El componente **AuthPage** es el responsable de la autenticación del usuario. Este componente utiliza un formulario para capturar el correo electrónico y la contraseña, los cuales son validados para asegurarse de que cumplen con los requisitos establecidos.

Cuando el usuario envía el formulario, se invoca la función **submit().** Esta función verifica si el formulario es válido y, de ser así, realiza una solicitud de inicio de sesión a través del servicio **FirebaseService.** Si la autenticación es exitosa, se obtiene la información del usuario mediante la función **getUserInfo()**, la cual consulta la base de datos de **Firebase** para obtener detalles adicionales del usuario.

En caso de éxito, la información del usuario se almacena en el almacenamiento local y se le redirige a la página principal.

Dentro del código, existen conexiones con dos servicios: **FirebaseService** y **UtilsService**. En este caso, el método **loading()** se utiliza para mostrar un indicador de carga en la interfaz del usuario, con el fin de informar que la aplicación está trabajando en segundo plano.

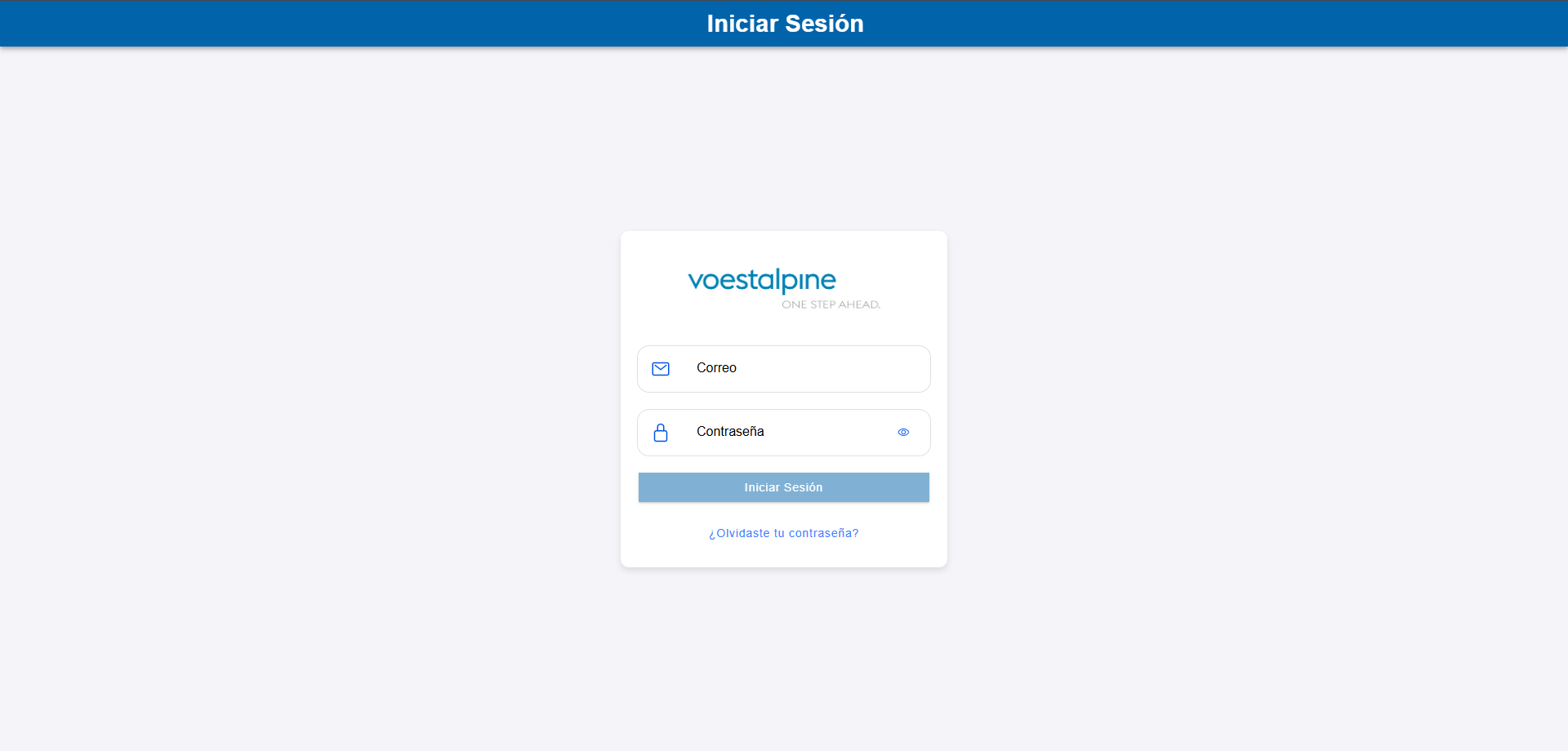
Luego, está el método **presentToast()**, el cual se emplea para mostrar un mensaje emergente en la interfaz con el propósito de notificar al usuario sobre algún evento o acción, como un mensaje de éxito, error o advertencia.

Otra conexión con un servicio es el método **SignIn**, el cual se llama desde **FirebaseService** y se utiliza para autenticar al usuario en **Firebase** mediante su correo y contraseña.

Luego, el método **getDocument()** se empleará para obtener información específica desde **Firestore**, en este caso, para recuperar los datos de un usuario.

Por último, tenemos el código de la interfaz gráfica con sus estilos, en el cual se tomó en cuenta un diseño simple pero agradable, utilizando una paleta de colores acorde con la identidad visual de la empresa.

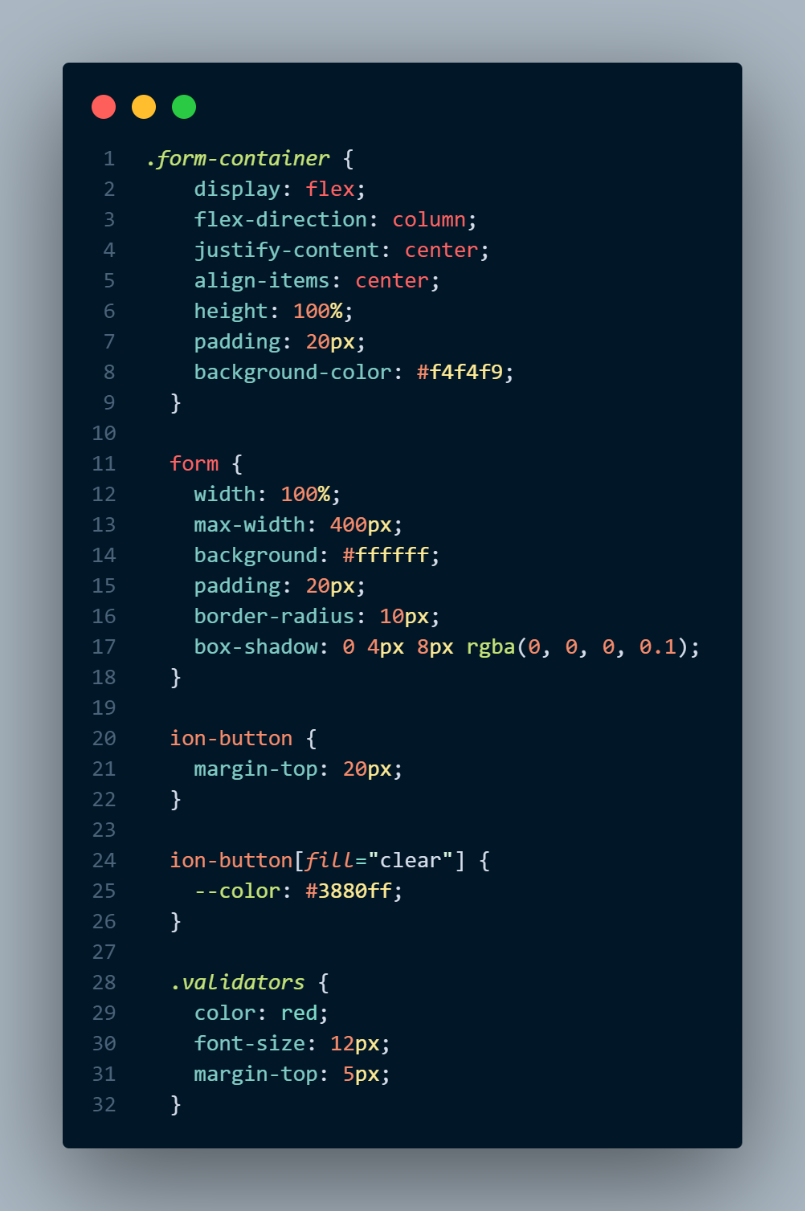


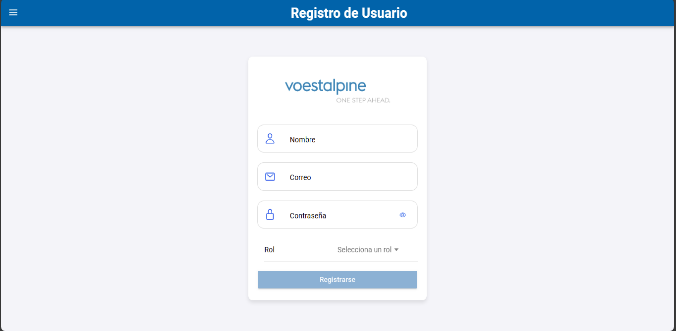


El componente SignUpUserPage es una página de registro diseñada para permitir a los administradores crear nuevos usuarios. Este componente resulta útil en escenarios donde un administrador o un usuario con permisos especiales necesita registrar a otros usuarios manualmente.

El formulario de registro captura información como el correo electrónico, la contraseña, el nombre y el rol del usuario, aplicando validaciones específicas en cada campo. Cuando el formulario es enviado, se ejecuta el método submit(), el cual primero verifica si el formulario es válido y, como segunda verificación, requiere que el usuario con rol de administrador ingrese su contraseña.

Este componente tiene una conexión con FirebaseService, donde el método correspondiente tiene como objetivo crear un nuevo usuario en Firebase sin que la sesión del usuario actual se cierre. Primero, solicita los datos del usuario a registrar y, al presionar Enviar, requiere la contraseña del usuario actual para verificar su identidad.

Por último, se encuentra el aspecto de la interfaz gráfica, donde se detallan todos los elementos necesarios para el ingreso de los datos del usuario.



El componente de HomePage es la página principal de la aplicación en donde aquí se majea la lógica para tener estadísticas de postulantes como el total de candidatos, postulantes aprobados, en estrevista etc, en el cual aparte de eso el listado de todos y recomendaciones que hace dependiendo de el estado en el que se encuentra el postulante.

Partes importantes del código es que en este componente utilizamos el ngOnInit el cual este se encarga de cargar los curriculums desde firebase, otro importante es para los filtros en el cual utilizamos en applyFilters en el cual va a filtrar candidatos según la búsqueda y el estado relacionado

El componente HomePage es el encargado de gestionar y mostrar la lista de candidatos. Este componente se conecta con el servicio FirebaseService para obtener los datos de los currículums almacenados en Firebase y permite a los usuarios filtrar y visualizar a los candidatos según su estado (por ejemplo, Pendiente de Revisión, Aprobado, Rechazado, etc.).