|  |  |  |
| --- | --- | --- |
| Asignatura: Programación IV | Día y hora: Ma y Ju de 6:20 a 8:00 a.m. | Grupo: 01 |
| Docente: Ing. Rodrigo Ortiz, MIR | **E-mail:** rortiz@ufg.edu.sv | **Aula: N/A** |
| Estudiante:  Ruiz Nativi | **Juan Carlos** | |

Apellidos

Nombres

|  |  |  |  |
| --- | --- | --- | --- |
| **Carrera: Ingeniería en ciencias de la computación** | **Carné:rn100216** | **Fecha:24/11/2020** | **Calificación:** |

**Objetivo: Medir el aprendizaje del estudiante en relación a la utilización de componentes de angular js y ionic.**

**Indicaciones para entregar el ejercicio.**

1. Desarrollar los ejercicios propuestos.
2. Alojar en la plataforma virtual una carpeta comprimida, la cual debe tener como nombre el número de **carnet**.
3. La carpeta debe contener los siguientes archivos:
   1. **Este archivo de Word con sus datos completos.**
   2. El proyecto de IONIC con los requerimientos desarrollados.

El código fuente debe estar debidamente **documentado**.

1. Metodología de evaluación
   1. Archivo de Word ------- 10%
   2. Proyecto de IONIC con los requerimientos desarrollados ------------ 90%.

Indicaciones previas:

Utilice como base el proyecto que desarrollo para el laboratorio 3

Desarrolle los ítems siguientes:

1. Agregue una **opción** (pagina) adicional a su proyecto, con nombre **login**, la cual debe cargarse **por defecto** al momento de lanzar la aplicación (configurada en el app-routing.module.ts).

Tip: Configure path: '' para la página que acaba de crear.

1. La página debe contener un **formulario reactivo** para poder loguearse (**de manera simulada**, **sin** conexión a base de datos).
2. El formulario reactivo debe contener **2 campos de texto** (usuario y password) y **un botón** que envíe la información del formulario a un método o función en su archivo **login.page.ts**.
3. El archivo login.page.ts debe contener:
   1. Un mensaje mandado a la consola de inspección utilizando la función **console.log**() en la que se muestren los datos capturados del formulario, es decir, (Usuario: valor, Password: valor).
   2. Un método o función llamada **DoLogin** la cual debe **validar** los valores capturados del formulario.
      1. Si el **usuario** y **password** son igual a “**admin**” debe redirigir a su página home o principal (utilice la función **this.router.navigateByUrl**(PaginaDestino); si trabaja con **Router**) o (utilice la función **this.navCtrl.navigateRoot**("PaginaDestino"); si trabaja con **NavController**)
      2. Si los valores son diferentes a “admin” debe mostrar una **alerta** (TOAST) indicando que los valores ingresados no son correctos.
4. Agregue las **validaciones** necesarias al formulario para que este **no** permita que el botón **ingresar** este activo sin antes llenar los campos usuario y password.
5. Agregue los estilos **css** necesarios para tener una mejor experiencia de usuario.

Criterios de evaluación:

* Creación de la Página login y configuración (10%)
* Diseño del Formulario de logueo (10%)
* Configuración del Formulario para el envío de datos al método **DoLogin** (10%)
* Captura de datos y envío a consola de inspección (10%)
* Función **DoLogin** con la validación de los datos capturados, reenvió a página principal y alerta con mensaje de error (30%)
* Validaciones al formulario para evitar el envío de campos vacíos (10%)
* Estilos agregados para que el usuario tenga mejor experiencia de usuario (10%)

Ejemplo de formulario:

