**PRUEBA TÉCNICA**

**Nombre Participante:**  Julio José Nieto Durán

**Objetivo:** La prueba consiste en un proyecto para generar un componente en Angular.

**Detalle:** El componente que se requiere es un inicio de sesión para niños el usuario típico se mantiene, pero en lugar de contraseña, el niño debe seleccionar 3 íconos de una lista de 10 íconos (use fontawesome https://fontawesome.com/icons?c=animals), dando clic en cada ícono y el componente le debe mostrar los mensajes de progreso y resultado final.

1. ¿Que se requiere para iniciar el proyecto? (indique lista de archivos mínimos y dependencias)

R:/

* + - 1. Computadora con conexión a internet.
      2. IDE o editor de texto preferiblemente visual studio code.
      3. Node js.
      4. Angular cli.
      5. Typescript.
      6. Bootstrap Framework CSS.
      7. Fontawesome Librería.
      8. Navegador web preferiblemente Google Chrome.
      9. Balsamiq para diseño de mockups de interfaz de componentes.
      10. Lucidchart para diseño de diagrama de flujos
      11. Creacion de proyecto angular con el siguiente comando ng new infant-login.
      12. Iniciar servidor de angular con el comando ng serve para iniciar con el desarrollo.

1. ¿Como sería la estructura del componente? Indique estructura general de métodos y partes del componente en Angular

[R:/](file:///R:/)

Componentes:

* + - Componente login que se comporta como una pagina.
    - Componente home que se comporta como una pagina.
    - Componente navbar que seria el header del componente home muestra un titulo y botón cerrar sesión.
    - Componente password utilizado para la organización de forma aleatoria, iteración y selección de animales (Retorna animales organizados de forma aleatoria para selección).
    - guards para validar la seguridad de las rutas.
    - app-routing.module utilizado para controlar logica de las rutas de la app.

Métodos

* método sort para organizar lista de animales de forma aleatoria
* método setIconPass utilizado para seleccionar animales de forma temporal.
* método noty utilizado para mostrar notificaciones relacionadas con la selección.

Componente navbar

1. closeSesion utilizado para cerrar la sesión y retornar a la ruta login.

Componente login

1. configItems utilizado para obtener ítems retornados por el componente password y asignados para validación de usuario.

2. verifyCredentials utilizado para realizar la validación del usuario y los ítems seleccionados de ser correcta esta validación retorna true y de lo contrario retorna false , para ambos casos muestra una notificación, se encarga de guardar el nombre del usuario en localStorage.

3. noty utilizado para mostrar notificaciones relacionadas con la validación de usaurio.

Componente home

1. Visualiza el navbar y muestra un slide de animales solo es visual

1. ¿Como diseñaría el template para lograr que los íconos se muestren siempre en un orden aleatorio?

R:/

Se organiza lista de animales con el método sort y la función Math.random multiplicada por 10 para organizar el array de forma aleatoria esta es una de las muchas formas que existen para generar un orden aleatorio.

1. ¿Qué usaría para realizar la validación del nombre de usuario? (Indique métodos y componentes de mensajería).

R:/  
1. Input para obtener valor del user.

2. Botón para validación del user.

Componentes

app-login

Métodos  
 método para validar usuario

1. ¿Que requiere modificar si se desea que la lista de íconos aparezca en una ventana modal? (indique componentes requeridos y métodos)

R:/

1. Modal de Bootstrap

2. Botón para abrir el modal.

3. Agregar etiqueta de componente que renderiza los iconos en el modal (Con funcionalidad de selección de ítem).

Componentes

app-password utilizado para renderizar iconos

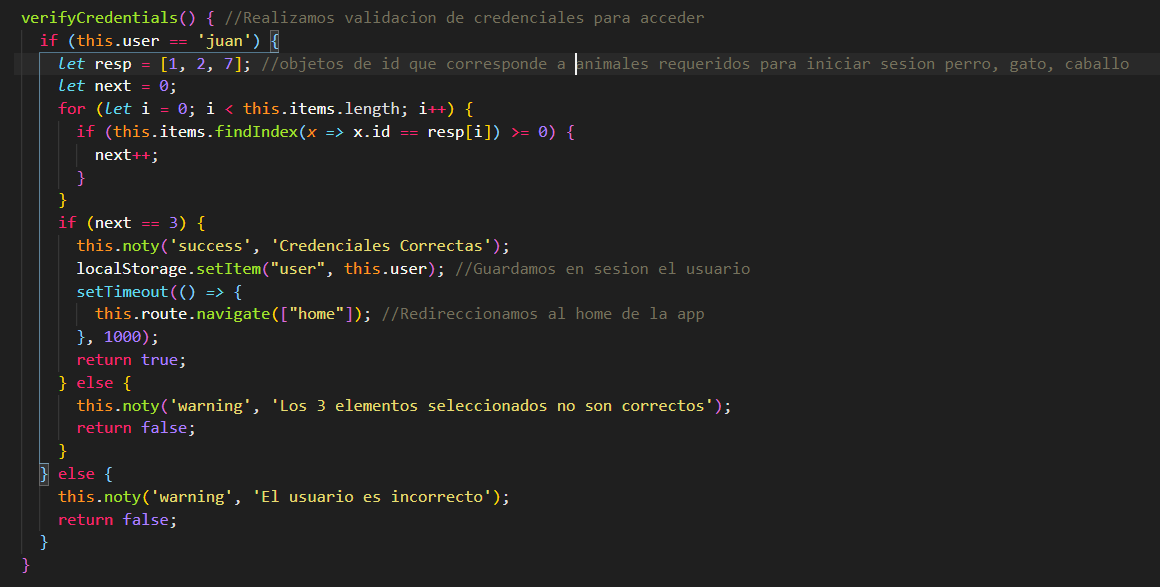
Métodos

método para abrir modal con iconos

método para cerrar modal con iconos

1. Realice un método Mock que valide al usuario "juan" y la contraseña sean los íconos "Gato, Perro y Conejo"

R:/



1. Validamos el usuario si es igual a ‘juan’ seguimos con el proceso si es diferente se muestra una notificación ‘El usuario es incorrecto’

2. Si el usuario es diferente a ‘juan’ mostramos notificación ‘El usuario es incorrecto’ y retornamos false.

2. Iteramos ítems seleccionados y los comparamos con los ítems requeridos para iniciar sección ‘perro, gato, caballo’ , si son iguales a los ítems seleccionados vamos aumentando un contador en 1.

3. Comparamos el valor que tiene el contador ‘next’ si es == 3 ,mostramos notificación ‘Credenciales Correctas’ , guardamos el user en localStorage, por ultimo hacemos re dirección a la ruta home y retornamos true.

4. Si el contador ‘next’ es <> 3 mostramos notificación ‘Los 3 elementos seleccionados no son correctos’ y retornamos false.

1. Realice el método de validación (que consuma el mock del punto anterior)

R:/

Método validar usuario

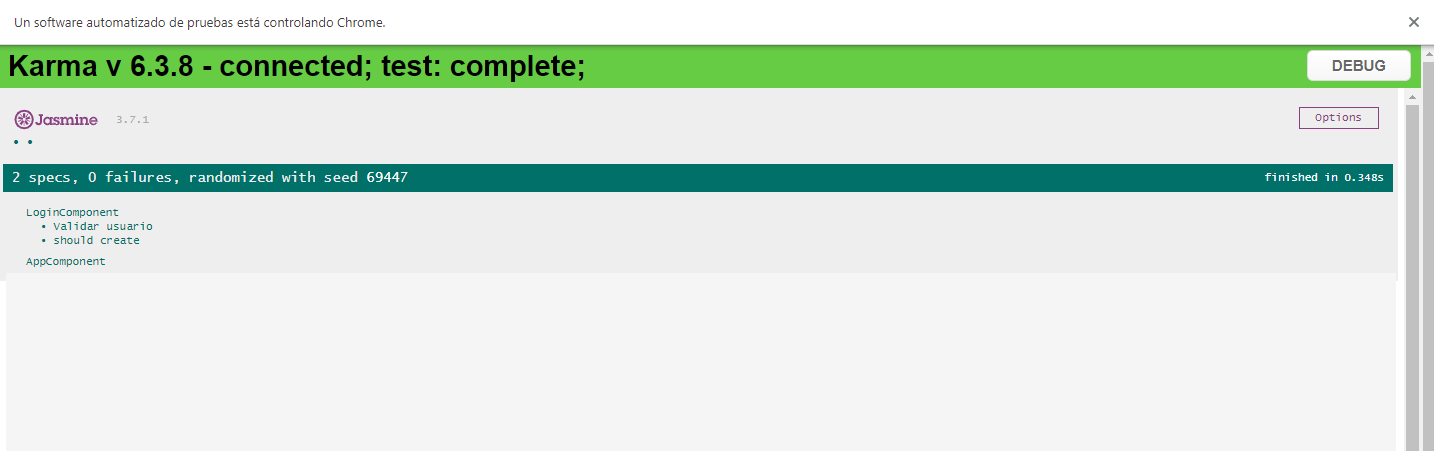


Se pasa por defecto el user

Se pasa por defecto los items para evitar la selección

Se aplica el expect.toBe() llamando la funcion verifyCredentials la cual retorna true o false y se compara con true o false para realizar la prueba del mock.

Resultado exitoso del mock



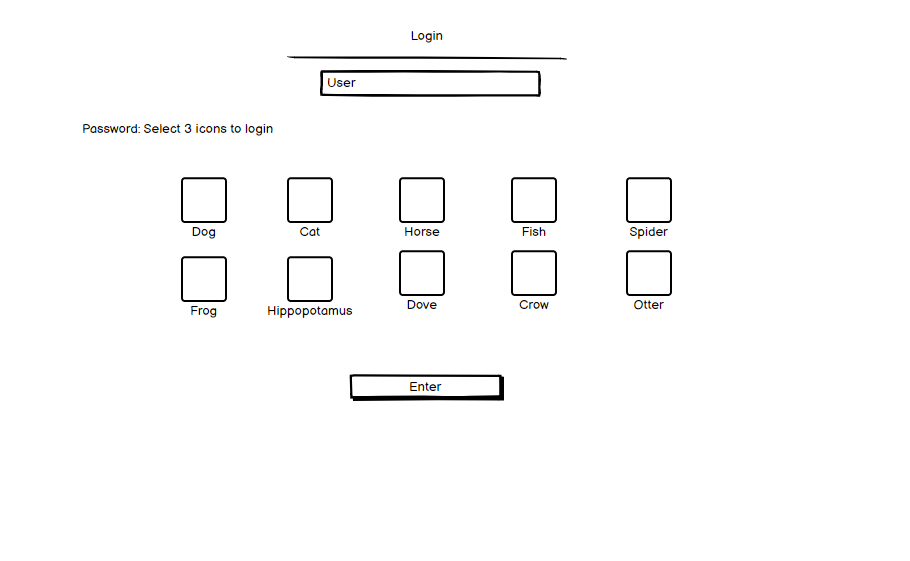
Resultado incorrecto del mock



1. Realice la documentación del proyecto, la que considere necesaria.

R:/

Mockups



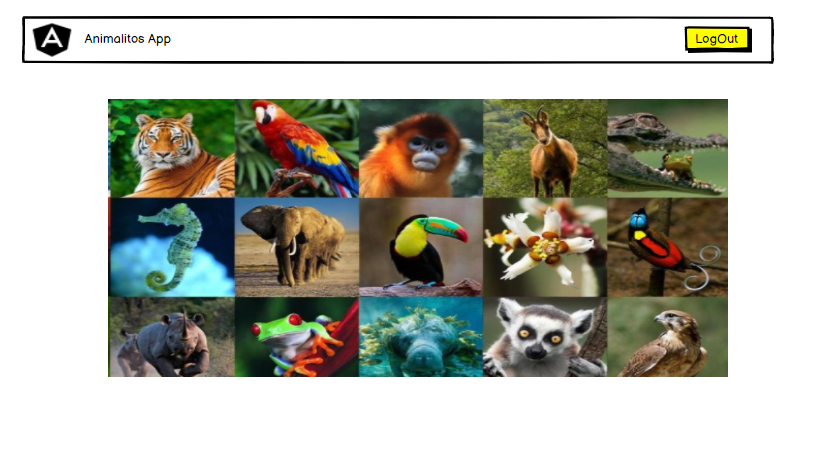
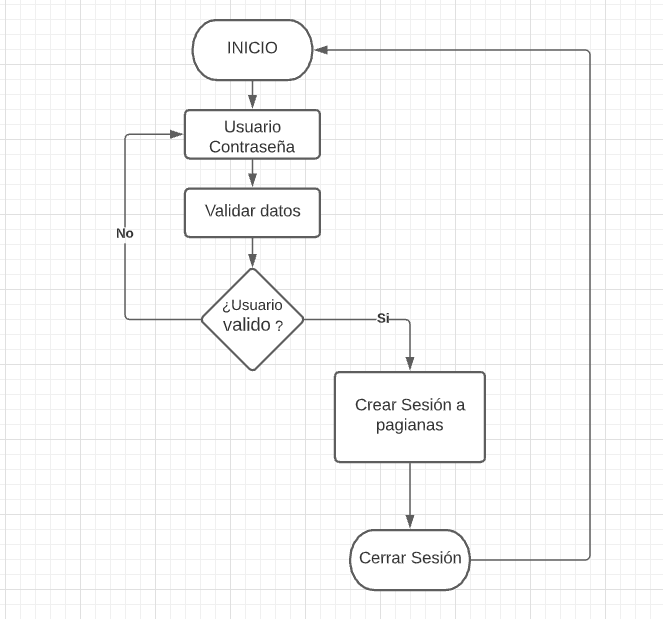
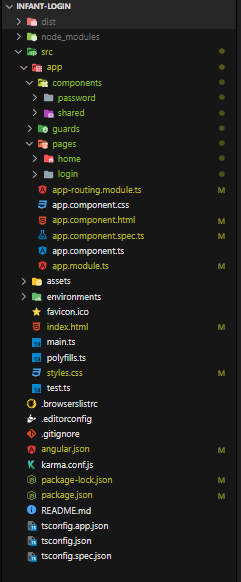


Diagrama de flujo



Estructura de proyecto angular



1. ¿Como se realiza el bundle de la app?

1. Se abre la consola dentro la carpeta del proyecto

2. Se ejecuta el siguiente comando ng build para generar el bundle de la app lo cual como resultado final devuelve una carpeta llamada dist y dentro retorna una carpeta llamada login-infant con el bundle de la app, el cual se puede utilizar para publicar la aplicación en un hosting o servidor

**Nota:**

1. Para ejecutar proyecto abrir un editor de texto y venta de comandos y ejecutar el comando npm install , para realizar descarga de las dependencias del proyecto.

2. Para ejecutar las pruebas unitarias ejecutar en la raíz del proyecto el comando ng test y se abre un venta del navegador con la validación de las pruebas.