para las rutas de administración.

- Las rutas hijas dentro de children definen las diferentes secciones del panel de administración:
- starred: Probablemente para elementos marcados como importantes o favoritos.
- recent: Para mostrar elementos o actividades recientes.
- saleslist: Para gestionar una lista de ventas.
- goals: Para el seguimiento y gestión de objetivos.
- dashboard: La página principal que probablemente muestra una visión general de la información relevante.
- La redirección de la ruta raíz ("/") a starred indica que la página de "Starred" será la página de inicio dentro de la sección de administración.

```
1 import { Routes } from '@angular/router';
2
3 export const routes: Routes = [
4   {
5     path: '',
6     loadComponent: () => import('./layout/admin/layout-admin/layout-admin.component'),
7
8     children:[
9       { path: '', redirectTo: 'starred', pathMatch: 'full' },
10      { path: 'starred', loadComponent: () => import ('./admin/starred/starred.component')},
11      { path: 'recent', loadComponent: () => import ('./admin/recent/recent.component')},
12      { path: 'saleslist', loadComponent: () => import ('./admin/saleslist/saleslist.component')},
13      { path: 'goals', loadComponent: () => import ('./admin/goals/goals.component')},
14      { path: 'dashboard', loadComponent: () => import ('./admin/dashboard/dashboard.component')}
15    ]
16  },
17 ];
18
19 ];
20
21
```

Se utiliza la carga lazy loading (loadComponent) para cargar los componentes de cada ruta solo cuando se navega a ellos, lo que puede mejorar el rendimiento inicial de la aplicación.



"Starred" es una carpeta en tu proyecto de Angular que contiene los archivos fuente (HTML, CSS, TypeScript) del componente llamado starred. Al hacer clic, **ves esos archivos** para poder editarlos. Sirve para **organizar el código** de ese componente específico.

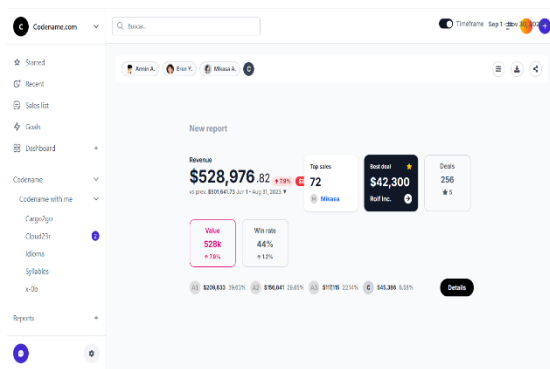
StarredComponent. Importa NgModule y Component de Angular. El decorador @Component configura el selector (app-starred), la plantilla HTML (. /starred.component.html) y los estilos SCSS (. /starred.component.scss). Define una propiedad users que es un array de objetos, cada uno con name e image (incluyendo link y alt). También declara una propiedad actions que parece ser un array vacío.

"Starred" con una barra de herramientas que contiene botones para diferentes usuarios (Armin A., Eren Y., Mikasa A.) y un botón para agregar más usuarios. Utiliza clases de Tailwind CSS para el diseño y la apariencia (flexbox, márgenes, padding, bordes, texto). También incluye un div para "Actions" y una sección de <script> con JavaScript que agrega listeners de eventos de clic a elementos con IDs "settingBtn", "downloadBtn" y "shareBtn".

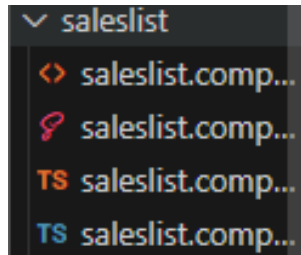
INTEGRANTES:

TENORIO TAMBRA ANDY

JESÚS VILCATOMA TUMBALOBOS



"Starred", este panel se muestra con información resumida. Presenta un "New report" con la "Revenue" total (\$528,976.82) y la cantidad de "Top sales" (72). También destaca el "Best deal" (\$42,300) y el número total de "Deals" (258). Adicionalmente, muestra métricas como "Value" (526k) y "Win rate" (64%), junto con un desglose de ingresos por diferentes fuentes y un botón de "Details".



"saleslist" en un proyecto de desarrollo. Se identifican posiblemente un archivo de componente HTML (saleslist.component.html), un archivo de estilos (quizás SCSS o CSS, saleslist.component.scss o similar), un archivo de lógica del componente en TypeScript (saleslist.component.ts), y un archivo de pruebas unitarias también en TypeScript (saleslist.component.spec.ts). Esta estructura es común en frameworks como Angular para organizar la presentación, el estilo, la lógica y las pruebas de un componente de interfaz de usuario.

```
saleslist.component.html • TS saleslist.component.ts • saleslist.component.scss •
plantilla-jueves > src > app > admin > saleslist > saleslist.component.html > html > head > meta
1 <html lang="en">
2 <head>
3   <meta charset="utf-8" />
4   <meta content="width=device-width, initial-scale=1" name="viewport" />
5   <title>Sales Dashboard</title>
6   <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet">
7   <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" rel="stylesheet">
8   <script src="https://cdn.tailwindcss.com"></script>
9
10 </head>
11 <body class="bg-white text-gray-700">
12   <div aria-label="Sales dashboard" role="main" class="max-w-xl mx-auto my-5 px-4 pb-5">
13     <section aria-label="Sales data for Armin A." class="flex items-center gap-3 mb-4 rounded-3xl bg-white">
27     </section>
28
29     <section aria-label="Sales data for Mikasa A." class="flex items-center gap-3 mb-4 rounded-3xl bg-white">
46     </section>
47
48     <section aria-label="Expanded sales details for Mikasa A." id="mikasa-expanded-content" class="mt-4">
142     </section>
143
144     <section aria-label="Sales data for Eren Y." class="flex items-center gap-3 rounded-3xl bg-white">
158     </section>
159   </div>
160 </body>
161 </html>
```



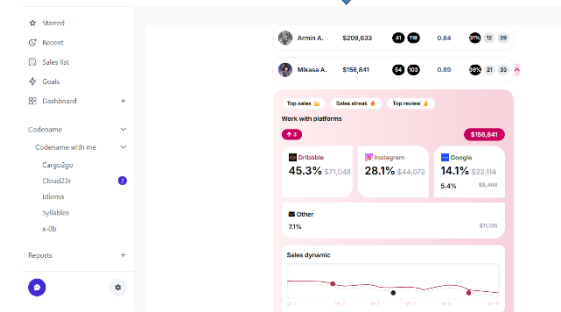
Estructura de la sección "Saleslist" de un dashboard. Define metadatos en el <head>, incluyendo título, enlaces a fuentes (Google Fonts, Font Awesome) y la biblioteca de estilos Tailwind CSS. El <body> contiene varias secciones (<section>) que muestran datos de ventas para diferentes usuarios (Armin A., Mikasa A., Eren Y.). Utiliza clases de Tailwind CSS para el diseño (fondo, texto, márgenes, padding, flexbox, bordes redondeados). Una sección parece estar etiquetada para mostrar detalles expandidos de las ventas de Mikasa A.

```
saleslist.component.html • TS saleslist.component.ts X saleslist.component.scss •
plantilla-jueves > src > app > admin > saleslist > TS saleslist.component.ts > ...
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-saleslist',
5   imports: [],
6   templateUrl: './saleslist.component.html',
7   styleUrls: ['./saleslist.component.scss']
8 })
9 export default class SaleslistComponent {
10
11 }
12
```



SaleslistComponent. Utiliza el decorador @Component para configurar su selector CSS (app-saleslist), la ruta a su plantilla HTML (./saleslist.component.html) y la ruta a su archivo de estilos SCSS (./saleslist.component.scss). La sección de imports está vacía, lo que indica que este componente no importa directamente otros módulos. Finalmente, exporta la clase SaleslistComponent para que pueda ser utilizada en otras partes de la aplicación Angular.

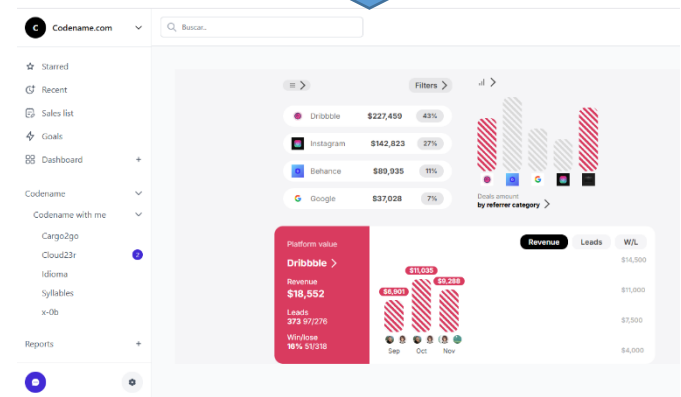
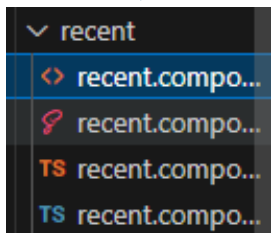
Al hacer clic en "Sales list" en el menú, este panel se muestra. Presenta información de ventas para diferentes usuarios (Armin A., Mikasa A.). También resume el trabajo con varias plataformas (Dribbble, Instagram, Google, Other) mostrando porcentajes y cantidades. Además, incluye una sección de "Sales dynamic" con un gráfico de tendencias de ventas a lo largo del tiempo. En resumen, ofrece una vista consolidada de datos de ventas y rendimiento por usuario y plataforma.

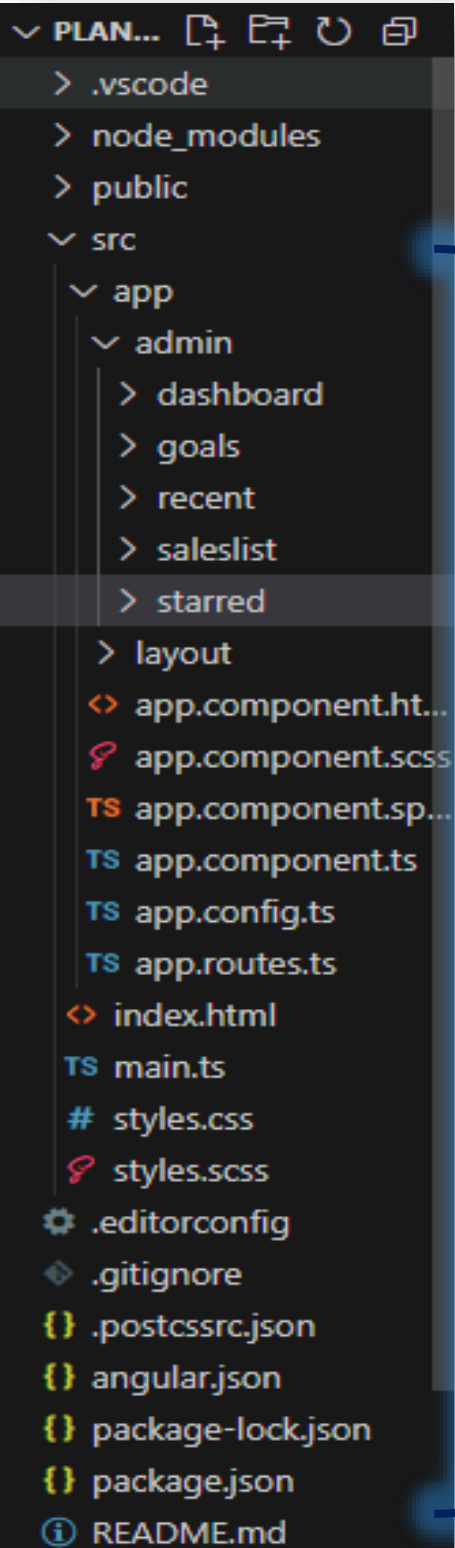


TS
(recent.component.ts):
Funcionalidad: Define la lógica y el comportamiento del componente. Qué hace: Contiene el código (en TypeScript) que maneja los datos, las interacciones del usuario y la funcionalidad del componente.

componente de Angular llamado `RecentComponent`. Utiliza el decorador `@Component` para configurar su selector (`app-recent`), la ruta a su plantilla HTML (`./recent.component.html`) y la ruta a su archivo de estilos SCSS (`./recent.component.scss`). La propiedad `imports` está vacía, indicando que no importa otros módulos directamente en este componente. Finalmente, exporta la clase `RecentComponent` para que pueda ser utilizada en otras partes de la aplicación Angular.

Este panel que ves probablemente corresponde a la sección "Recent" de la aplicación Codename. Al hacer clic en "Recent" en el menú de la izquierda, se despliega esta vista que muestra un resumen de la actividad o datos recientes relacionados con diferentes plataformas (Dribbble, Instagram, Behance, Google). Incluye métricas como ingresos, porcentajes y posiblemente tendencias visualizadas en gráficos de barras.





Dentro de src app/

Contiene la lógica y vistas principales de la aplicación Angular.

admin/: Submódulo de administración, parece estar dividido en componentes como:

dashboard/goals/recent/saleslist/starred/

Esto sugiere un diseño modular, donde cada carpeta contiene probablemente un componente Angular separado (HTML, SCSS, TypeScript).

layout/: Probablemente contiene componentes compartidos del diseño, como cabeceras, barras laterales, etc.

Archivos en app/:app.component.html: Vista principal del componente raíz.

app.component.scss: Estilos específicos del componente raíz.

app.component.ts: Lógica del componente raíz (AppComponent).

app.component.spec.ts: Pruebas unitarias del componente raíz.

app.config.ts: Configuraciones generales para la aplicación.

app.routes.ts: Definición de rutas (navegación entre componentes).

Archivos del proyecto Angular (fuera de app/)

index.html: HTML base de la aplicación Angular (punto de entrada).

main.ts: Punto de arranque de la aplicación Angular, donde se llama a bootstrapModule.

styles.scss y styles.css: Estilos globales para toda la aplicación.

Archivos de configuración

.editorconfig: Convenciones de codificación (espacios, saltos de línea, etc.).

.gitignore: Archivos o carpetas que Git debe ignorar.

angular.json: Configuración del proyecto Angular (build, assets, estilos, etc.).

package.json: Lista de dependencias y scripts del proyecto.