



TALLER 1

Evaluación Unidad N°1

UNIDAD DE APRENDIZAJE I
PROGRAMACIÓN HÍBRIDA

JONATHAN SOLANO F.

Introducción

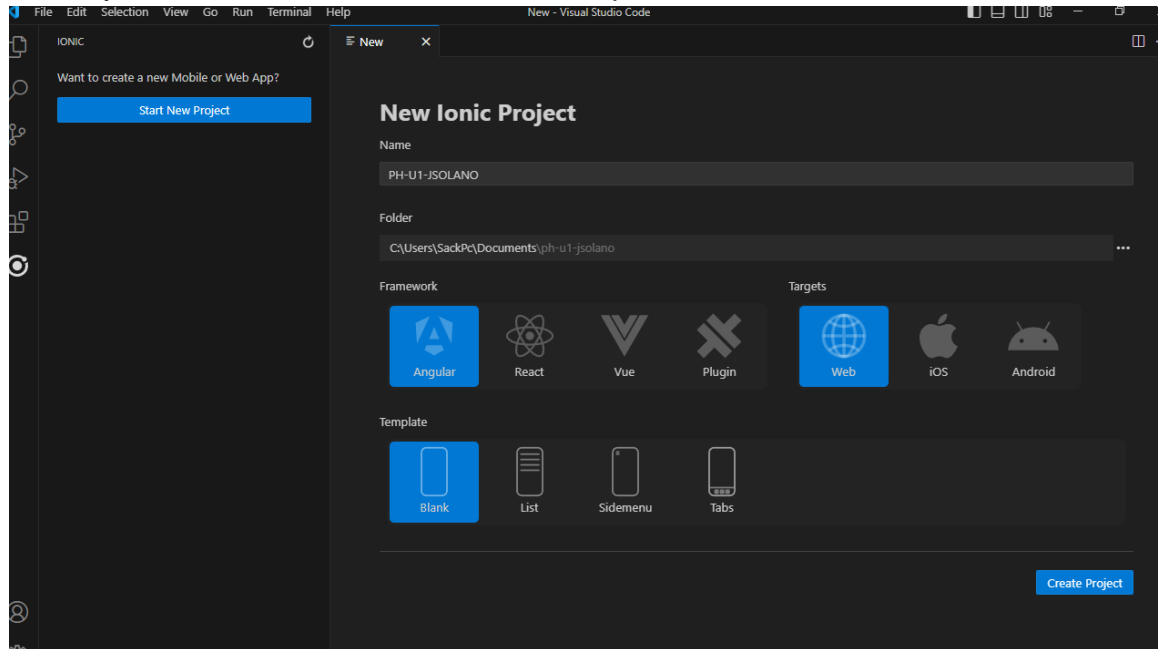
En este informe el cual consta de imágenes y una breve explicación, vamos a examinar el proyecto desarrollado para determinar los perímetros de figuras geométricas como círculos y triángulos. El proyecto se creó utilizando una plantilla en blanco, en el programa Visual Studio Code en conjunto de Angular e Ionic, el propósito es de ofrecer una manera sencilla e intuitiva de calcular los perímetros de estas figuras, también se destaca que al momento de cambiar la vista de la app en el navegador esta se ajusta de forma automática.

Dentro del proyecto se incluyen dos componentes principales: "Círculo" y "Triángulo". Estos elementos fueron diseñados para ser independientes y reutilizables, lo que permite su implementación en distintas partes de la aplicación sin afectar su funcionamiento. Cada componente cuenta con su propio formulario de entrada y lógica de cálculo, facilitando así su uso y mantenimiento.

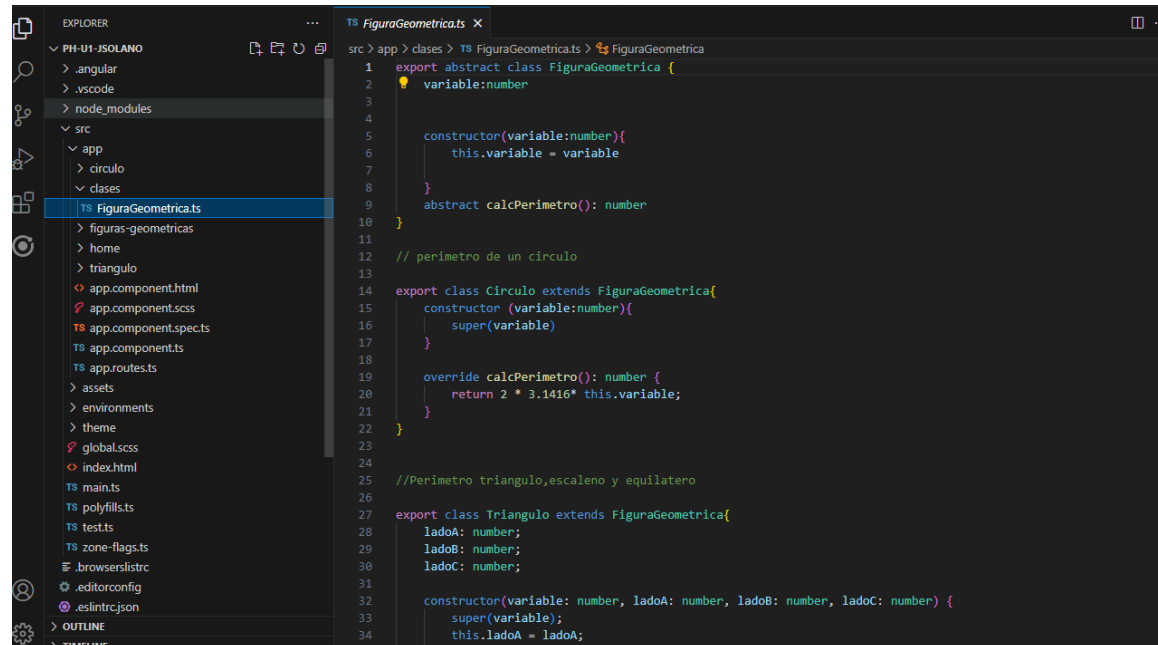
En cada componente se han integrado formularios para que los usuarios puedan introducir los valores necesarios para calcular el perímetro de la figura geométrica correspondiente. Por ejemplo, la sección "Círculo" se emplea la fórmula para calcular el perímetro de un círculo siendo ($2 * \pi * \text{radio}$) y se solicita al usuario ingresar el radio del círculo, mientras que en la sección "Triángulo" se utiliza la fórmula correspondiente al perímetro de un triángulo, siendo la suma de sus lados ($a+b+c$), los cuales deben ser ingresados. Estos formularios están diseñados para ser simples y accesibles, con elementos intuitivos. Una vez que el usuario introduce los valores en los formularios y presiona el botón "Calcular Perímetro", se activa una función encargada de efectuar el cálculo respectivo. La interfaz de usuario muestra el resultado del cálculo de forma clara.

Desarrollo del proyecto

1. Proyecto de Ionic Framework usando la plantilla en blanco



2. Las jerarquía de clases en Typescript según diagrama UML



3. Los 2 componentes Standalone de Angular dentro del proyecto: círculo y triángulo

The image displays two screenshots of a code editor, likely Visual Studio Code, showing the implementation of two standalone Angular components within a project structure.

Top Screenshot: CirculoComponent

The Explorer panel on the left shows the project structure with the file `circulo.component.ts` selected. The main editor displays the code for `CirculoComponent`:

```
src > app > circulo > TS circulo.component.ts > CirculoComponent
1 import { Component, OnInit } from '@angular/core';
2 import { Circulo } from '../clases/FiguraGeometrica';
3 import { IonImg, IonItem, IonButton, IonList, IonInput, IonSelect, IonText, IonContent, IonLabel, IonCard,
4       IonCardContent, IonCardHeader, IonCardTitle } from '@ionic/angular/standalone';
5 import { FormsModule } from '@angular/forms';
6
7 @Component({
8   selector: 'app-circulo',
9   templateUrl: './circulo.component.html',
10  styleUrls: ['./circulo.component.scss'],
11  standalone: true,
12  imports: [IonImg, IonCard, IonItem, IonList, IonCardTitle, IonCardContent, IonCardHeader, IonButton,
13           IonInput, IonSelect, FormsModule, IonText, IonLabel, IonContent]
14 })
15 export class CirculoComponent implements OnInit {
16
17   variableStr: string = "";
18   resultado: string = "";
19
20   CalcularPerimetro() {
21     const variable = parseInt(this.variableStr)
22     const circulo = new Circulo(variable)
23     const perimetro = circulo.calcPerimetro()
24     this.resultado = `El perimetro del circulo es ${perimetro.toFixed(1)} cm`
25   }
26   ngOnInit() {}
27 }
```

Bottom Screenshot: TrianguloComponent

The Explorer panel on the left shows the project structure with the file `triangulo.component.ts` selected. The main editor displays the code for `TrianguloComponent`:

```
src > app > triangulo > TS triangulo.component.ts > TrianguloComponent > CalculaPerimetro > ladoB
1
2 import { Component, OnInit } from '@angular/core';
3 import { IonImg, IonTitle, IonToolbar, IonHeader, IonCardTitle, IonCardHeader, IonItem, IonButton, IonList, IonInput, IonSelect,
4       FormsModule } from '@ionic/angular/standalone';
5 import { Triangulo } from '../clases/FiguraGeometrica';
6
7 @Component({
8   selector: 'app-triangulo',
9   templateUrl: './triangulo.component.html',
10  styleUrls: ['./triangulo.component.scss'],
11  standalone: true,
12  imports: [IonImg, IonTitle, IonToolbar, IonCardTitle, IonHeader, IonCardHeader, IonItem, IonList, IonButton, IonInput, IonSelect,
13           FormsModule]
14 })
15 export class TrianguloComponent implements OnInit {
16   ladoAstr: string = "";
17   ladoBstr: string = "";
18   ladoCstr: string = "";
19   resultado: string = "";
20
21   CalcularPerimetro() {
22     const ladoA = parseInt(this.ladoAstr);
23     const ladoB = parseInt(this.ladoBstr);
24     const ladoC = parseInt(this.ladoCstr);
25     const triangulo = new Triangulo(0, ladoA, ladoB, ladoC);
26     const perimetro = triangulo.calcPerimetro();
27     this.resultado = `El perimetro del triángulo es ${perimetro.toFixed(1)} cm`;
28   }
29
30   constructor() {}
31
32   ngOnInit() {}
33 }
34
35
```

4. Elemento ion-select en el componente "home", el que sirve para elegir la figura geométrica con la que se desea trabajar.
5. Usando la directiva *ngIf muestra el componente que corresponda según se seleccione en ion-select.

```

1  <ion-content [fullscreen]="true">
2
3  <div id="container">
4    <ion-select
5      (ionChange)="handlechange($event)"
6      label = "Figuras Geometricas"
7      placeholder="Escoje una Figura ">
8      <ion-select-option value="circulo">Circulo</ion-select-option>
9      <ion-select-option value="triangulo">Triangulo</ion-select-option>
10     </ion-select>
11
12     <app-figura geometrica />
13     <app-circulo *ngIf="infCirculo()" />
14     <app-triangulo *ngIf="infTriangulo()" />
15
16   </div>
17
18 </ion-content>
19
20
  
```

- Los formularios dentro de cada componente (círculo y triángulo) solicitan al usuario los valores necesarios para calcular el perímetro de la figura.
- Evento "click" sobre un elemento ion-button para desencadenar el cálculo del perímetro. El mensaje del resultado solo se muestra una vez que se presiona el botón.
- Dentro de cada componente se utiliza ion-card y ion-image para colocar más información de la figura geométrica.

```

src > app > círculo > círculo.component.html > ion-content > ion-card > ion-card-content#infoCirculo > ion-list > ion-item
Go to component
1
2 <ion-content [fullscreen]="true">
3   <ion-card>
4     <ion-card-header>
5       <ion-card-title>
6         Información del Círculo
7       </ion-card-title>
8     <p>El perímetro de un círculo es la longitud de la línea que forma su borde y
9       se calcula utilizando la fórmula:  $\text{perímetro} = 2 * \pi * \text{radio}$ .</p>
10
11   </ion-card-header>
12   <ion-card-content id="infoCirculo">
13     <div class="image-container">
14       <ion-img src="../../assets/imagenes/circulo.JPG" alt="Círculo"></ion-img>
15     </div>
16     <ion-list lines="none">
17       <ion-item>
18         <ion-label position="floating">Radio (cm)</ion-label>
19         <ion-input [(ngModel)]="variableStr" type="number" placeholder="ingresa un N°"></ion-input>
20       </ion-item>
21     </ion-list>
22     <ion-item>
23       <!-- color="success" para cambiar el color del botón -->
24       <ion-button color="success" (click)="CalcularPerimetro()">Calcular Perímetro</ion-button>
25     </ion-item>
26     <ion-item>
27       {{ resultado }}
28     </ion-item>
29   </ion-card-content>
30 </ion-card>
31 </ion-content>

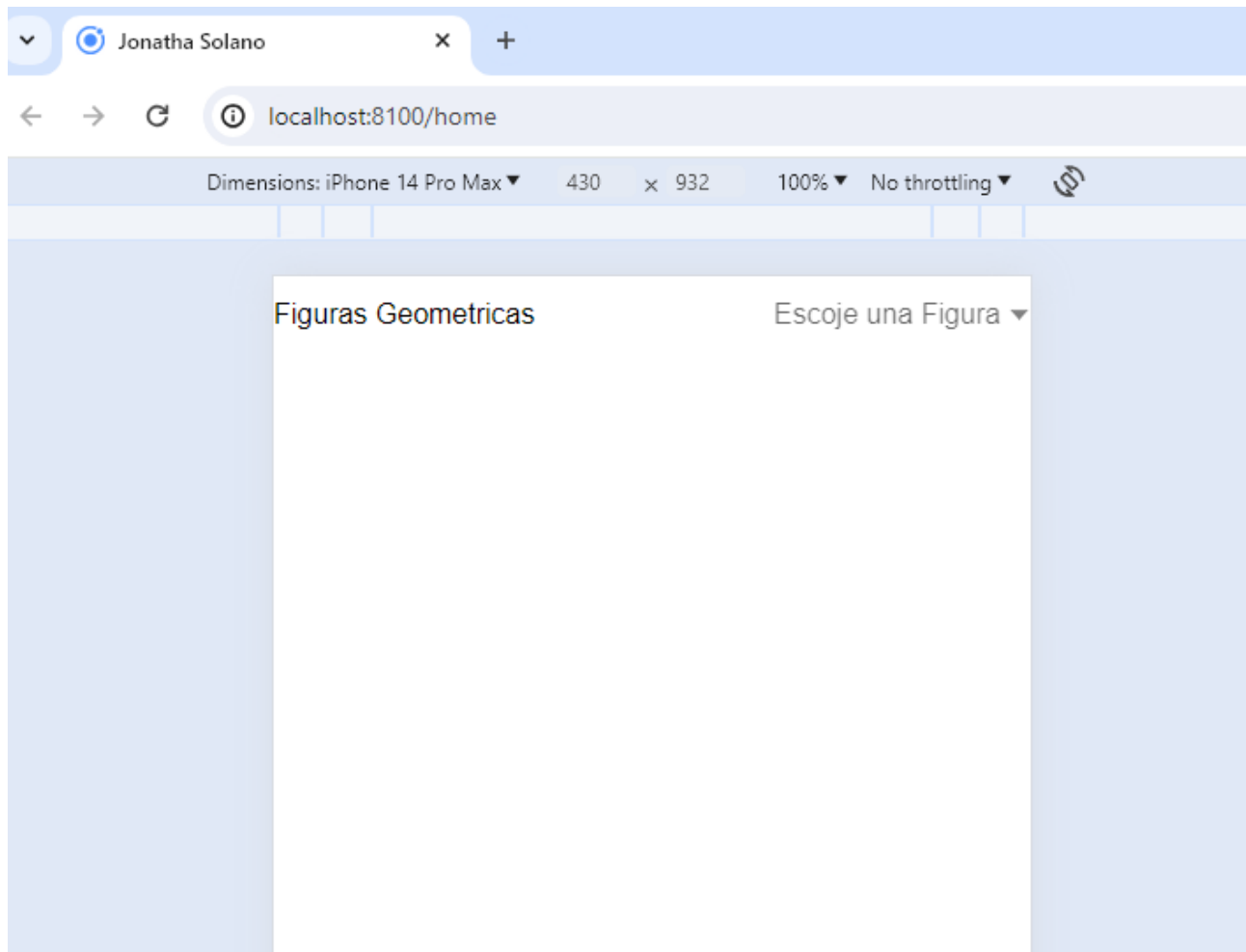
```

```

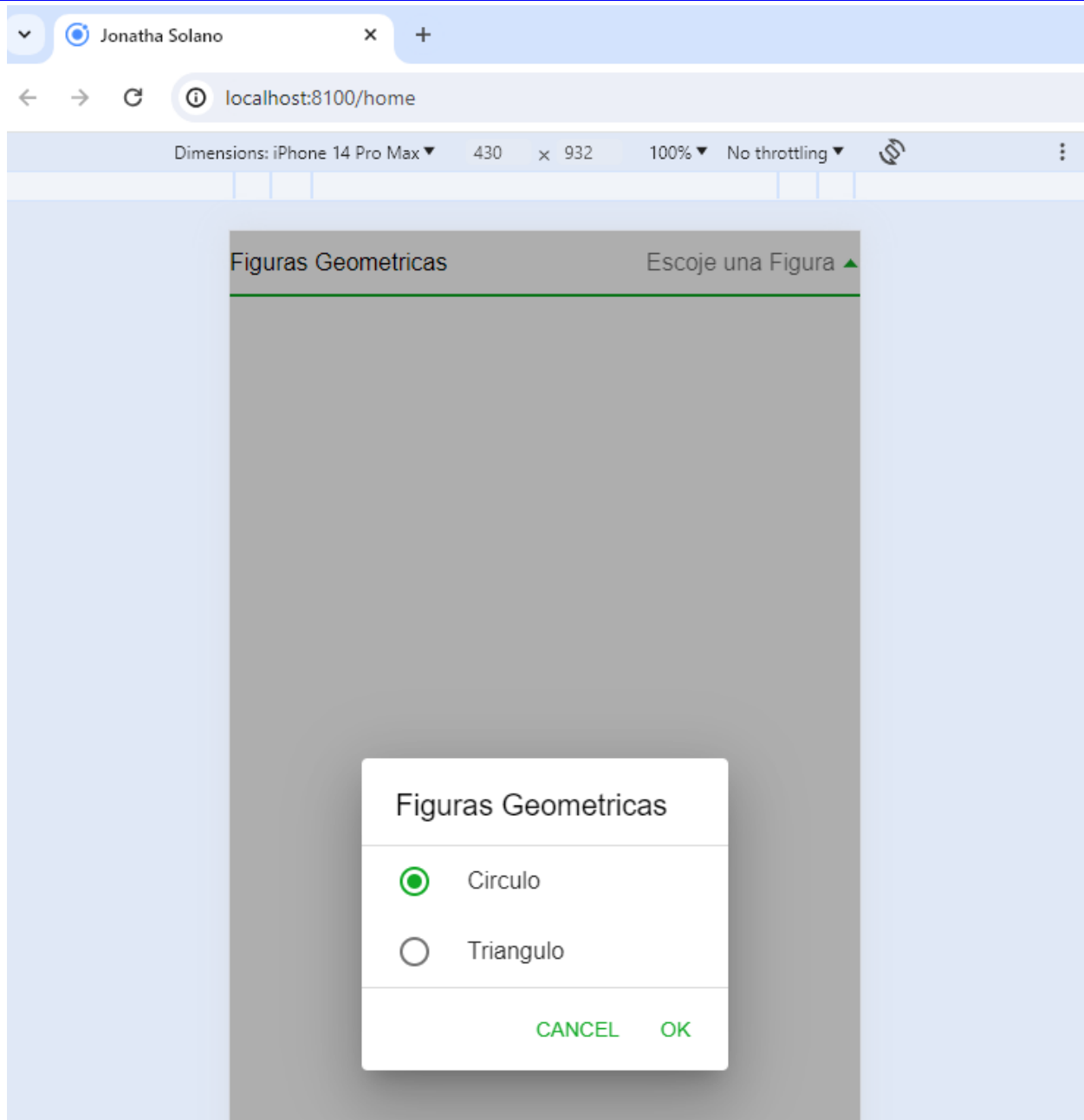
src > app > triángulo > triángulo.component.html > ion-content > ion-card > ion-card-header
Go to component
1
2 <ion-content [fullscreen]="true">
3   <ion-card>
4     <ion-card-header>
5       <ion-card-title>
6         Información del Triángulo
7       </ion-card-title>
8     <p>Para calcular el perímetro de un triángulo, necesitamos conocer la longitud de cada uno de sus lados. Un triángulo
9       comúnmente llamados a, b, y c. El perímetro de un triángulo es la suma de las longitudes de sus tres lados.</p>
10   </ion-card-header>
11   <div id="infoLados">
12     <ion-img src="../../assets/imagenes/triangulo.JPG" alt="Triángulo"></ion-img>
13     <ion-list lines="none">
14       <ion-item>
15         <ion-label position="floating">Lado A (cm)</ion-label>
16         <ion-input [(ngModel)]="ladoAstr" type="number" placeholder="ingresa un N°"></ion-input>
17       </ion-item>
18       <ion-item>
19         <ion-label position="floating">Lado B (cm)</ion-label>
20         <ion-input [(ngModel)]="ladoBstr" type="number" placeholder="ingresa un N°"></ion-input>
21       </ion-item>
22       <ion-item>
23         <ion-label position="floating">Lado C (cm)</ion-label>
24         <ion-input [(ngModel)]="ladoCstr" type="number" placeholder="ingresa un N°"></ion-input>
25       </ion-item>
26     </ion-list>
27     <ion-item>
28       <ion-button color="success" (click)="CalcularPerimetro()">Calcular Perímetro</ion-button>
29     </ion-item>
30     <ion-item>
31       {{ resultado }}
32     </ion-item>
33   </div>
34 </ion-card>

```

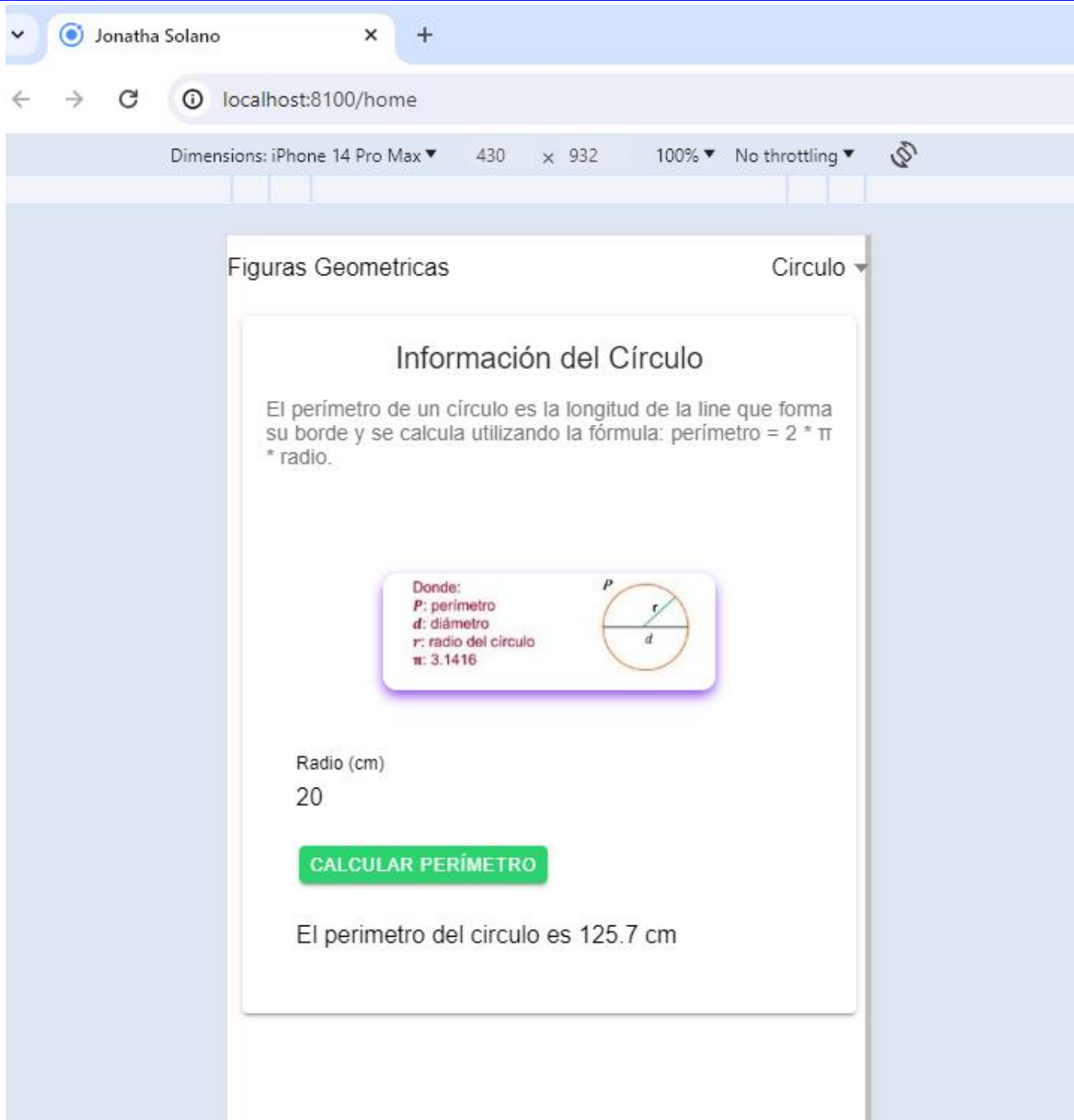
Se adjuntan imágenes del funcionamiento de la app en el navegador Chrome.



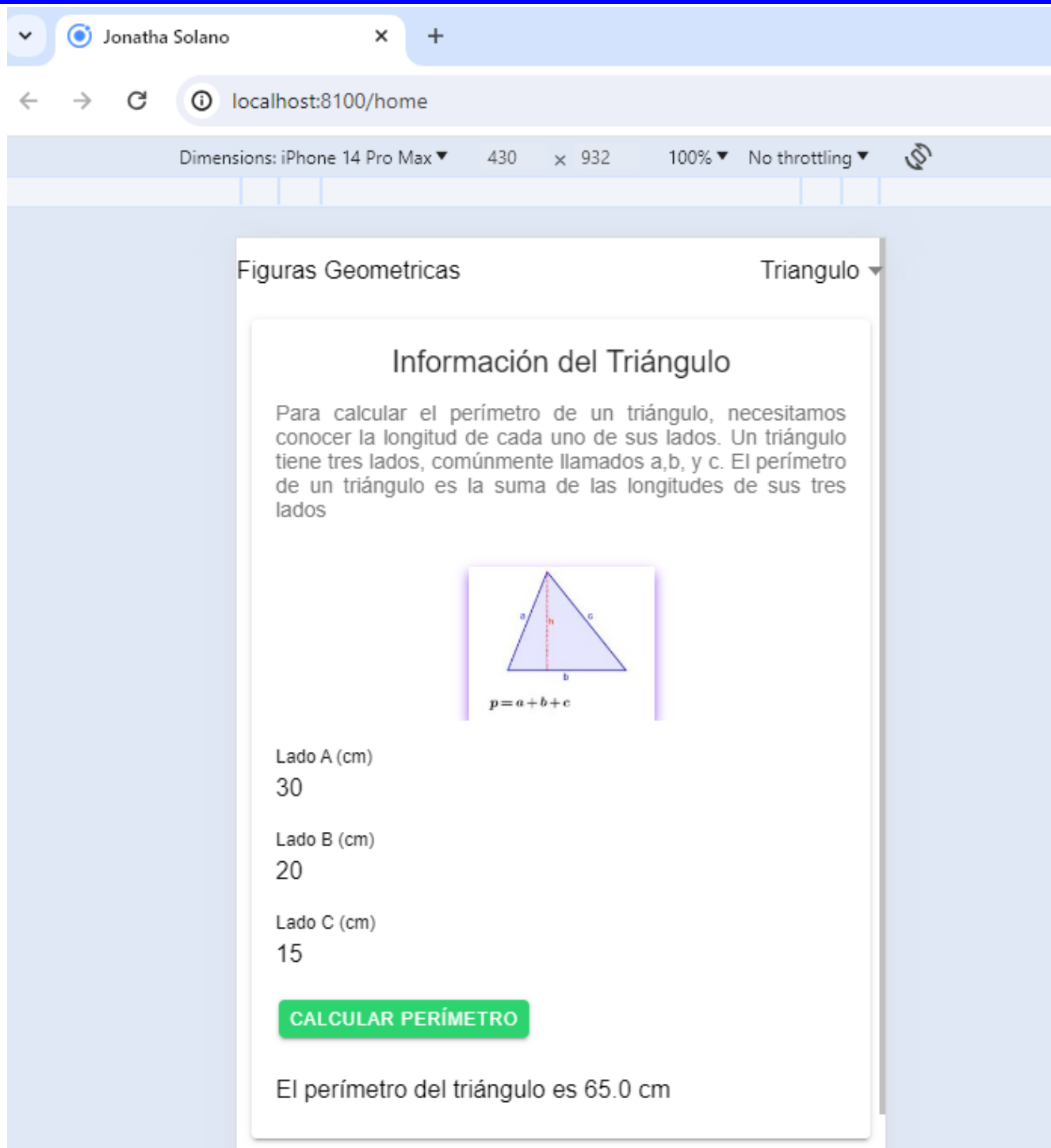
Pantalla principal



Selección de figura geométrica a trabajar usando ion-select



Ejemplo de componente Circulo en funcionamiento



Ejemplo de componente Triángulo en funcionamiento



4 INSTITUCION
ACREDITADA
NIVEL AVANZADO
AÑOS Hasta octubre 2025



GESTIÓN INSTITUCIONAL Y DOCENTE DE PREGRADO