

TALLER 1

Evaluación Unidad N°I

**UNIDAD DE APRENDIZAJE I**

**PROGRAMACIÓN HÍBRIDA**

## Instrucciones generales

Una vez realizada la lectura comprensiva de la unidad I del material de estudio, es fundamental la realización de este taller, el que tiene por objetivo medir la correcta internalización y aplicación de los conceptos abordados en las mencionadas unidades.

El desarrollo de este trabajo es individual, no se permitirán entregas de talleres en duplas o en grupos.

**Contenidos asociados al taller:**

1. Preparación del entorno para trabajar con Ionic Framework
2. Creación de proyecto de Ionic Framework
3. Codificación de componentes de Angular dentro de Ionic Framework

## Instrucciones específicas

El taller propuesto requiere articular contenidos abordados en la Unidad I , a través de las siguientes acciones:



**Al final de este documento podrás encontrar la pauta de evaluación y los indicadores de logro asociados a cada actividad.**

# Actividad 1

### Instrucciones

*Desarrolle una aplicación híbrida usando Ionic Framework que permita calcular el perímetro para distintas figuras geométricas. Se pide lo siguiente:*

1. *Crear proyecto de Ionic Framework usando la plantilla en blanco*
2. *Crear jerarquía de clases en Typescript según diagrama UML*
3. *Crear 2 componentes Standalone de Angular dentro del proyecto: círculo y triángulo*
4. *Incorpore un elemento ion-select en el componente "home", que sirva para elegir la figura geométrica con la que se desea trabajar. Este componente está previamente configurado utilizando la plantilla en blanco de Ionic para representar la página inicial.*
5. *Usando la directiva \*ngIf muestre el componente que corresponda según se seleccione en ion-select*
6. *Cree los formularios dentro de cada componente (círculo y triángulo) para solicitar al usuario los valores necesarios para calcular el perímetro de la figura. En el caso del triángulo, tenga en cuenta que se trata de un triángulo escaleno.*
7. *Utilice el evento “click” sobre un elemento ion-button para desencadenar el cálculo del perímetro. El mensaje con el resultado solo debe mostrarse una vez se presione el botón.*
8. *Dentro de cada componente utilice ion-card y ion-image para colocar más información de la figura geométrica, algo similar a como se muestran en las figuras a continuación.*

### Se pide:

1. *Elabore un informe donde explique, con sus propias palabras, los fragmentos más significativos del código, de acuerdo con los criterios de evaluación establecidos. A continuación, se presenta una estructura sugerida para la redacción del informe.*
2. *Adjunte el proyecto o incluya un enlace al mismo dentro del informe, preferiblemente en plataformas como Github, Google Drive, OneDrive, u otras similares.* ***Es esencial eliminar la carpeta node\_modules antes de adjuntar el trabajo, ya que su inclusión haría que el tamaño del proyecto sea considerablemente grande.***

### Diagrama de Clases

*A diagram of a number

Description automatically generated*

Figura 1 Diagrama de clases en UML

*Para aquellos que no estén familiarizados con UML, el diagrama presenta cuatro clases: FiguraGeometrica, Circulo, TrianguloEscaleno y TrianguloEquilatero. La primera clase, FiguraGeometrica, es abstracta, lo que significa que no se puede instanciar. Además, contiene un método llamado calcularPerimetro, que también se declara como abstracto (indicado en cursiva). Cuando un método es abstracto, implica que no es necesario proporcionar una implementación en la clase abstracta; sin embargo, las clases derivadas deben implementar este método.*

*Las otras tres clases son concretas, es decir, tienen implementación completa y son instanciables. Cada una de estas clases debe conocer la manera específica de calcular su perímetro. Es crucial respetar los tipos de datos y las firmas para cada miembro de la clase, ya sean atributos, constructores o métodos.*

### Estructura sugerida para el informe:

1. *Creación del proyecto de Ionic + Angular para plataforma Web*
2. *Programación de jerarquía de clases en Typescript (TS)*
3. *Creación de Componentes Círculo y Triángulo*
   1. *Desarrollo de la plantilla*
   2. *Desarrollo de la clase en TS del componente*
4. *Integración de los componentes en el “Home”*
5. *Dificultades encontradas*
6. *Reflexión final (cómo puedo mejorar, en qué necesito ayuda)*

A screenshot of a computer

Description automatically generated

Figura 2 Pantalla principal sugerida

A screenshot of a computer

Description automatically generated

Figura 3 Selección de figura geométrica a trabajar usando ion-select

A screenshot of a computer

Description automatically generated

Figura 4 Ejemplo de componente Circulo en funcionamiento

A screenshot of a cell phone

Description automatically generated

Figura 5 Ejemplo de componente Triángulo en funcionamiento

**Información Importante**

Esta actividad será evaluada por el profesor de la asignatura, por lo que se sugiere, para su realización, considerar los Criterios de Evaluación detallados al final de este documento. El puntaje total de este taller es de 100 puntos, y el puntaje mínimo de aprobación es de 60 puntos.

**DESARROLLO**

1. *Creación del proyecto de Ionic + Angular para plataforma Web*

Para la creación del proyecto abro Visual Studio Code. Me dirijo al logo de Ionic (descargado anteriormente) clic en él nos dará la opción START NEW PROJECT. Se abrirá una nueva ventana donde puedo elegir nombre del proyecto, el target (web) y el framework (angular) + la plantilla que en este caso comenzará en blanco (template).

Adjunto fotografías de referencias.

*Captura de pantalla de un celular

Descripción generada automáticamente*

*Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente*

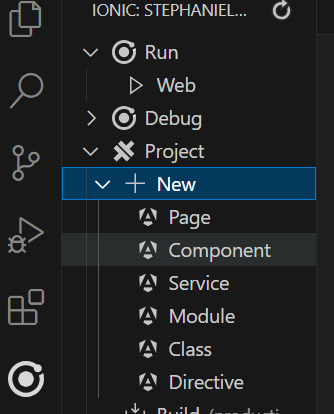
1. *Programación de jerarquía de clases en Typescript (TS)*

Para programar la jerarquía de clases debo entender que clase será la clase base. En este caso es figura-geometrica (donde se definen propiedades y métodos en comunes) luego las otras clases heredan de esta, como las clases de circulo, triangulo-escaleno y triangulo-equilatero.

1. *Creación de Componentes Círculo y Triángulo*
   1. *Desarrollo de la plantilla*
   2. *Desarrollo de la clase en TS del componente*

Tenemos 2 formas de realizar la creación de componentes, desde la terminal y desde la GUI.

Desde la interfaz se ve de la siguiente manera:



Para desarrollar la clase en TS del componente iremos a component.html y component.ts en los componentes triangulo y circulo. En estos escribiremos la lógica y lo que queremos que se desarrolle y lo que se mostrara en pantalla.

1. *Integración de los componentes en el “Home”*

La integración al Home la realicé utilizando una propiedad llamada selectedFigura, que almacena cuál de los componentes (círculo o triángulo) debe mostrar según la selección del usuario.

También utilice \*ngIf como condición para mostrar los componentes de forma dinámica.

Por ejemplo, el usuario da clic en Circulo:

* selectedFigura es 'circulo', se muestra el contenido del componente Círculo.

Además, implemente una lógica mediante una ventana emergente (AlertController) que permite al usuario elegir la figura geométrica que desea trabajar, actualizando la propiedad selectedFigura como resultado.

A mi parecer, esta es la parte más importante del código, la lógica de integración y visualización dinámica de los componentes en el Home, ya que permite que la interfaz sea interactiva y cambie según la selección del usuario.

**Propiedad selectedFigura**

selectedFigura: string = '';

**Lógica del AlertController (openFiguraOptions)**

async openFiguraOptions() {

const alert = await this.alertController.create({

header: 'Figura Geométrica',

inputs: [

{

name: 'circulo',

type: 'radio',

label: 'Círculo',

value: 'circulo',

checked: this.selectedFigura === 'circulo',

},

{

name: 'triangulo',

type: 'radio',

label: 'Triángulo',

value: 'triangulo',

checked: this.selectedFigura === 'triangulo',

},

],

buttons: [

{ text: 'CANCEL', role: 'cancel' },

{

text: 'OK',

handler: (data) => {

this.selectedFigura = data;

this.resultado = null;

},

},

],

});

await alert.present();

}

**Uso de \*ngIf para visualización dinámica. En el HTML del Home**

<div \*ngIf="selectedFigura">

<ion-card \*ngIf="selectedFigura === 'circulo'">

<!-- Contenido del componente Círculo -->

</ion-card>

<ion-card \*ngIf="selectedFigura === 'triangulo'">

<!-- Contenido del componente Triángulo -->

</ion-card>

</div>

**Lógica para el cálculo del perímetro. Si las entradas son invalidas, muestra mensaje de error.**

calcularPerimetro() {

if (this.selectedFigura === 'circulo' && this.radio > 0) {

this.resultado = 2 \* Math.PI \* this.radio;

} else if (

this.selectedFigura === 'triangulo' &&

this.ladoA > 0 &&

this.ladoB > 0 &&

this.ladoC > 0

) {

this.resultado = this.ladoA + this.ladoB + this.ladoC;

} else {

this.resultado = null;

alert('Por favor, ingrese valores válidos.');

}

}

1. *Dificultades encontradas*

Las principales dificultades que encontré fue el integrar los componentes al Home y que no se duplicara la información que estaba escribiendo.

Por otra parte, no logre la actualización automática al correr el proyecto y modificar contenido, descargue server y no fue posible. Tuve que cerrar y abrir cada vez que modificaba, lo que dificulto un poco más saber donde estaba realizando los cambios.

1. *Reflexión final (cómo puedo mejorar, en qué necesito ayuda)*

Requiero seguir estudiando la integración en el Home de las acciones que quiero ejecutar y conocer mejor el lenguaje que se utiliza.

**ENLACE PARA VER PROYECTO**: https://github.com/stephanielazo/Evaluacion1StephanieLazo.git

# Formalidades

* Tipo de letra: Arial 12
* Interlineado: 1,15
* Formato - Entrega del taller en archivo Word o PDF, que lleve como nombre: SuNombre\_SuApellido.
* Subir archivo a la plataforma

# Pauta de evaluación

|  |  |  |
| --- | --- | --- |
| **Criterios de Evaluación** | **Indicadores de Logro** | **Puntaje** |
| Crear una aplicación híbrida funcional utilizando Ionic y Angular  Utilizar la CLI de Ionic de manera efectiva para generar componentes y proyectos | Crea proyecto de Ionic + Angular usando plantilla en blanco para plataformas web | 10 |
| Crea correctamente componente para el Círculo usando CLI o el plugin | 10 |
| Crea correctamente componente para el Triángulo usando CLI o el plugin | 10 |
| Utiliza los componentes de Ionic Framework ion-card, ion-img, ion-select y ion-text para desarrollar las interfaces de usuario | 10 |
| Utilizar HTML5 y Typescript para implementar lógica en la aplicación | Codifica correctamente la jerarquía de clases en Typescript usando herencia | 10 |
| Calcula correctamente los perímetros de las figuras en el método correspondiente | 10 |
| Desarrollar y utilizar componentes básicos en la aplicación para encapsular y reutilizar una funcionalidad | Maneja el evento click de los 2 botones para calcular el perímetro | 5 |
| Muestra el resultado del perímetro una vez se hace click sobre el botón usando la directiva \*ngIf | 5 |
| Utiliza correctamente formularios reactivos o dirigidos por plantilla (template-driven) para recuperar los datos de las figuras | 10 |
| Adjunta el código o proporciona un enlace al proyecto, siguiendo buenas prácticas como: mantener un código ordenado con una correcta indentación, la inclusión de comentarios explicativos, utilización de nombres significativos para variables y métodos, y se asegura de que la carpeta node\_modules no esté incluida. | 10 |
|  | Adjunta el informe en formato Word o PDF, asegurando mantener una ortografía impecable. El informe explica de manera detallada los fragmentos de código relevantes. Se asegura de que el código incorporado esté ordenado y sea fácilmente legible, con un resaltado de sintaxis adecuado. La estructura del informe debe ser coherente, abordando en detalle tanto los problemas que pudo solucionar como aquellos que no pudo resolver. | 10 |
| Puntaje Total | | 100 |

