



Presentació Técnic FitHub

Aitor, Alex, Eric i Miquel

Índex

Arquitectura de l'aplicació

Esquema de la nostra arquitectura

01



Rutes de l'aplicació

Resum de les rutes que s'utilitzen en la nostra pagina web

02



Esqema base de dades

Mostrem esquema de la base de dades i resum de que fa cada part

03



04

Esquema de components

Mostrem l'esquema de components i resum de que fa cada cosa



05

Problemes i Solucions

Explicar els nostres problemes i les solucions trobades

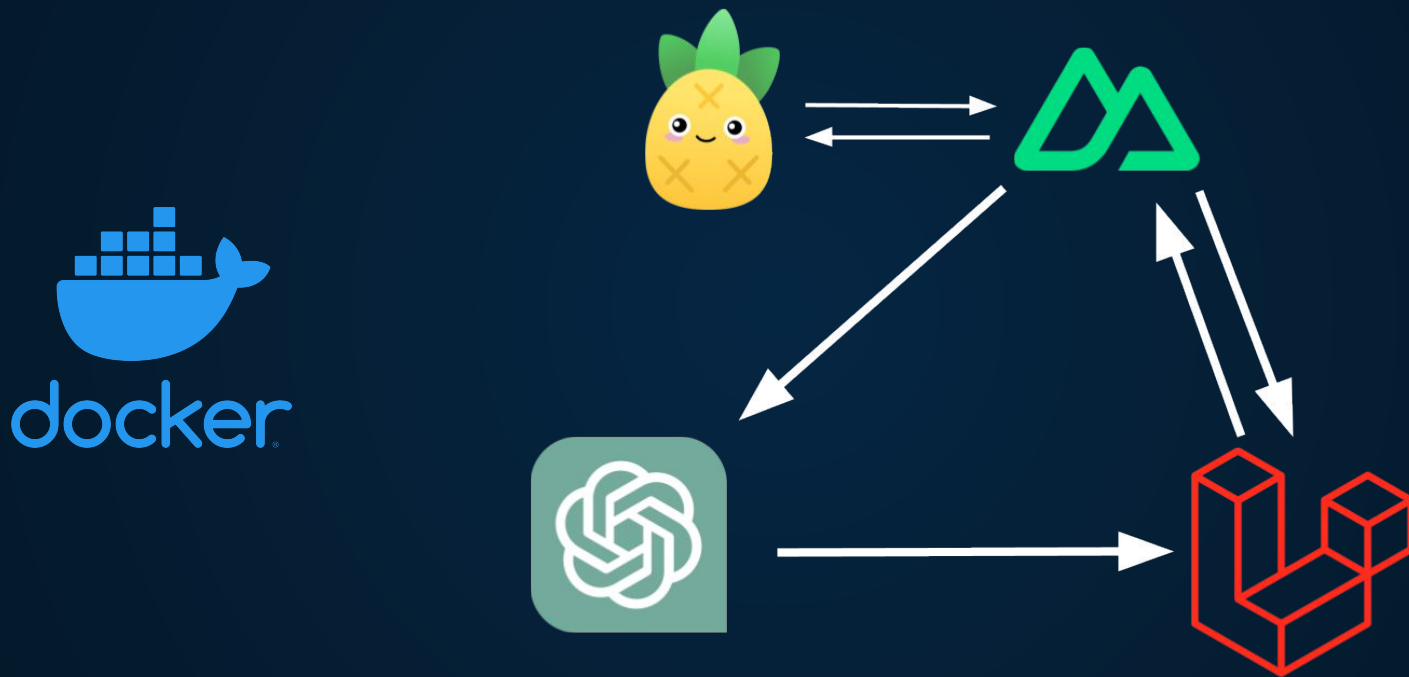


06

Aspectes Tècnics

Here you could describe the topic of the section

Arquitectura de la nostra aplicació



Rutes de l'aplicació

Getters

getDatosUsuario
getTotosUsuaris
getUsuariosChat:
getDatosEjercicio
getDatosAliments
getDieta
mostrarUltimoMensajeEllos

Post

iniciarSesion
enviarRutinaAlServidor
enviarDietaAlServidor
enviarSolicitudAmistad

Put

actualizarDatosUsuario

Delete

borrarRutinaDia
borrarDietaHoy
borrarUsuari
deleteRutinaByDate
deleteDietaByDate

API Chat GPT

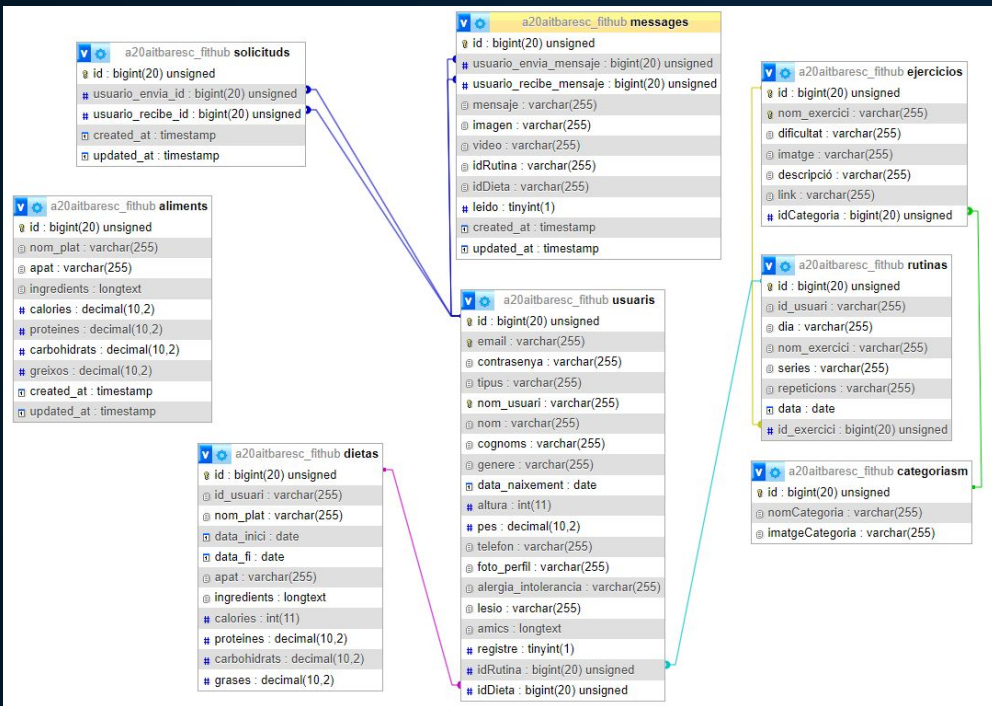
enviarMensajeOpenAIRutina.
enviarMensajeOpenAIDIeta
enviarMensajeOpenAI

Esquema base de dades

La nostra aplicació es centra en l'usuari, permetent-li gestionar rutines i dietes personalitzades. Cada usuari pot tenir múltiples rutines i dietes, vinculades per l'ID d'usuari.

També poden enviar i rebre sol·licituds d'amistat i missatges, que es guarden amb l'ID de l'emissor i receptor per mantenir un historial del xat.

Les rutines inclouen diversos exercicis, i aquests es classifiquen en categories, facilitant la gestió i personalització. Tot i que les dietes no estan directament enllaçades amb els aliments, aquests s'utilitzen per generar les dietes a partir de la base de dades d'aliments.



Esquema de components frontend



Pages

Les pages contenen totes les pàgines del projecte, utilitza la carpeta public per carregar elements visuals. També utilitza stores per gestionar crides a la API i l'estat de l'aplicació amb pinia. Inclouen components com la navbar o la capçalera, que es mostren a la majoria de pàgines.

Components

Els components són elements reutilitzables inclosos dins les pàgines, també utilitzant stores per obtenir dades.

- Capçalera: Header general
- HeaderPerfil: Header específic per al xat, amb funcions per gestionar sol·licituds d'amistat.
- HeaderChat: Header per a la pàgina de perfil, permet canviar la foto de perfil.
- navBar: Barra de navegació a la part inferior amb enllaços a les altres pàgines.

Stores

Contenen dos arxius principals:

- pinia (index.js): Gestiona l'estat global de l'aplicació.
- communicationManager.js: Conté totes les crides fetch al Laravel i a la API de Chat GPT

Public

Inclou imatges, logos i icones utilitzades per les pages i components.

Problemes i solucions

Implementar sockets al
chat



socket.io

Canvi de Next a Nuxt



Chat IA



GPT-3.5

Fantàstic per a tasques del dia a dia

ChatGPT 4 ▾

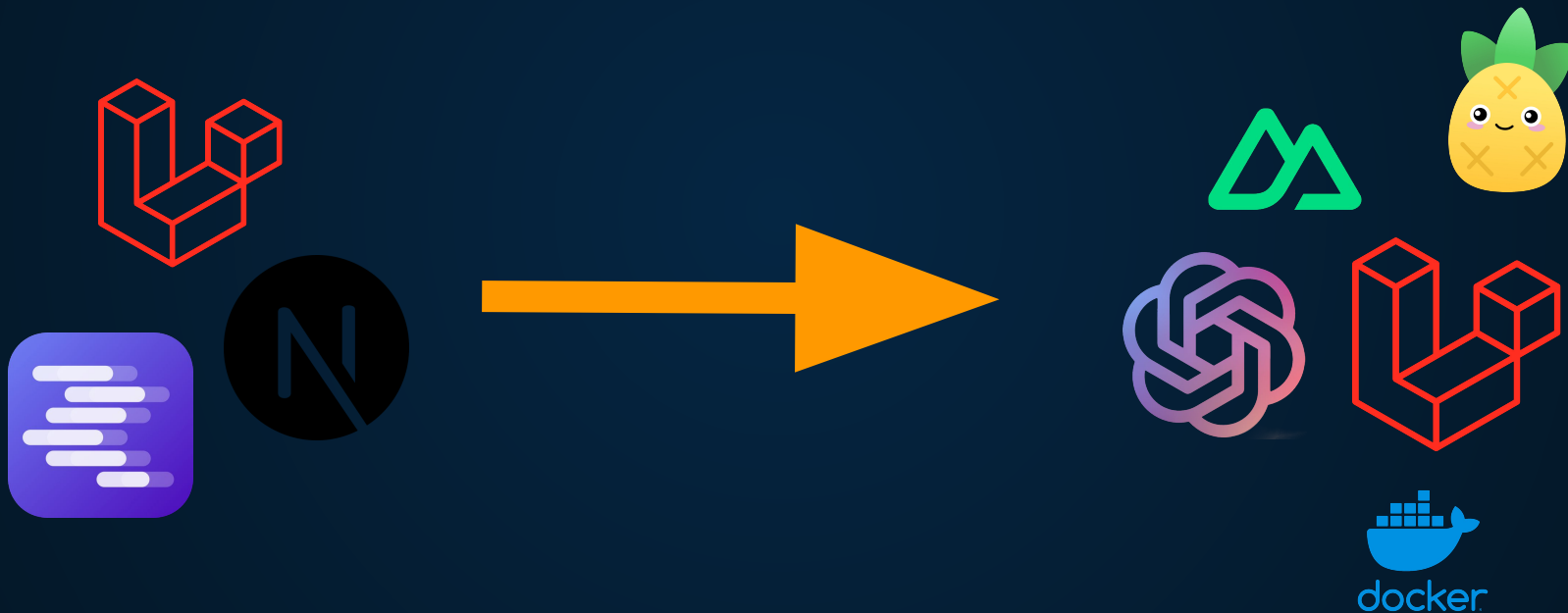
Problemes amb la gestió
d'usuaris



Laravel

Aspectes tècnics

Evolució



Aspectes tècnics

Funcionalitats

CommunicationManager

```
src > stores > communicationManager.js / ...
1 // const url = 'http://localhost:8000/api';
2 const url = 'http://fithub.daw.inspedralbes.cat/back/public/api';
3 const apiUrl = 'https://api.openai.com/v1/chat/completions';
4 const apiKey = 'sk-Kqz2dcsp7NB8gatVPV61T381bkFJ08Rmqj86cxT8hqQ6dFhX'; // Usa la variable de en
5
6
7 //ejemplo de petición fetch get
8
9 export function getDatosUsuario(idUsuario) {
10   return new Promise((resolve, reject) => {
11     fetch(`${url}/usuari/${idUsuario}`)
12       .then(response => {
13         if (response.ok) {
14           return response.json();
15         } else {
16           reject('Error al obtener los datos del usuario: ' + response.statusText);
17         }
18       })
19       .then(data => {
20         resolve(data);
21       })
22       .catch(error => {
23         reject('Error de red al obtener los datos del usuario: ' + error.message);
24       });
25   });
26 }
27
28 export async function getDatosUsuario2(idUsuario) {
29   try {
30     const response = await fetch(`${url}/usuari/${idUsuario}`);
31     if (!response.ok) {
32       throw new Error('Error al obtener los datos del usuario: ' + response.statusText);
33     }
34
35     const data = await response.json();
36     return data; // Devuelve los datos del usuario en JSON
37   } catch (error) {
38     throw new Error('Error de red al obtener los datos del usuario: ' + error.message);
39   }
40 }
41
42
```

```
export async function enviarMensajeOpenAIRutina(message, ejercicios, datosUsuario) {
  try {
    const payload = {
      model: 'gpt-3.5-turbo',
      messages: [
        {
          role: 'system',
          content: "Ets una persona que només parla en català i tens prohibit parlar d'alguna cosa que no tingui relació amb el fitness ja que ets un expert en fitness només tens permès parlar de fer rutines." +
            " Si et demanen alguna cosa que no sigui una rutina digues el següent: En aquest apartat només puc donar consells de nutrició i generar rutines." +
            " Pots donar consells i arguments però fetxo de forma resumida en unes 2 línies a menys que t'indiqui que vols més informació." +
            " Només pots respondre amb format JSON quan i seguint aquesta estructura: " +
            " { id_usuario: '', dias: [[dia: '1', ejercicios: [['nom_ejercicio': '', 'series': '', 'repeticiones': '', 'id_ejercicio': ''], ...]], ...]]" +
            " Segueix aquesta estructura de JSON pero posa per dia un mínim de 5 exercicis i un màxim de 7 exercicis." +
            " Fes un grup muscular per dia i no repetir exercicis en la mateixa rutina. A no ser que et digui el contrari o algo més concret." +
            " Si el missatge conte dades específiques o informació afegeix tota l'informació i crea la dieta; si no fes una rutina de 5 dies amb 5 músculs per dia i que sigui variada." +
            " Si en el missatge s'especifica quin tipus de rutina volen i els dies, fes-la com ho diuen. Si falta informació fes 5 dies amb 5 músculs per dia variats." +
            " Afegeix les dades del usuari per fer rutines més personalitzades i també afegeix el id del usuari per posarlo a id_usuario."
        },
        {
          role: 'system',
          content: JSON.stringify(ejercicios)
        },
        {
          role: 'system',
          content: JSON.stringify(datosUsuario)
        },
        {
          role: 'user',
          content: message
        }
      ]
    };
  } catch (error) {
    console.error('Error en enviar mensaje:', error);
  }
}

const response = await fetch(apiUrl, {
  method: 'POST',
  headers: {
    'Authorization': 'Bearer ' + apiKey,
    'Content-type': 'application/json',
  },
  body: JSON.stringify(payload),
});
```

Aspectes tècnics

Funcionalitats

Chat IA

```
const arbrePreguntes = {
  pregunta: "Quin tipus de dieta vols?",
  opcions: {
    Volum: {
      pregunta: "Quants àpats prefereixes fer al dia per a una dieta de volum?",
      opcions: {
        "4": "Dieta de volum de 4 àpats esmorzar, dinar, berenar, sopar",
        "5": {
          pregunta: "Vols incloure un segon esmorzar o un post-entrenament?",
          opcions: {
            "Segon esmorzar": "Dieta de volum de 5 àpats Esmorzar, Segon esmorzar, Dinar, Berenar, Sopar",
            "Post-entrenament": "Dieta de volum de 5 àpats Esmorzar, Dinar, Berenar, Post-entrenament, Sopar"
          }
        },
        "6": "Dieta de volum de 6 àpats Esmorzar, Segon esmorzar , Dinar, Berenar, Post-entrenament, Sopar"
      }
    },
    Definició: {
      pregunta: "Quants àpats prefereixes fer al dia per a una dieta de definició?",
      opcions: {
        "4": "Dieta de definició de 4 àpats Esmorzar, Dinar, Berenar, Sopar",
        "5": {
          pregunta: "Vols incloure un Segon esmorzar o un Post-entrenament?",
          opcions: {
            "Segon esmorzar": "Dieta de definició amb 5 àpats Esmorzar, Segon Esmorzar, Dinar, Berenar, Sopar",
            "Post-entrenament": "Dieta de definició amb 5 àpats: Esmorzar, Dinar, Post-entrenament, Berenar, Sopar."
          }
        },
        "6": "Dieta de definició amb 6 àpats: Esmorzar, Segon Esmorzar, Dinar, Post-entrenament, Berenar, Sopar"
      }
    },
    Equilibrada: {
      pregunta: "Quants àpats prefereixes al dia per a una dieta equilibrada?",
      opcions: {
        "4": "Dieta equilibrada amb 4 àpats",
        "5": {
          pregunta: "Vols incloure un segon esmorzar o un post-entrenament?",
          opcions: {
            "Segon esmorzar": "Dieta equilibrada amb 5 àpats: Esmorzar, Segon Esmorzar, Dinar, Berenar, Sopar",
            "Post-entrenament": "Dieta equilibrada amb 5 àpats: Esmorzar, Dinar, Post-entrenament, Berenar, Sopar"
          }
        },
        "6": "Dieta equilibrada amb 6 àpats"
      }
    }
  }
}
```

```
handleOptionSelect(optionKey) {
  let nextStep = this.currentOptions[optionKey];

  // Comprobar si hay un siguiente nivel de opciones
  if (nextStep && typeof nextStep === 'object' && nextStep.pregunta) {
    // Actualizar la pregunta y opciones actuales si hay más preguntas
    this.currentQuestion = nextStep.pregunta;
    this.currentOptions = nextStep.opciones || {};
    // Añadir la respuesta del asistente al chat
    this.chatMessages.push({
      role: 'assistant',
      content: this.currentQuestion
    });
  } else {
    // Manejar el final del árbol de preguntas
    this.currentQuestion = nextStep; // Esta sería la respuesta final
    this.currentOptions = {};
    // Establecer el mensaje final en 'message' para ser enviado
    this.message = nextStep;
    // Añadir al chat y preparar para enviar
    this.chatMessages.push({
      role: 'assistant',
      content: this.currentQuestion
    });
    // Opcional: Llamar a enviarMensaje directamente si se desea enviar inmediatamente
    this.enviarMensaje();
  }
},
```

```
<!-- Opciones de respuesta como botones -->
<div v-if="currentOptions && showOptions" class="botones-preseleccionados">
  <button v-for="(option, key) in currentOptions" :key="key" @click="handleOptionSelect(key)">
    {{ key }}
  </button>
</div>
```

Aspectes tècnics

Funcionalitats

Registre

```
<template>
  <div class="register-container">

    <div v-if="showStartButton">

      <div class="logo-container">
        
      </div>

      <div class="button-container">
        <button class="button button--secondary" @click="startRegistration">Començar Registre</button>
        <button class="button button--secondary" @click="goToIniciarSesion">Iniciar sessió</button>
      </div>
    </div>

    <div v-else>
      <div class="progress-bar" :style="{ width: progressPercentage + '%' }"></div>

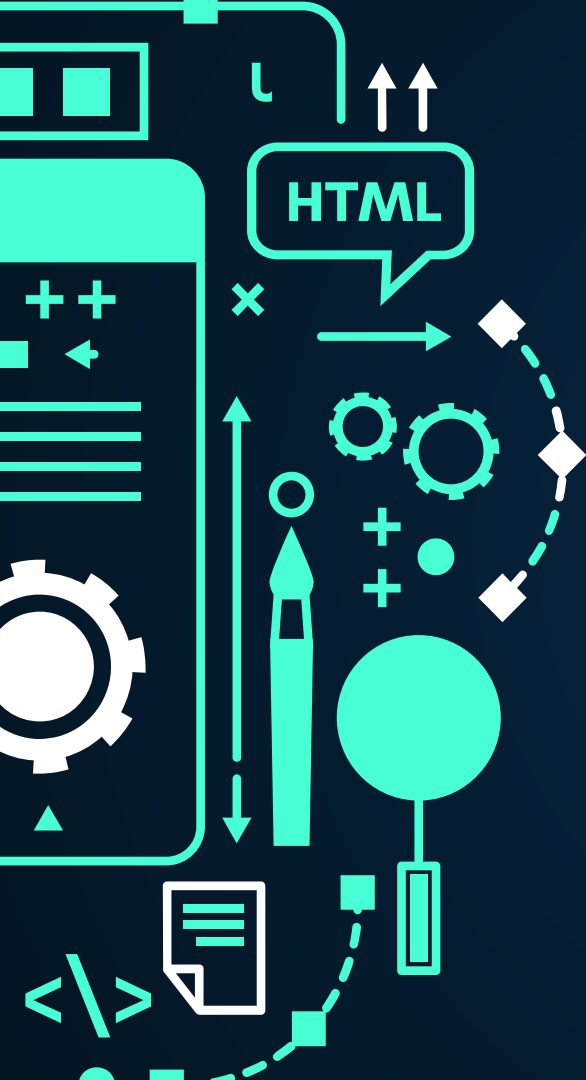
      <div v-if="currentQuestionIndex === registrationQuestions.length" class="loading">
        
      </div>
      <div v-if="currentQuestionIndex < registrationQuestions.length" class="question-container fade-in">
        <div v-if="currentQuestionIndex < registrationQuestions.length" class="question-counter">
          {{ currentQuestionIndex + 1 }} / {{ totalQuestions }}
        </div>
        <h2 class="registrationQuestion fade-in">{{ registrationQuestions[currentQuestionIndex].question }}</h2>

        <input v-if="registrationQuestions[currentQuestionIndex].inputType === 'textarea'"
          v-model="currentAnswer" placeholder="Escriu la teva resposta" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'email'"
          v-model="currentAnswer" type="email" placeholder="Correu electrònic" @input="validateEmailInput" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'password'"
          v-model="currentAnswer" type="password" placeholder="Contrasenya" @input="validatePassword" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'nom_usuari'"
          v-model="currentAnswer" type="text" placeholder="Nom d'usuari" maxlength="20" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'nom'"
          v-model="currentAnswer" type="text" placeholder="Nom" @input="validateNameInput" maxlength="25" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'cognoms'"
          v-model="currentAnswer" type="text" placeholder="Cognoms" @input="validateNameInput" maxlength="80" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'altura'"
          v-model="currentAnswer" type="text" placeholder="Altura (cm)" @input="validateAlturaInput"
            maxlength="4" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'pes'"
          v-model="currentAnswer" type="text" placeholder="Pes (kg)" @input="validatePesInput" maxlength="6" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'telefon'"
          v-model="currentAnswer" type="text" placeholder="Número de telèfon" @input="validateTelefonInput" />
        <input v-else-if="registrationQuestions[currentQuestionIndex].inputType === 'data_naixement'"
          v-model="currentAnswer" type="date" @input="validateDate" />
        <div v-if="registrationQuestions[currentQuestionIndex].inputType === 'genre'"
          <div class="gender-options">
            <div v-for="(option, index) in registrationQuestions[currentQuestionIndex].resposta"
              <key="index" class="gender-option">
              <input type="radio" :id="option" + index :value="option" v-model="currentAnswer">
              <label :for="option" + index @click="selectGenderOption(option)">{{ option }}</label>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</template>
```

```
<script>
import { useUsuariPerFilStore } from '@stores/index'

export default {
  data() {
    return {
      showStartButton: true,
      checkingEmail: false,
      errorMessage: "",
      showErrorMessage: false,
      currentQuestionIndex: 0,
      currentAnswer: "",
      selectedOptions: [],
      phoneNumberValid: false,
      passwordErrorMessage: "",
      dateErrorMessage: "",
      userData: {
        email: "",
        contrasenya: "",
        nom_usuari: "",
        nom: "",
        cognoms: "",
        data_naixement: "",
        genere: "",
        pes: "",
        altura: "",
        telefon: "",
        registre: false,
      },
    }
  },
}
```

```
registrationQuestions: [
  {
    question: "Quin es el teu correu electrònic?",
    inputType: 'email',
    required: true,
  },
  {
    question: "Quina es la teva contrasenya?",
    inputType: 'contrasenya',
    required: true,
  },
  {
    question: "Quin es el teu nom d'usuari?",
    inputType: 'nom_usuari',
    required: true,
  },
  {
    question: "Quin es el teu nom?",
    inputType: 'nom',
    required: true,
  },
  {
    question: "Quins son els teus cognoms?",
    inputType: 'cognoms',
    required: true,
  },
]
```



Gracies!

Alguna pregunta?

<http://fithub.daw.inspedralbes.cat/>