

Proyecto Ingeniería Web y Móvil

VersaAssist

Integrantes:

- Sofía Labra
- Maite Villalón
- Sebastian Soza
- Paul Rojas



Lista de Figuras

Figura 1 - Bienvenida	4
Figura 2 - Inicio Sesión	5
Figura 3 - Crear cuenta	6
Figura 4 - Dashboard	7
Figura 5 - Editar Perfil	8
Figura 6 - Información Robot	9
Figura 7 - Asistente Deportivo	10
Figura 8 - Chat Asistente Deportivo	11
Figura 9 - Chat Preguntas	12
Figura 10 - Formulario Validación 1	13
Figura 11 - Formulario Validación 2	14
Figura 12 - Mecanismo de Validación	15
Figura 13 - Lectura Json	16
Figura 14 - Lectura Json	16
Figura 15 - OHBOT información	17
Figura 16 - Patrones de Lectura	21
Figura 17 - Patrones de Lectura	22
Figura 18 - Patrón de Navegación	23



Introducción

Hoy en día, la tecnología se ha vuelto parte esencial de nuestra vida diaria, ofreciéndonos formas más avanzadas de interactuar con el mundo que nos rodea. A medida que buscamos simplificar nuestras tareas y mejorar nuestra calidad de vida, surge la necesidad de crear aplicaciones que combinen lo básico con lo innovador, utilizando la inteligencia artificial para facilitar la interacción entre humanos y máquinas. Es por esto que se presenta nuestra aplicación basada en Ionic, una plataforma versátil y robusta que proporciona una experiencia de usuario fluida y receptiva en múltiples dispositivos.

Uno de los elementos destacados de nuestra aplicación es la integración de un robot de servicio, diseñado para interactuar con el usuario de manera inteligente y personalizada. Este robot ofrece una variedad de servicios, desde recordatorios de tareas y citas hasta asesoramiento emocional. Con la capacidad de comprender el lenguaje natural y expresar emociones a través de su interfaz facial, el robot brinda una experiencia de usuario única y enriquecedora.

Además, nuestra aplicación proporciona acceso a información detallada sobre el robot, incluidos sus datos ambientales, historial de interacciones e historial de búsqueda y navegación. Esto permite al usuario comprender mejor el funcionamiento del robot y maximizar su utilidad en diferentes contextos.



P1.1 Análisis y diseño de funcionalidades (al menos 5).

Nuestro proyecto consta con las funcionalidades base de una aplicación, como:

- **Inicio de sesión:** Página en la cual el usuario podrá ingresar a la aplicación ingresando sus datos de ingreso, como nombre de usuario y contraseña.
- **Crear cuenta:** Para los nuevos usuarios existe la opción de registro, para un registro exitoso los usuarios deben ingresar los datos requeridos, nombre de usuario, contraseña, correo electrónico, región, comuna, confirmación de contraseña y además se deben aceptar los términos y condiciones. En esta sección, se da la opción de regresar a inicio de sesión en caso de ya contar con una cuenta,
- **Editar perfil:** Consiste en una sección donde el usuario podrá modificar los datos que tiene almacenados, como su nombre de usuario, su nombre, apellido y su fecha de nacimiento.
- **Introducción y Presentación del robot,** presenta algunos datos del robot como tipo de robot, nombre y sensores. Además da la opción de interactuar con el robot
- **Dashboard:** Página principal en la cual se ofrecen los servicios principales del robot como, acceder a recordatorios, asesoramiento emocional, actividad física y ejercicios, además muestra actividades recientes (estas funciones aún no son implementadas).

A la vez, nuestra aplicación cuenta con funcionalidades específicas del robot,

- **Recordatorios de tareas y citas**
- **Asesoramiento emocional**
- **Asesoramiento deportivo**
- **Interacción con robot (chat)**
- **Entrega información del robot,** como sus datos del entorno, su historial de interacciones e historial de búsqueda y navegación

EP1.2 Prototipado en Figma de al menos 5 UI (pantallas), que representen las funcionalidades propuestas. Se deben considerar 5 funcionalidades y diseños diferentes, y las UI deben ser prototipadas utilizando interfaces móviles.

Interfaz de bienvenida, se presenta una pequeña descripción.



Figura 1: Bienvenida

Interfaz inicio de sesión, se deben ingresar datos de un usuario registrado.



The image shows a mobile application interface for logging in. At the top, there is a status bar with the time 11:11, signal strength, Wi-Fi, and battery icons. Below the status bar is a back arrow icon. The main content area features the VERSAASSIST logo, which consists of a stylized purple robot head above the text "VERSAASSIST" in purple. Below the logo is the heading "Iniciar de sesión" in bold black text. There are two light blue input fields: "Nombre de usuario" and "Contraseña". Below these fields is a purple button with the text "Iniciar sesión". At the bottom, there is a link "¿Aún no tienes cuenta?" and a text "Crear una nueva cuenta".

Figura 2: Inicio de sesión

Interfaz de Crear cuenta, se piden datos para registrar al usuario.



11:11

VERSAASSIST

Crear Cuenta

Nombre de usuario

Correo Electronico

RUT

Seleccione su región

Seleccione su comuna

Contraseña

Confirmación Contraseña

Acepto los términos y condiciones

Registrar

¿Ya tienes cuenta?

Iniciar sesión

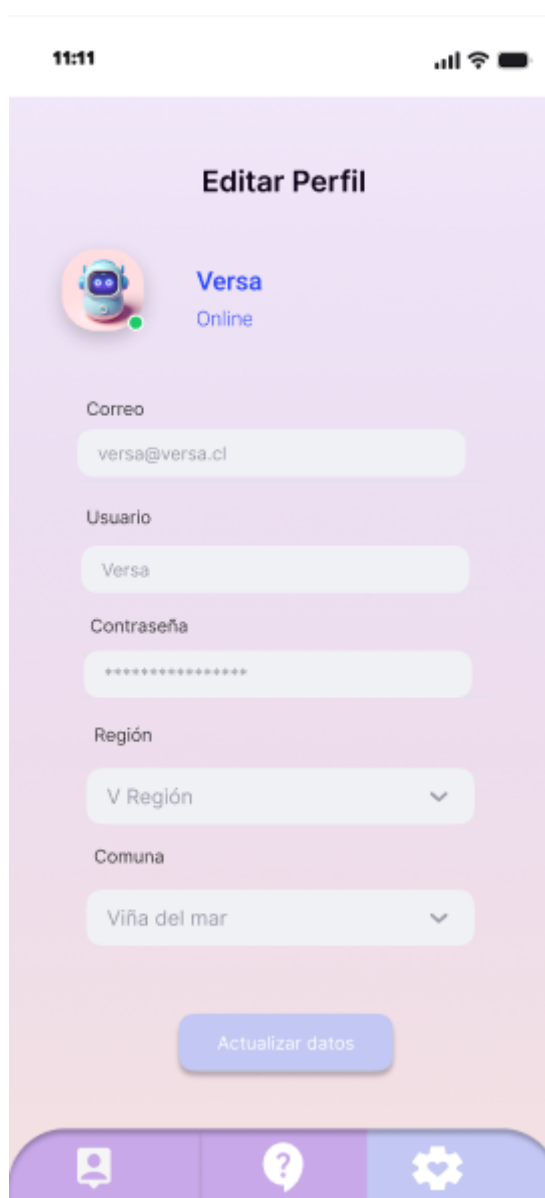
Figura 3: Crear Cuenta

Interfaz de Dashboard principal, opciones principales.




Figura 4: Dashboard

Interfaz de Editar perfil, se puede editar datos ya registrados.



11:11

Editar Perfil

 **Versa**
Online

Correo
versa@versa.cl

Usuario
Versa

Contraseña

Región
V Región

Comuna
Viña del mar

Actualizar datos

Figura 5: Editar perfil

Interfaz de información del robot, se muestran opciones de visualización.

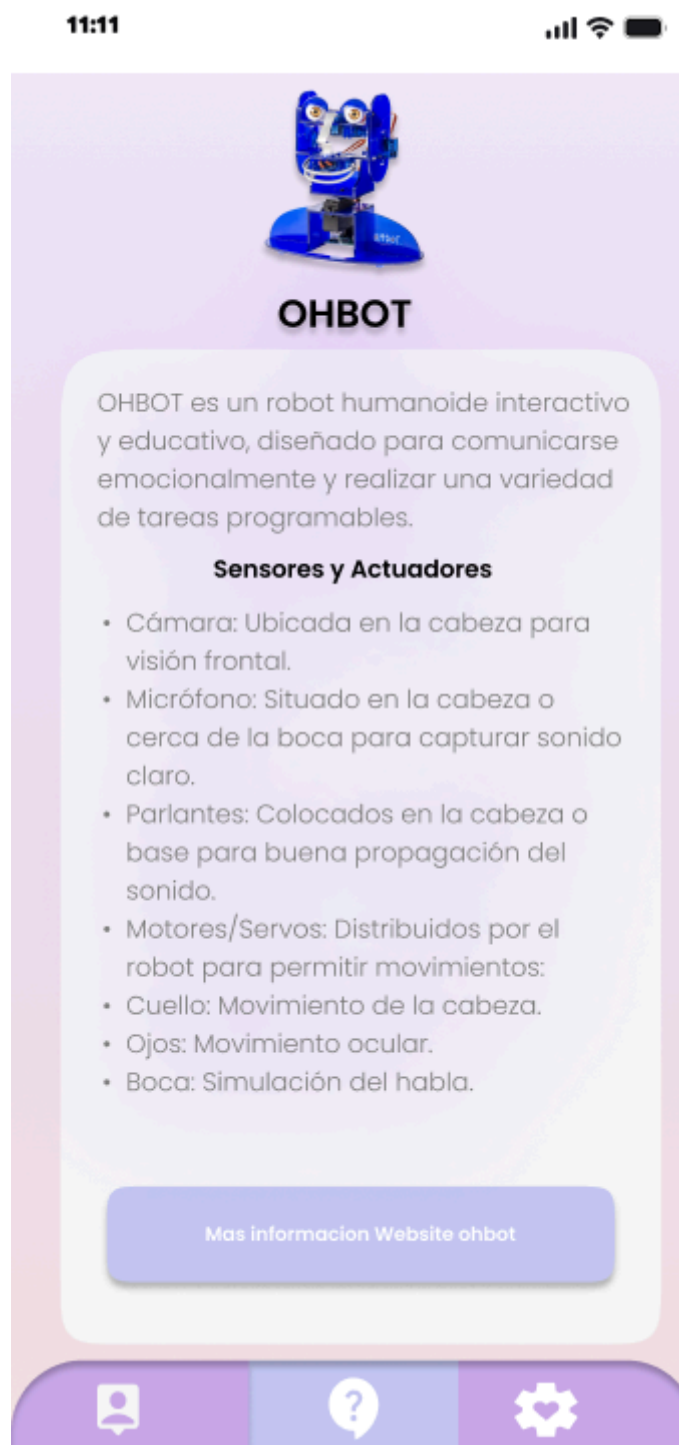


Figura 6: Información del Robot

Interfaz de Asesor deportivo, introducción para iniciar conversación.

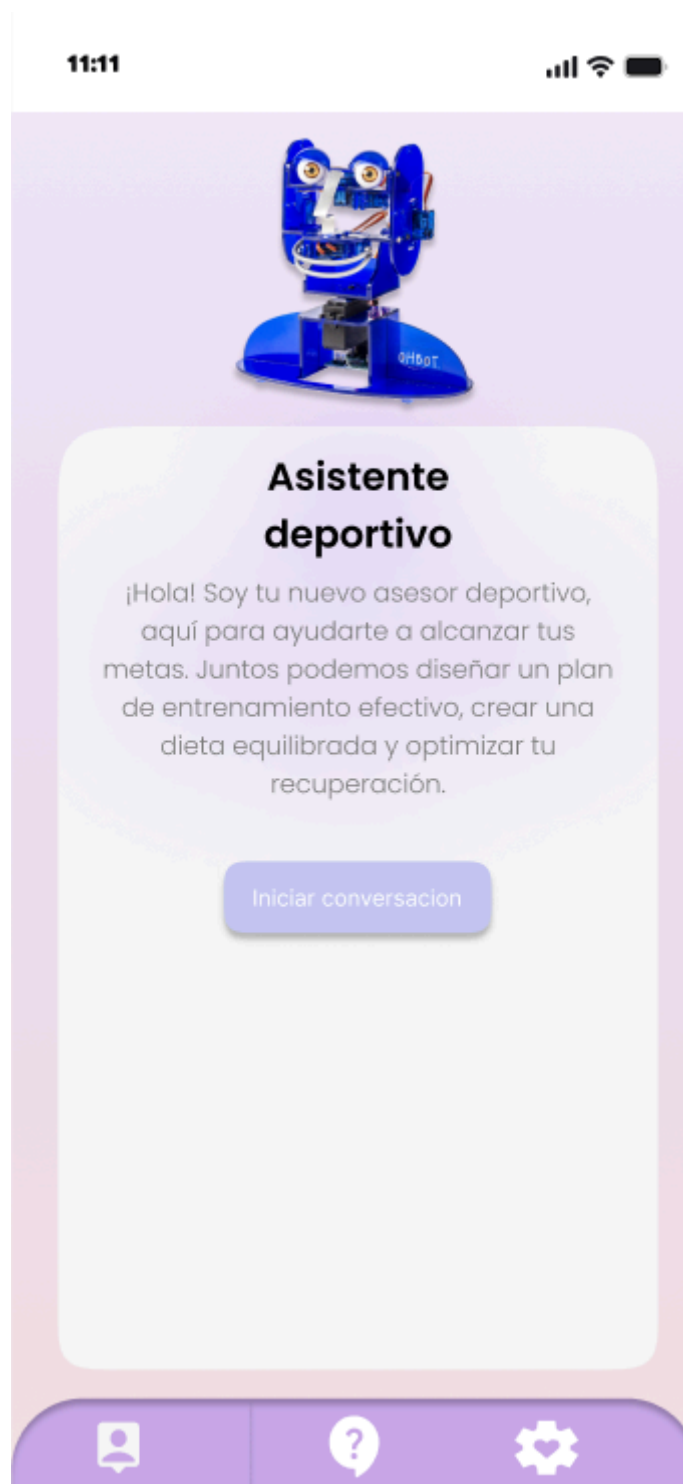


Figura 7: Asistente deportivo

Interfaz de Asesor deportivo, Chat 1



Figura 8: Chat asistente deportivo

Interfaz de Asesor deportivo, Chat Preguntas



Figura 9: Chat preguntas

EP1.3 Maquetación responsiva de las 5 UI utilizando HTML y CSS. Se puede utilizar preprocesadores de CSS, o librerías de diseño como Bootstrap o Tailwind CSS.

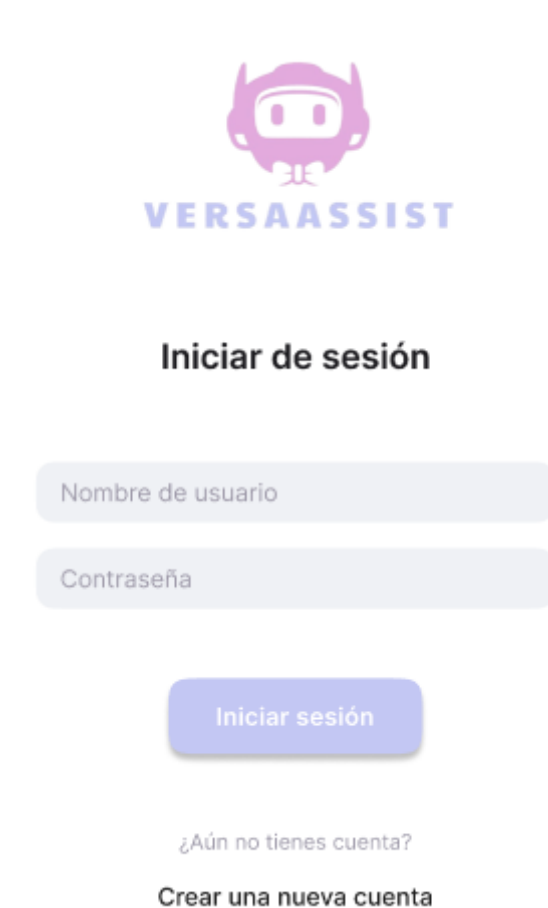
Implementación en código.

EP1.4 Incluir dos formularios relacionados a la sesión de usuarios (inicio de sesión y registro), considerando los campos: Nombre de usuario (username),

RUT, Correo Electrónico (email), Región, Comuna, Contraseña, Confirmación de contraseña y Aceptación de términos y condiciones.

Implementación en código,

Formulario 1:



The image shows a login form for 'Versaassist'. At the top, there is a logo consisting of a stylized purple robot head above the word 'VERSAASSIST' in blue capital letters. Below the logo, the text 'Iniciar de sesión' is centered. Underneath, there are two light gray input fields: the first is labeled 'Nombre de usuario' and the second is labeled 'Contraseña'. Below these fields is a purple button with the text 'Iniciar sesión'. At the bottom, there is a link that says '¿Aún no tienes cuenta?' followed by the text 'Crear una nueva cuenta'.

Figura 10: Formulario 1

Formulario 2:



Crear Cuenta

▼

▼

☒ Acepto los términos y condiciones

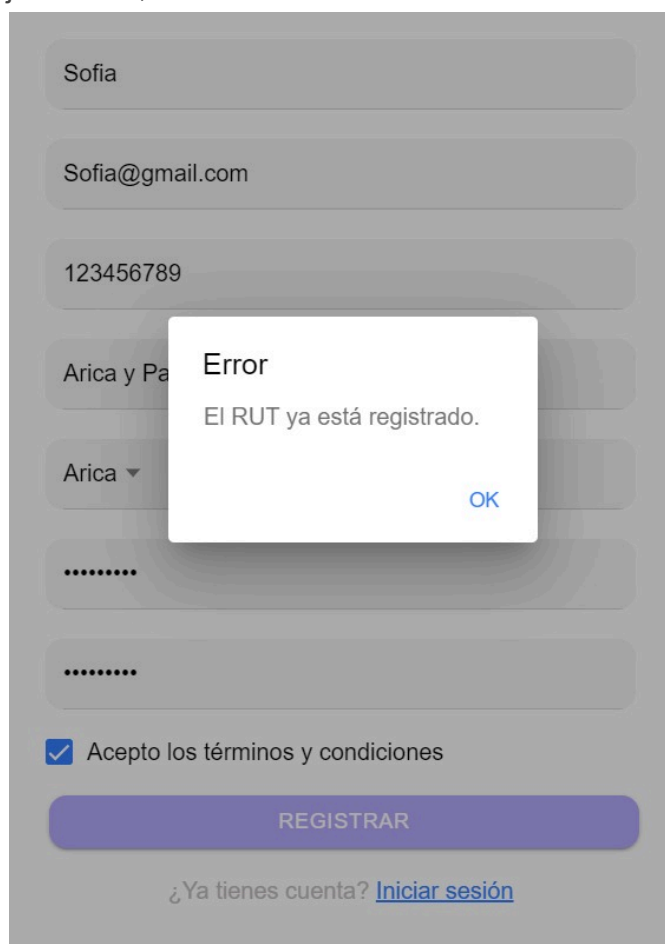
[¿Ya tienes cuenta?](#)

[Iniciar sesión](#)

Figura 11: Formulario 2

EP1.5

Mecanismo de Validación, el usuario intenta hacer un registro con un rut ya ingresado. Se despliega mensaje de error,



The image shows a registration form with the following fields: Name (Sofia), Email (Sofia@gmail.com), RUT (123456789), Address (Arica y Pa...), City (Arica), Password (masked with dots), and Confirm Password (masked with dots). A checkbox for 'Acepto los términos y condiciones' is checked. A purple 'REGISTRAR' button is at the bottom. An error message dialog box is overlaid on the form, displaying 'Error' and 'El RUT ya está registrado.' with an 'OK' button. At the bottom of the form, there is a link: '¿Ya tienes cuenta? [Iniciar sesión](#)'.

Figura 12: Mecanismo de validación

EP2.1 Implementación de las 5 UI en el framework Ionic.

Implementación en código para framework Ionic.

EP2.2 Hacer lectura de datos desde un archivo JSON (puede ser local, o alguna fuente externa de datos), y mostrarlos en alguna de las pantallas.

```
{
  "components": [
    {
      "name": "Cámara",
      "description": "Ubicada en la cabeza para visión frontal"
    },
    {
      "name": "Micrófono",
      "description": "Situado en la cabeza o cerca de la boca para capturar sonido claro"
    },
    {
      "name": "Parlantes",
      "description": "Colocados en la cabeza o base para buena propagación del sonido"
    },
    {
      "name": "Motores/Servos",
      "description": "Distribuidos por el robot para permitir movimientos",
      "subcomponents": [
        {
          "name": "Cuello",
          "description": "Movimiento de la cabeza"
        },
        {
          "name": "Ojos",
          "description": "Movimiento ocular"
        },
        {
          "name": "Boca",
          "description": "Simulación del habla"
        }
      ]
    }
  ]
}
```

```
<p>
  <ng-container *ngFor="let component of componentsData">
    - {{ component.name }}: {{ component.description }}<br>
    <ng-container *ngIf="component.subcomponents">
      <ng-container *ngFor="let subcomponent of component.subcomponents">
        &nbsp;{{ subcomponent.name }}: {{ subcomponent.description }}<br>
      </ng-container>
    </ng-container>
  </ng-container>
</p>
```

Figura 13-14: Lectura Json

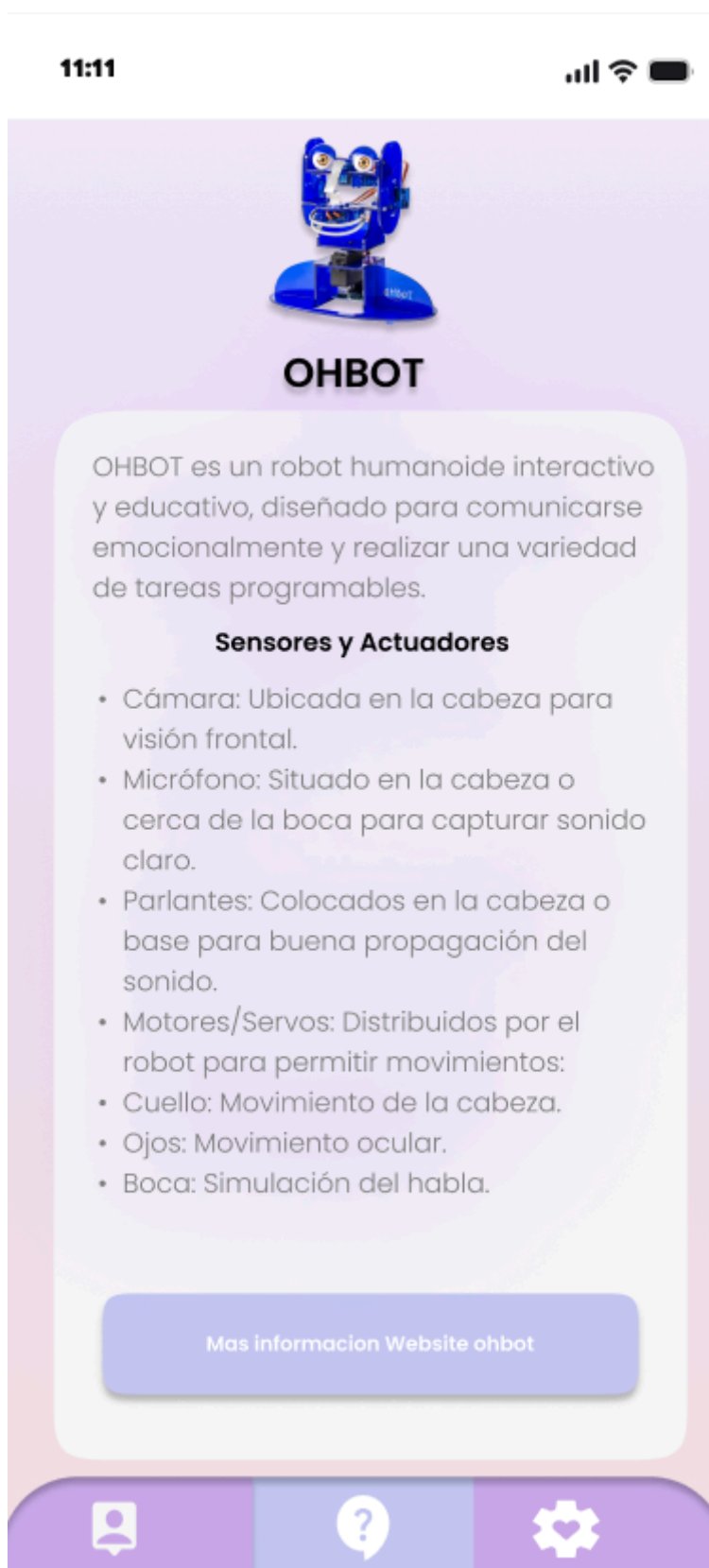


Figura 15: OHBOT información



EP2.3 Definir estructura de datos inicial de la aplicación, y definir la estructura de la base de datos, y el modelo de datos. Algunos motores de bases de datos que se pueden utilizar son: MySQL, PostgreSQL, SQLite, MongoDB, Firebase, entre otros.

Ocuparemos Firebase (con Firestore como base de datos), debido que contiene sincronización en tiempo real en todas las plataformas y la idea de nuestro proyecto, es que sea apta para android e ios, contiene una configuración sencilla y rápida. Contiene firebase authentication, lo cual nos servirá para gestionar la autenticación de usuarios, contiene funciones en la nube y permite el almacenamiento de archivos.

Por lo mismo, se propone la siguiente estructura para las siguientes funcionalidades:

- Autenticación de Usuarios: Cada usuario podrá registrarse con los siguientes datos: nombre de usuario, email, rut, región, comuna y contraseña. Y pueden iniciar sesión con los datos, nombre de usuario y contraseña.
- Perfil de usuario: Cada usuario podrá ver y modificar sus datos.
- Información de OHBOT: Los usuarios podrán ver la descripción del robot.
- Interacción con el asistente deportivo: Los usuarios podrán interactuar con el robot en su formato asistente deportivo, donde podrán tener conversaciones con él. Las conversaciones serán registradas y almacenadas en la base de datos .
- Registro de conversaciones: Todas las conversaciones entre el usuario y OHBOT, como se mencionó anteriormente serán registradas para poder ser visualizadas posteriormente si se desea.



Estructura de la base de datos

1. Users

- Documento : "userId"
 - "username" : String
 - "email" : String
 - "rut" : String
 - "region" : String
 - "comuna" : String
 - "password" : String
 - "conversaciones" : Array

2. Robots

- Documento : "robotId"
 - "nombreRobot" : String
 - "descripcion" : String
 - "estado" : String

3. Conversaciones

- Documento : "conversacionesId"
 - "userId" : String
 - "robotId" : String
 - "mensajes" : Array
 - "mensaje" : String
 - "enviado" : String (En este punto, se dice el mensaje fue enviado por el userId o robotId)
 - "datoMensaje" : timestamp (Se registra fecha y hora del mensaje)



Ejemplo de estructura en Json

```
{
  "users":
  {
    "userId1:{
      "username" : "Sebastian",
      "email" : "ssoza@gmail.com",
      "rut" : "12345678-9",
      "region" : "Valparaíso",
      "comuna": "ViñaDelMar",
      "password": "holasoyseba",
      "conversaciones": ["conversacionId1", "conversacionId2"]
    }
  },
  "robots":{
    "robotId1":{
      "nombre": "Deportivo",
      "descripcion": "Asistente deportivo, puede ayudarte en tu progreso",
      "estado": "activo"
    }
  },
  "Conversaciones":{
    "conversacionId1":{
      "UserId": "UserId1",
      "robotId": "roborId1",
      "mensajes":[
        {
          "mensaje": "Como estas? en que te puedo ayudar hoy?",
          "enviado": "robot",
          "datos": "2024-05-06T12:06:30Z"
        },
        {
          "mensaje": "Necesito una rutina de hombro",
          "enviado": "usuario",
          "datos": "2024-05-06T12:07:00Z"
        }
      ]
    }
  }
}
```

EP2.4 Hacer uso de al menos dos (2) patrones de diseño, ya sea web o móvil, en la implementación de las pantallas, teniendo como foco principal el uso desde un dispositivo móvil.

En nuestro trabajo encontramos dos patrones de diseño:

1. Patrones de Lectura: En nuestro proyecto podemos notar que en las pantalla de “Iniciar Sesión” y “Crear cuenta” ocurre un error si los datos ingresados son inválidos, en el cuadro de mensaje en vez de poner tal cual el error (“Error 234”), le mencionamos al usuario el porqué del error “Error, correo o contraseña incorrectos”, “Error, todos los campos son obligatorios”, “La contraseña debe tener más de 6 caracteres”, “Debe aceptar los términos y condiciones”

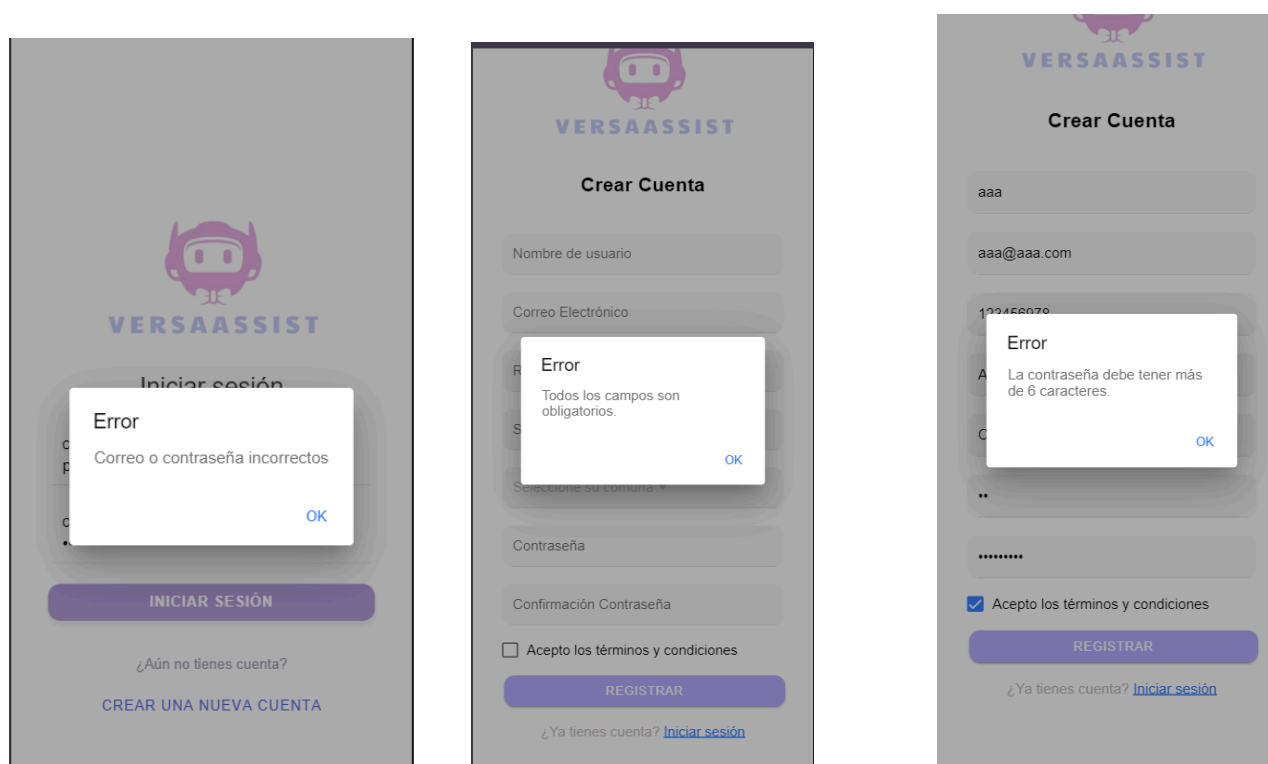


Figura 16: Patrones de Lectura

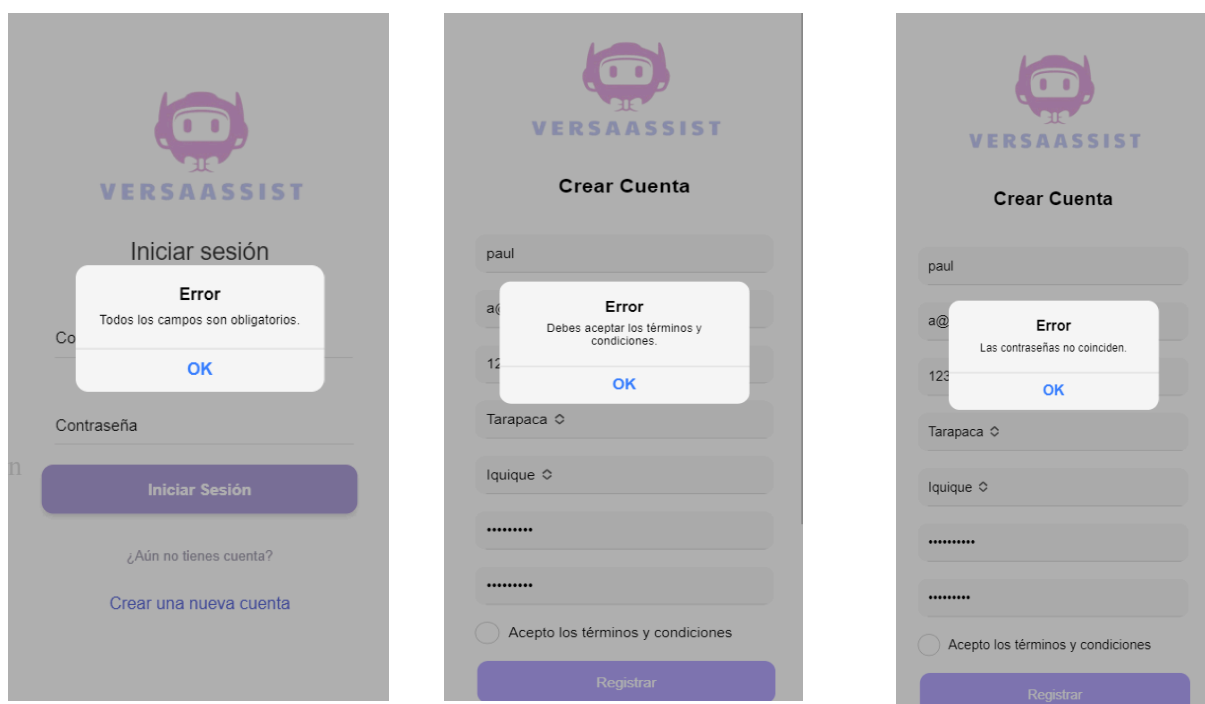


Figura 17: Patrones de Lectura

2. Patrón de Navegación: En nuestro proyecto podemos encontrar una tab bar, la encontramos en todas las pantallas, menos la de bienvenida, inicio sesión, crear cuenta. la tab bar nos permite acceder de manera rápida a otras pantallas.

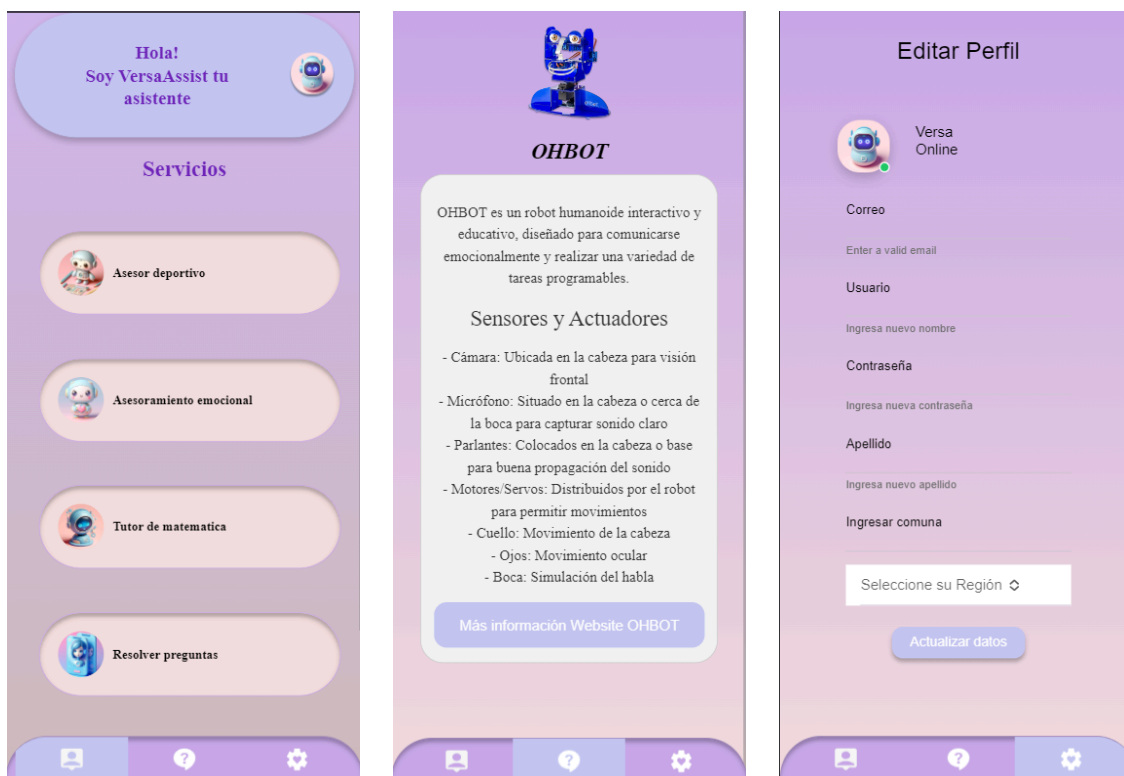


Figura 18: Patrón de Navegación