PRUEBA N°1

Nombre: miguel Salinas

Desarrollo, pruebas y consideraciones Generales

1.- Se crea el Proyecto con el comando:

ionic start PRUEBA1 blank --type=angular

Aunque también se puede crear desde el modulo de Ionic dándole a new proyect.

1.1.- Se Crea en primera instancia la pagina de Circulo y Triangulo con los siguientes comandos en el terminal “Foto de ejemplo de como se creo”

“ionic generate component circulo –standalone”

“ionic generate component triangulo –standalone”

Texto

Descripción generada automáticamente

Foto de la estructura final del proyecto:

Texto, Aplicación

Descripción generada automáticamente

2.- Programación de la Jerarquía de Clases en TypeScript.

En el caso del circulo se realiza utilizando la fórmula matemática estándar:

Perímetro=2 × π × radio

Esto se implementó en el componente **circulo.page.ts**. Cuando el usuario introduce el radio y presiona el botón, el cálculo se ejecuta y se muestra el resultado

Texto

Descripción generada automáticamente

En el caso del triagulo se calcula sumando los 3 lados, sin embargo antes de realizar el calculo se valida que se cumplan ciertas condiciones que se piden el el requerimiento N°6 que dice que debe considerarse que es un triagulo “escaleno”

**Cumpla la desigualdad triangular**:

* La suma de dos lados es siempre mayor que el tercer lado.
* **Sea escaleno**, es decir, los tres lados deben ser diferentes.

Si ambas condiciones se cumplen, se calcula el perímetro:

* Perimetro=ladoA + ladoB+ ladoC

Esta parte del código es la valida.

Texto

Descripción generada automáticamente

3.- Formularios

Cada figura tiene su propio formulario para solicitar los valores necesarios:

* **Círculo:** Solo solicita el radio.
* **Triángulo Escaleno:** Solicita los tres lados del triángulo.

Ambos formularios están en sus respectivos archivos **.page.html**. Por ejemplo:

* **Ubicación desde donde empieza código para el triángulo “triangulo.page.html”:**

Un conjunto de letras blancas en un fondo negro

Descripción generada automáticamente con confianza media

En el caso del circulo” circulo.page.html” el botón para desencadenar el cálculo: El botón utiliza el evento (click) para ejecutar el cálculo que cumple con el requisito de la prueba:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

4.- Navegacion de entrada

La página principal (home.page.ts) actúa como un punto de entrada.

Aquí se puede seleccionar la figura geométrica y la aplicación lo dirige al formulario correspondiente.

Texto

Descripción generada automáticamente

Pantalla de celular con imagen de la pantalla de un celular

Descripción generada automáticamente con confianza media

Un ejemplo del formulario de triangulo:

Pantalla de un celular con letras

Descripción generada automáticamente

5. Validación y Resultados

Los resultados solo se muestran si el usuario presiona el botón correspondiente, y se validan las condiciones de cada figura antes de realizar el calculo por ejemplo en el caso del triangulo: se valida que sea escaleno en caso que no sea escaleno lanzara un mensaje “Los lados ingresados no forman un triángulo válido.”

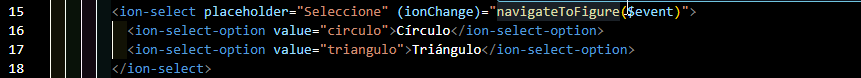
Texto

Descripción generada automáticamente

Página Principal y Resolución de Problemas

Logica de la Pagina Principal ([home.page.ts](http://home.page.ts))

La página principal actúa como un menú de navegación asi el usuario puede elegir la figura geométrica que desea trabajar. Utilizamos un elemento **ion-select** para mostrar las opciones de "Círculo" y "Triángulo", y manejamos el evento **ionChange** para redirigir al componente que se elija.



**Explicación:**

* El evento (ionChange) se activa cuando el usuario selecciona una opción.
* La función **navigateToFigure()** se encarga de redirigir al usuario a la página correspondiente utilizando el servicio **Router**.

Errores

Algunos de los problemas resueltos fue un error de servicio.

Inicialmente, el servicio **Router** no funcionaba porque olvide importar su definición desde **@angular/router**. Esto generaba errores como:

No suitable injection token for parameter 'router' of class 'HomePage'

La solución fue agregar el import correcto.

import { Router } from '@angular/router';

Problema de rutas no encontradas:

Cuando intentaba navegar a una ruta inexistente, la aplicación mostraba un error. Para resolver esto, configure una ruta comodín (path: '\*\*') en app.routes.ts que redirige al usuario a la página principal:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente