

INTEGRANTES:TENORIO TAMBRA ANDY

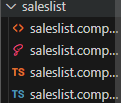
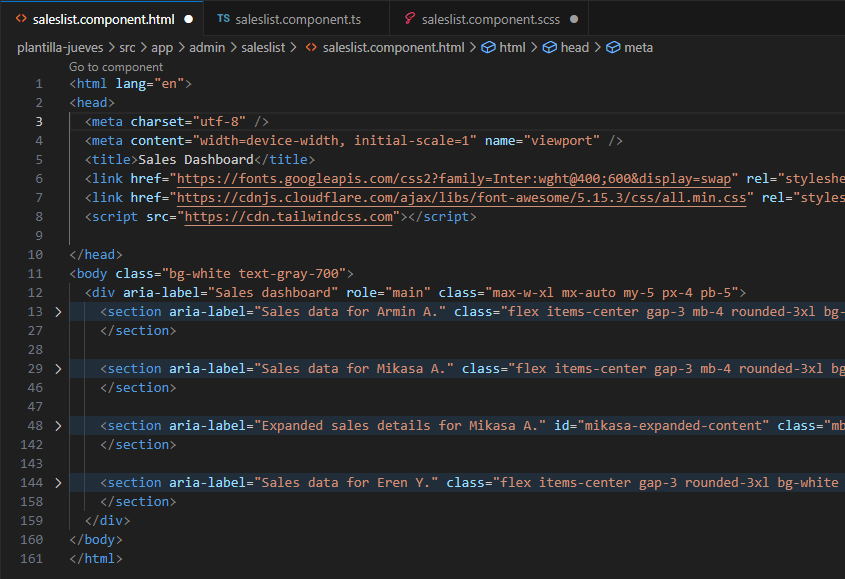
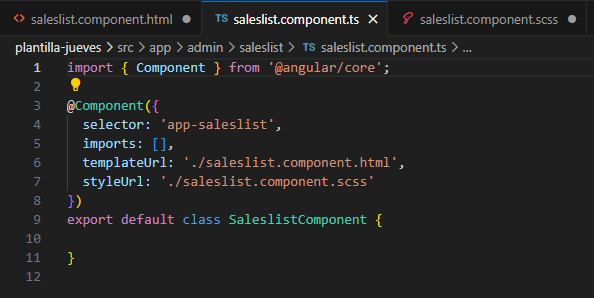
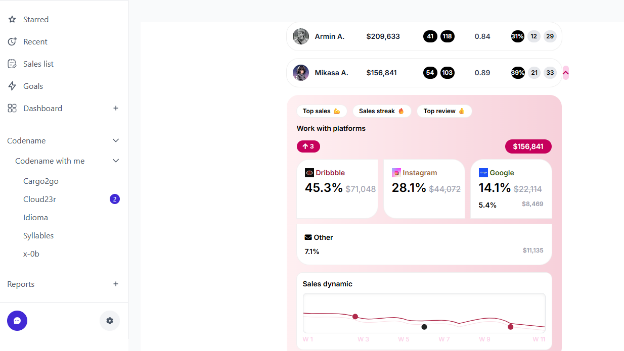
JESUS VILCATOMA TUMBALOBOS

"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".

"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.

"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".

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.

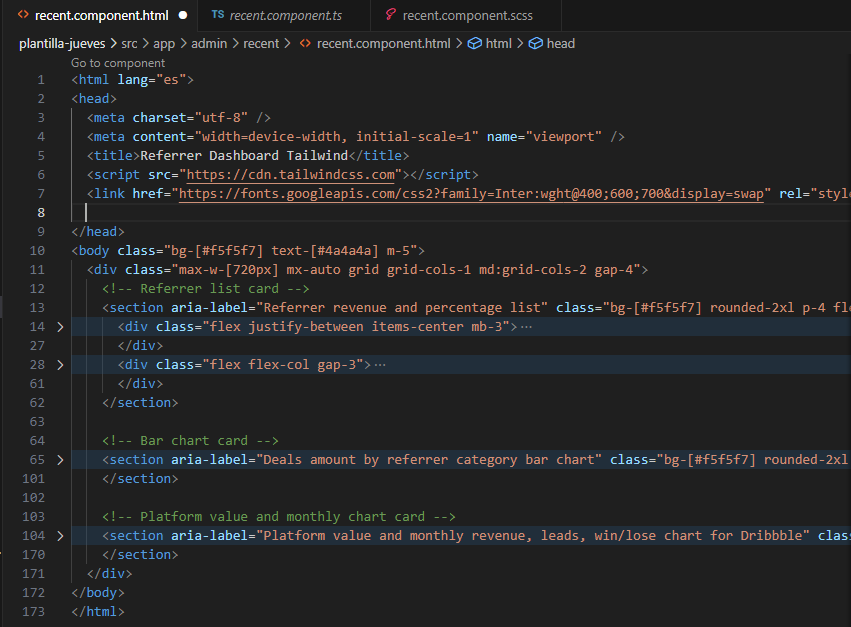
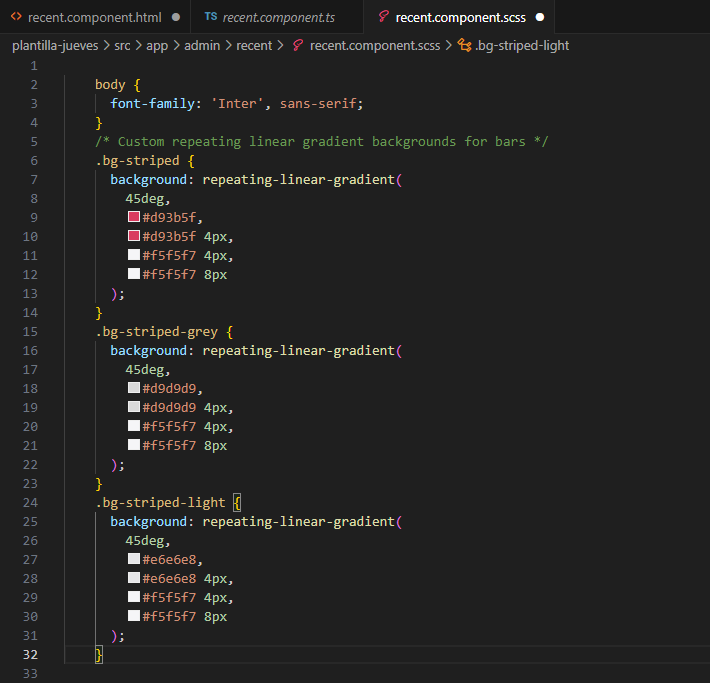


SaleslistComponent. Utiliza el decorador @Component para configurar su selector CSS (app-saleslist), la ruta a su plantilla HTML (./saleslist.component.html) y la ruta a su archivo de estilos SCSS (./saleslist.component.scss). La sección de imports está vacía, lo que indica que este componente no importa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

"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.

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.

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.

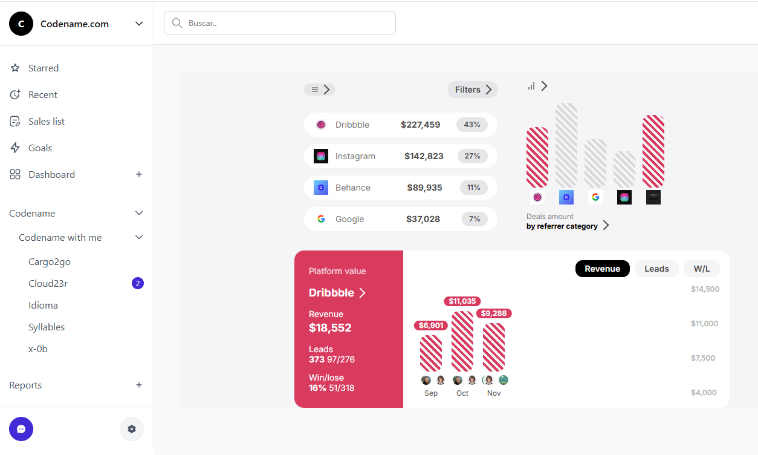
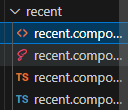


define estilos para el cuerpo (body) y tres clases personalizadas de fondo rayado: .bg-striped, .bg-striped-grey y .bg-striped-light. El estilo del body establece la fuente "Inter". Las clases de fondo utilizan la función repeating-linear-gradient para crear patrones de rayas diagonales con diferentes combinaciones de colores y tamaños de píxel. Cada clase define un conjunto específico de colores para las rayas.

recent.component.html: Funcionalidad: Define la estructura visual del componente. Qué hace: Dice qué elementos HTML se muestran y cómo están organizados.

El icono rosado (probablemente recent.component.css o similar): Funcionalidad: Define la apariencia del componente. Qué hace: Controla los estilos (colores, fuentes, espaciado, etc.) de los elementos HTML.

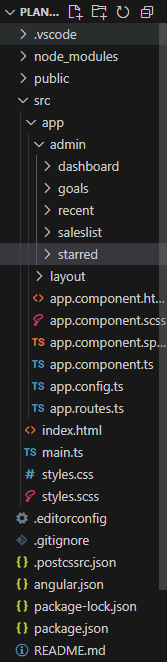
TS (recent.component.ts): Funcionalidad: Define la lógica y el comportamiento del componente. Qué hace: Contiene el código (en TypeScript) que maneja los datos, las interacciones del usuario y la funcionalidad del componente.



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.

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.

El código muestra la estructura HTML de una página web dentro de Visual Studio Code. Define el encabezado con metadatos, estilos CSS (Tailwind y Google Fonts) y un script. El cuerpo contiene varias secciones principales: una lista de referidos, una tarjeta con un gráfico de barras y otra con información de la plataforma y gráficos mensuales. Se utilizan clases CSS para el diseño y la apariencia de estos elementos. Hay comentarios HTML que explican la función de cada sección.



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.

package-lock.json: Archivo generado automáticamente que bloquea versiones exactas de dependencias.

README.md: Documento de introducción o instrucciones del proyecto.