



CÓDIGO KUMARI

# Contenido

<a href="#"><u>APP</u></a>	1
<a href="#"><u>Auth</u></a>	1
<a href="#"><u>Login.txt</u></a>	1
<a href="#"><u>Boot</u></a>	9
<a href="#"><u>Auth.txt</u></a>	9
<a href="#"><u>Firebase.txt</u></a>	9
<a href="#"><u>Layouth</u></a>	10
<a href="#"><u>Mainlayouth.txt</u></a>	10
<a href="#"><u>User</u></a>	12
<a href="#"><u>Buttonstickers.txt</u></a>	12
<a href="#"><u>Buttontickets.txt</u></a>	20
<a href="#"><u>Dailyactivation.txt</u></a>	25
<a href="#"><u>Dashboard.txt</u></a>	44
<a href="#"><u>Forfathers.txt</u></a>	68
<a href="#"><u>Lesson.txt</u></a>	74
<a href="#"><u>Lessons.txt</u></a>	109
<a href="#"><u>Actions.txt</u></a>	128
<a href="#"><u>Mutations.txt</u></a>	133
<a href="#"><u>Routes.txt</u></a>	134
<a href="#"><u>state.txt</u></a>	135
<a href="#"><u>Web</u></a>	135
<a href="#"><u>Admin</u></a>	135
<a href="#"><u>Courses.txt</u></a>	135
<a href="#"><u>Instructors.txt</u></a>	164
<a href="#"><u>Lessons.txt</u></a>	181
<a href="#"><u>Orders.txt</u></a>	209
<a href="#"><u>Questions.txt</u></a>	253
<a href="#"><u>Users.txt</u></a>	262
<a href="#"><u>Auth</u></a>	293

<a href="#">Login.txt</a>	293
<a href="#">Profile.txt</a>	299
<a href="#">Boot</a>	335
<a href="#">Auth.txt</a>	335
<a href="#">Firebase.txt</a>	336
<a href="#">Instructors</a>	337
<a href="#">Clases.txt</a>	337
<a href="#">Courses.txt</a>	365
<a href="#">Progresscomponent.txt</a>	384
<a href="#">Layouts</a>	386
<a href="#">Mainlayout.txt</a>	386
<a href="#">Market</a>	396
<a href="#">Firstcollection.txt</a>	396
<a href="#">Market.txt</a>	398
<a href="#">Users</a>	427
<a href="#">Account.txt</a>	427
<a href="#">Activity.txt</a>	458
<a href="#">Questions.txt</a>	469
<a href="#">Receipts.txt</a>	476
<a href="#">Test.txt</a>	488
<a href="#">Actions.txt</a>	521
<a href="#">Dashboard.txt</a>	527
<a href="#">Index.txt</a>	531
<a href="#">Mutations.txt</a>	553
<a href="#">Routes.txt</a>	554
<a href="#">State.txt</a>	556

## APP

### Auth

Login.txt

<style scoped>

.card{

```
    width: 700px;
    border-radius: 20px;
}
```

```
.card .content{
    padding: 20px 40px;
}
```

```
.card .side{
    width: 200px;
    background-color: #F5F6F8;
}
```

```
.card .side img{
    width: 200px;
    object-fit: cover;
    object-position: center;
}
```

```
.card p{
    line-height: 12px;
}
```

```
.card p:nth-child(1){
    font-weight: bold;
    font-size: 22px;
}
```

```
.card p:nth-child(2){
```

```
color: #676879;  
font-size: 16px;  
}
```

```
.card input[type=text], input[type=password], input[type=email]{  
width: 100%;  
background-color: transparent;  
padding: 5px;  
border: 1px solid #E2E3E7;  
border-radius: 5px;  
}
```

```
.card button{  
width: 100%;  
background-color: #0085ff;  
color: white;  
margin-top: 10px;  
padding: 10px 10px;  
border: none;  
cursor: pointer;  
}
```

```
.card button:focus{  
outline: none;  
}
```

```
.img-container{  
display: flex;  
justify-content: center;
```

```
    align-items: center;
}
```

```
.btn-login{
    background-color: #A1C62E;
    color: white;
    border-radius: 20px;
    width: 120px;
    padding: 5px;
    display: inline-block;
}
```

```
.btn-google{
    background-color: white !important;
    color: black !important;
    border: 1px solid gray !important;
    border-radius: 20px;
}
```

```
.btn-apple{
    background-color: black !important;
    color: white !important;
    border: 1px solid gray !important;
    border-radius: 20px;
}
```

```
.background-all{
    background-image: url('https://firebasestorage.googleapis.com/v0/b/kumari-
d465a.appspot.com/o/bg%2FVerde.jpg?alt=media&token=392d70cf-73c9-4a77-aa8f-
6e01f043eb98');
}
```

```

        background-position: center;

        background-size: cover;
    }
</style>

<template>

    <q-page
        class="window-height window-width row justify-center items-center background-all"
    >

        <div class="column q-pa-lg">

            <div class="row">

                <div class="col-12">

                    <div class="card shadow-3" v-if="step == 1" style="background-color: white;">

                        <div class="row q-pa-md">

                            <div class="col-7 img-container">

                                

                            </div>

                            <div class="col-5 text-center" style="display: flex; flex-direction: column; justify-
content: center;">

                                <input v-model="user.email" type="email" placeholder="Correo">

                                <input class="q-mt-md" v-model="user.password" type="password"
placeholder="Contraseña">

                                <div style="width: 100%; display: flex; justify-content: flex-end;">

                                    <!-- <router-link style="font-size: 11px;" class="q-mt-md"
to="/register">¿Olvidaste tu contraseña?</router-link> -->

                                </div>

                                <div style="width: 100%; display: flex; justify-content: center; margin-top: 20px;"
class="cursor-pointer">

                                    <div class="btn-login" @click="onSubmit">Iniciar sesion</div>

                                </div>

```

```
<!-- <button class="btn-google">  
    Continuar con Google  
</button>  
  
<button class="btn-apple">  
    Continuar con Apple  
</button> -->  
</div>  
</div>  
</div>  
  
<div class="card" v-if="step == 2">  
    <div class="row">  
        <div class="col-12" style="text-align: center;">  
            <q-spinner-ball  
                color="primary"  
                size="4em"  
            />  
            <q-tooltip :offset="[0, 8]">Wait a moment!</q-tooltip>  
        </div>  
    </div>  
</div>  
</div>  
</div>  
</div>  
</q-page>  
</template>  
  
<script>
```



```
import { db } from 'boot/firebase'
```

```
import { mapState, mapActions } from 'vuex'
```

```
import { required, minLength, email } from 'vuelidate/lib/validators'
```

```
export default {
```

```
  name: 'Login',
```

```
  data(){
```

```
    return{
```

```
      isPwd: true,
```

```
      step: 1,
```

```
      user: {
```

```
        email: "",
```

```
        password: ""
```

```
      }
```

```
    }
```

```
  },
```

```
  validations: {
```

```
    user: {
```

```
      email: {
```

```
        email,
```

```
        required,
```

```
      },
```

```
      password: {
```

```
        required,
```

```
    },  
    },  
  },
```

```
mounted(){  
  window.screen.orientation.lock('landscape')  
},
```

```
methods: {  
  ...mapActions('auth', [ 'loginUser', 'closeSesion' ]),
```

```
  async onSubmit(){
```

```
    try {  
      this.$q.fullscreen.request()  
      .then(() => {  
        // success!  
      })  
      .catch(err => {  
        // oh, no!!!  
      })
```

```
      if (!this.$v.user.email.required || !this.$v.user.password.required) {  
        this.$q.notify({  
          message: 'The info is required',  
          color: 'negative'  
        })
```

```
      return
```

```

    }

    this.loginUser(this.user).then((response) => {
      console.log('binnieeeee: ', response);
    }).catch(error => {
      console.log(error);

      this.$q.notify({
        message: error.message,
        color: 'negative'
      })
    })

    } catch (error) {
      console.log(error);
    }
  },
}
}
</script>

```

## Boot

### Auth.txt

```

// import something here

import { auth } from 'boot/firebase'

// "async" is optional;

// more info on params: https://quasar.dev/quasar-cli/boot-files

export default async ({ store }) => {
  auth.onAuthStateChanged(user => {

```

```
if(user){  
  store.dispatch('auth/getUser', user.uid)  
  // localStorage.setItem('loggedIn', true)  
}else{  
  store.dispatch('auth/closeSession')  
}  
})  
}
```

### Firestore.txt

```
import firebase from 'firebase/app';
```

```
require('firebase/auth')  
require('firebase/firestore')  
require('firebase/storage')  
require('firebase/functions')
```

```
var firebaseConfig = {  
  apiKey: "AlzaSyClo3o_NygtY1YsY_86No_PJD8x1zVJZvQ",  
  authDomain: "kumari-d465a.firebaseio.com",  
  projectId: "kumari-d465a",  
  storageBucket: "kumari-d465a.appspot.com",  
  messagingSenderId: "569332193641",  
  appId: "1:569332193641:web:e35f82245845e5cb76c5f8",  
  measurementId: "G-ZXMNN30GEC"  
};
```

```
// Initialize Firebase
```

```
firebase.initializeApp(firebaseConfig);
```

```
// firebase.analytics();

const auth = firebase.auth()
const db = firebase.firestore()
const storage = firebase.storage()
const functions = firebase.functions()

export {
  firebase,
  auth,
  db,
  storage,
  functions
}
```

## Layouth

### Mainlayout.txt

```
<style scoped lang="scss">
```

```
</style>
```

```
<template>
```

```
  <q-layout view="IHh Lpr IFf">
```

```
    <q-page-container class="content" style="background-size: cover; background-repeat: no-repeat;" :style="{ 'background-image': 'url(/bg/' + backgroundImage + ')' }">
```

```
      <div>
```

```
        <router-view />
```

```
      </div>
```

```
    </q-page-container>
```

```
  </q-layout>
```

```
</template>
```

```
<script>
```

```
import { db } from 'boot/firebase'
```

```
import { mapState, mapActions } from 'vuex'
```

```
export default {
```

```
  name: 'MainLayout',
```

```
  components: {
```

```
  },
```

```
  computed: {
```

```
    ...mapState('auth', [ 'user' ]),
```

```
    backgroundImage(){
```

```
      if (this.user) {
```

```
        return this.user.background_image
```

```
      }else{
```

```
        return 'https://wallpapercave.com/wp/wp3118534.jpg'
```

```
      }
```

```
    }
```

```
  },
```

```
  data () {
```

```
    return {
```

```

    }
  },

  methods: {
    ...mapActions('auth', [ 'closeSesion' ]),

    logOut(){
      this.closeSesion()
    }
  }
}
</script>

```

## User

### Buttonstickers.txt

```

<style scoped lang="scss">
  .btn-star{
    width: 120px;

    background-color: #26B7CA;

    display: flex;

    justify-content: center;

    align-items: center;

    border-radius: 0 0 20px 20px;

    margin-top: -40px;

    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);

    cursor: pointer;

    img{
      margin-top: 40px;

      margin-bottom: 10px;
    }
  }

```

```
        width: 60px;
    }
}
```

```
.stickers-header{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
```

```
    img{
        width: 300px;
    }
```

```
    span{
        text-align: center;
    }
}
```

```
.sticker-container{
    padding: 8px;
```

```
    img{
        width: 100%;
    }
```

```
.img-filter{
    -webkit-filter: grayscale(100%); /* Safari 6.0 - 9.0 */
```



```
        filter: grayscale(100%);
    }
}
```

```
@media only screen and (max-device-width: 900px) and (orientation: landscape) {
```

```
    .btn-star{
        margin-top: -30px;
        width: 80px;

        img{
            width: 40px !important;
        }
    }
}
```

```
    .btn-ticket{
        margin-top: -30px;
        width: 80px;

        img{
            width: 40px !important;
        }
    }
}
```

```
    .tickets-header{
        span{
            font-size: 2em;
        }
    }
}
```

```
@media only screen
and (max-width : 1135px)
and (orientation : portrait) {

    .btn-star{

        margin-top: -30px;

        width: 30px;


        img{

            width: 20px !important;

        }

    }


    .stickers-header{

        img{

            width: 150px;

        }


        span{

            font-size: 13px;

            text-align: center;

        }

    }

}

</style>
```

```
<template>

<div>

    <div class="flex">
```

```

<div class="btn-star" @click="openModalStickets = true">
  
</div>
</div>

<q-dialog v-model="openModalStickets" full-width>
  <div>
    <q-card>
      <q-card-section>
        <div class="row">
          <div class="col-3 stickers-header">
          </div>
          <div class="col-6 stickers-header">
            
            <span>Para desbloquear mas stickers completa mas lecciones</span>
          </div>
          <div class="col-3 flex justify-end items-start">
            <span style="font-weight: bold; font-size: 25px; cursor: pointer;"
@click="openModalStickets = false">X</span>
          </div>
        </div>
      </q-card-section>
      <q-card-section>
        <div class="row">
          <div class="col-6 col-sm-2" v-for="item in stickers">
            <div class="sticker-container">
              
            </div>
          </div>
        </div>
      </q-card-section>
    </q-card>
  </div>
</q-dialog>

```

```

        </div>
      </q-card-section>
    </q-card>
  </div>
</q-dialog>

</div>
</template>

<script>
import { db, storage } from 'boot/firebase'
const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'

export default {
  name: 'ButtonStickets',

  data(){
    return{
      openModalStickets: false,
      openModalTickets: false,
      MyStickers: [],
      stickers: [],
      tickets: [],

      ticket: {},
    }
  },

```

```

computed: {
  ...mapState('auth', ['user', 'courses']),
},
watch: {
  openModalTickets(){
    if (this.openModalTickets) {
      this.ticket = this.tickets[Math.floor(Math.random() * this.tickets.length)]
    }
  }
},

```

```

mounted(){
  this.getAllLessons()
  this.getMyStickers()
},

```

```

methods: {
  async getAllLessons(){
    try {
      let response = await db.collection('clases')
        .get()
        .then(query => {
          query.forEach(doc => {
            let sticker = {}
            sticker.sticker = doc.data().sticker
            sticker.id = doc.id

            this.stickers.push(sticker)

```

```
    })  
  })
```

```
  } catch (error) {  
    console.log(error);  
  }  
},
```

```
async getMyStickers(){  
  try {  
    let response = await db.collection('stickers')  
      .where('user', '==', this.user.uid)  
      .get()  
      .then(query => {  
        query.forEach(doc => {  
          let sticker = doc.data()  
  
          this.MyStickers.push(sticker)  
        })  
      })  
  } catch (error) {  
    console.log(error);  
  }  
},
```

```
findSticker(id){  
  let response = this.MyStickers.find(doc => doc.clase == id)  
  if (response) {  
    return true  
  }  
}
```

```
        }else{
            return false
        }
    }
},
}
```

</script>

### Buttontickets.txt

<style scoped lang="scss">

```
.btn-ticket{
    width: 120px;
    background-color: #26B7CA;
    display: flex;
    justify-content: center;
    align-items: center;
    border-radius: 0 0 20px 20px;
    margin-top: -40px;
    margin-left: 10px;
    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);
    cursor: pointer;
```

```
    img{
        margin-top: 40px;
        margin-bottom: 10px;
        width: 60px;
    }
}
```

```
.tickets-header{
```

```
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
```

```
span{
    text-align: center;
    font-size: 3em;
}
}
```

@media only screen and (max-device-width: 900px) and (orientation: landscape) {

```
.btn-star{
    margin-top: -30px;
    width: 80px;

    img{
        width: 30px !important;
    }
}
```

```
.btn-ticket{
    margin-top: -30px;
    width: 80px;

    img{
        width: 40px !important;
    }
}
```



```
.tickets-header{  
  span{  
    font-size: 2em;  
  }  
}  
}
```

```
@media only screen  
and (max-width : 1135px)  
and (orientation : portrait) {  
  .btn-ticket{  
    margin-top: -30px;  
    width: 30px;  
  
    img{  
      width: 23px !important;  
    }  
  }  
}  
</style>
```

```
<template>  
<div>  
  <div class="flex">  
    <div class="btn-ticket" @click="openModalTickets = true">  
        
    </div>  
  </div>  
</div>
```

```

<q-dialog v-model="openModalTickets">
  <div>
    <q-card>
      <q-card-section>
        <div class="row">
          <div class="col-12 tickets-header">
            <span>{{ this.ticket.ticket }}</span>
          </div>
        </div>
      </q-card-section>
    </q-card>
  </div>
</q-dialog>
</div>
</template>

```

```

<script>
import { db, storage } from 'boot/firebase'
const ref = storage.ref()
import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'

```

```

export default {
  name: 'ButtonStickets',

  data(){
    return{
      openModalStickets: false,

```

```

    openModalTickets: false,

    MyStickers: [],

    stickers: [],

    tickets: [],

    ticket: {},
  }
},

computed: {
  ...mapState('auth', ['user', 'courses']),
},
watch: {
  openModalTickets(){
    if (this.openModalTickets) {
      this.ticket = this.tickets[Math.floor(Math.random() * this.tickets.length)]
    }
  }
},

mounted(){
  this.getTickets()
},

methods: {
  async getTickets(){
    try {
      let response = await db.collection('tickets')
        .get()
    }
  }
}

```

```
        .then(query => {  
            query.forEach(doc => {  
                let ticket = doc.data()  
  
                this.tickets.push(ticket)  
            })  
        })  
    } catch (error) {  
        console.log(error);  
    }  
},  
},  
}  
</script>
```

[Dailyactivation.txt](#)

<style scoped lang="scss">

```
.content{  
    position: relative;  
    min-width: 100vw;  
    min-height: 100vh;  
    display: flex;  
    flex-direction: column;  
    justify-content: space-around;  
    align-items: center;  
  
.main-container{  
    width: 100%;  
    display: flex;  
    justify-content: center;
```

```
align-items: center;
```

```
.video-container{  
    position: relative;  
    width: 60%;  
    padding: 30px;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    border-radius: 60px;  
    background-color: #E93081;  
    object-fit: cover;  
    object-position: center;
```

```
.video_layer{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    z-index: 1000;  
}
```

```
video{  
    width: 100%;  
    object-fit: cover;  
    object-position: center;  
}
```

```
}
```

```
}
```

```
.footer{
```

```
  width: 100%;
```

```
  padding: 10px;
```

```
  display: flex;
```

```
  justify-content: space-between;
```

```
  align-items: center;
```

```
  & > div{
```

```
    flex: 1;
```

```
}
```

```
.controls-container{
```

```
  width: 50%;
```

```
}
```

```
.controls{
```

```
  display: flex;
```

```
  justify-content: center;
```

```
  align-items: center;
```

```
.btn-control-img{
```

```
  width: 50px;
```

```
  border-radius: 50%;
```

```
  box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);
```

```
}
```

```
Button{
```

```
        background-color: transparent;

        border: none;
    }
}
```

```
.info{
    display: flex;
    justify-content: flex-end;
    align-items: center;
```

```
.btn-info-img{
    width: 50px;
    margin: 10px;
    border-radius: 50%;
    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);
}
```

```
Button{
    background-color: transparent;
    border: none;
}
}
}
}
```

```
header{
    width: 100%;
```

```
.box-1{
```

```
display: flex;
justify-content: center;
align-items: center;
```

```
img{
  width: 55%;
}
}
```

```
.box-2{
  display: flex;
  justify-content: space-around;
  align-items: flex-start;
```

```
.title{
  width: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
```

```
p{
  color: white;
  font-size: 3em;
}
}
}
```

```
.box-3{
  display: flex;
```



justify-content: center;

align-items: center;

padding-top: 15px;

text-align: center;

a{

img{

width: 50%;

border-radius: 50%;

box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);

}

}

}

}

@media only screen and (max-device-width: 900px) and (orientation: landscape) {

header{

.box-1{

img{

width: 75%;

}

}

.box-2{

.title{

```
p{  
  color: white;  
  font-size: 2em;  
}  
}  
}
```

```
.box-3{
```

```
  a{  
    img{  
      width: 30%;  
    }  
  }  
}  
}
```

```
.main-container{  
  width: 100%;
```

```
.video-container{  
  width: 60% !important;  
  height: 180px !important;  
  padding: 10px !important;  
  border-radius: 0 !important;
```

```
.video_layer{  
  height: 180px !important;  
}
```

```
        video{
            width: 100%;
            height: 160px !important;
        }
    }
}
```

```
}
```

```
@media only screen
and (max-width : 1135px)
and (orientation : portrait) {
```

```
    .mood{
        flex-direction: column;
```

```
        .mood-image{
            padding: 0px;
        }
```

```
        .status-container{
            img{
                width: 70px !important;
            }
        }
    }
```

```
    .box-1{
```

```
display: flex;
justify-content: center;
align-items: center;
```

```
img{
    width: 100% !important;
}
}
```

```
.box-2{
    .title{
        p{
            font-size: 2em !important;
        }
    }
}
```

```
.main-container{
    width: 100%;
```

```
.video-container{
    width: 100% !important;
    height: 180px !important;
    padding: 10px !important;
    border-radius: 0 !important;
```

```
.video_layer{
    height: 180px !important;
}
```



```

</div>

<div class="col-8">

  <div class="box-2">

    <div class="title">

      <p>Activa tu energía</p>

    </div>

  </div>

</div>

<div class="col-2">

  <div class="box-3">

    <router-link :to="`/dashboard`">

      

    </router-link>

  </div>

</div>

</div>

</header>

<div class="main-container" v-show="!loading">

  <div class="video-container">

    <div class="video_layer"></div>

    <video ref="video_player" controlsList="nodownload"
:src="daily_activation.url"></video>

  </div>

</div>

<div class="footer" v-show="!loading">

  <div></div>

  <div class="controls-container">

    <input style="border-radius: 0px; width: 100%;" v-model="seekRange" @input="getSeek"
class="" ref="seek" type="range" name="" id="">

    <div class="controls">

```

```

<div @click="playVideo" class="btn-play" style="margin: 5px;">
  <button class="cursor-pointer">
    
  </button>
</div>

<div @click="pauseVideo" class="btn-play" style="margin: 5px;">
  <button class="cursor-pointer">
    
  </button>
</div>

<div @click="fullScreen" class="btn-play" style="margin: 5px;">
  <button class="cursor-pointer">
    
  </button>
</div>
</div>
</div>
<div class="info">
  <a href="#" download target="_blank">
    
  </a>
  <button @click="finishLesson" class="cursor-pointer">
    
  </button>
</div>
</div>
</q-page>
</template>

```

```
<script>

import { db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'

import moment from 'moment-timezone'

import axios from 'axios'

export default {
  data(){
    return{
      isPlaying: false,
      isMuted: false,
      volume: 0.5,
      seekDuration: 0,
      videoDuration: 0,
      videoTimeElapsed: 0,
      videoPlayer: null,
      loading: true,
      seekRange: 0,
      videoFinish: false,
    }
  },

  computed: {
    ...mapState('auth', ['user', 'courses', 'daily_activation'])
  },

  mounted(){
    this.videoPlayer = this.$refs.video_player
```



```

this.videoPlayer.addEventListener('loadedmetadata', () => {
    console.log(this.videoPlayer.duration);

    if (this.videoPlayer.duration == Infinity) {
        console.log('binnieeeeeeeeeeeee');
        this.videoPlayer.currentTime = 10000;
        this.videoPlayer.ontimeupdate = () => {
            this.ontimeupdate = () => {
                return;
            }

            this.initializeVideo()
            console.log('daleeee');
            return;
        }
    }else{
        this.initializeVideo()
    }
})

this.videoPlayer.addEventListener('timeupdate', this.updateTimeElapsed);
this.videoPlayer.addEventListener('timeupdate', this.updateProgress);
},

watch: {
    videoDuration(){
        console.log('Cambioooooo: ', this.videoDuration);

        this.videoPlayer.currentTime = 0
    },

```

```
videoTimeElapsed(){  
  if (this.videoTimeElapsed == this.videoDuration) {  
    this.videoFinish = true  
  }else{  
    this.videoFinish = false  
  }  
},
```

```
videoFinish(){  
  if (this.videoFinish) {  
    this.pauseVideo()  
    this.videoPlayer.currentTime = 0  
  
    this.$q.dialog({  
      title: 'Finalizado',  
      message: 'Termino el video.'  
    }).onOk(() => {  
  
    })  
  }  
}  
},
```

```
methods: {  
  finishLesson(){  
    if (this.lesson_homework != null) {  
      this.$q.dialog({  
        title: 'Finalizado',  
        message: 'Ya terminaste esta leccion.'      })  
    }  
  }  
}
```

```

        }).onOk(() => {

        })

    return
}

this.$q.dialog({
    title: 'Finalizar activacion',
    message: '¿Quieres finalizar tu activacion?',
    cancel: true,
    persistent: true
}).onOk(async () => {
    try {
        this.$router.push('/dashboard')
    } catch (error) {
        console.log(error);
    }
});

},

initializeVideo() {
    let videoDuration = Math.round(this.videoPlayer.duration)
    let time = this.formatTime(videoDuration)
    this.videoDuration = `${time.minutes}:${time.seconds}`
    this.$refs.seek.setAttribute('max', this.videoPlayer.duration);
    // this.$refs.progress_bar.setAttribute('max', this.videoPlayer.duration);
    this.loading = false

```

```

},
updateTimeElapsed() {
    let time = this.formatTime(Math.round(this.videoPlayer.currentTime));
    this.videoTimeElapsed = `${time.minutes}:${time.seconds}`;
},
updateProgress() {
    this.$refs.seek.value = Math.floor(this.videoPlayer.currentTime);
    // if (this.videoTimeElapsed == this.videoDuration) {
    //     alert('termino el video')
    // }
    // this.$refs.progress_bar.value = Math.floor(this.videoPlayer.currentTime);
},
undoSkip(){
    this.videoPlayer.currentTime = this.videoPlayer.currentTime - 10
},
getSeek($event){
    this.videoPlayer.currentTime = $event.target.value
    this.$refs.seek.value = $event.target.value
    // this.$refs.progress_bar.value = $event.target.value
},
skip(){
    this.videoPlayer.currentTime = this.videoPlayer.currentTime + 10
},
playVideo(){

    this.videoPlayer.play()

    this.isPlaying = true
},
clickVideo(){

```

```
    if (this.isPlaying == false) {  
        this.playVideo()  
    }else{  
        this.pauseVideo()  
    }  
},  
pauseVideo(){  
    this.videoPlayer.pause()  
    this.isPlaying = false  
},  
mutedVideo(){  
    this.videoPlayer.muted = true  
    this.isMuted = true  
},  
unmutedVideo(){  
    this.videoPlayer.muted = false  
    this.isMuted = false  
},  
fullScreen(){  
    if (document.fullscreenElement) {  
        document.exitFullscreen();  
    } else if (document.webkitFullscreenElement) {  
        // Need this to support Safari  
        document.webkitExitFullscreen();  
    } else if (this.videoPlayer.webkitRequestFullscreen) {  
        // Need this to support Safari  
        this.videoPlayer.webkitRequestFullscreen();  
    } else {  
        this.videoPlayer.webkitEnterFullscreen();  
    }  
}
```

```
    }  
  },  
  changeVol(){  
    this.videoPlayer.volume = this.volume  
  },  
  formatTime(timeInSeconds) {  
    const result = new Date(timeInSeconds * 1000).toISOString().substr(11, 8);  
    return {  
      minutes: result.substr(3, 2),  
      seconds: result.substr(6, 2),  
    };  
  },  
}  
}  
</script>
```

<style>

</style>

Dashboard.txt

<style scoped lang="scss">

```
.content{  
  position: relative;  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
  align-items: center;  
  background-color: rgba(0, 0, 0, 0.5);
```

```
.header{  
  width: 100%;  
  
  .header-menu{  
    width: 100%;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
  
    .btn-logout{  
      width: 80px;  
      height: 80px;  
      display: flex;  
      justify-content: center;  
      align-items: center;  
      border-radius: 50%;  
      margin-right: 10px;  
  
      img{  
        width: 50px;  
      }  
    }  
  }  
}
```

```
.main-container{
```

width: 100%;

overflow-x: hidden;

.course-container{

display: flex;

justify-content: center;

align-items: center;

padding: 10px;

cursor: pointer;

.course-body{

aspect-ratio: 16/9;

display: flex;

justify-content: center;

align-items: center;

position: relative;

border-radius: 10px;

box-shadow: 5px 5px 10px rgba(46, 46, 46, 0.377);

img{

width: 80%;

}

.layer{

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;



```
        border-radius: 10px;

        background-color: rgba(0, 0, 0, 0.1);

        // background-color: red;
    }
}
```

```
}
```

```
.name-title{

    text-align: center;

    font-size: 22px;

    font-weight: bold;

    color: white;

}

}
```

```
.footer{

    visibility: hidden;

}
```

```
.check-status{

    width: 100%;

    height: 100%;

    background-color: rgba(0, 0, 0, 0.9);

    position: absolute;

    top: 0;

    left: 0;

    display: flex;
```

```
justify-content: center;

align-items: center;

flex-direction: column;

color: white;


.mood{

  display: flex;

  justify-content: center;

  align-items: center;


  .mood-image{

    padding: 5px;


    .status-container{

      display: flex;

      justify-content: center;

      align-items: center;

      flex-direction: column;


      img{

        width: 150px;

      }

    }

  }

}
```

```
}
```

```
header{
```

```
  width: 100%;
```

```
  .box-1{
```

```
    display: flex;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
    img{
```

```
      width: 55%;
```

```
    }
```

```
  }
```

```
  .box-2{
```

```
    display: flex;
```

```
    justify-content: space-around;
```

```
    align-items: flex-start;
```

```
  .for-dad{
```

```
    a{
```

```
      background-color: #F29305;
```

```
      color: white;
```

```
      border: none;
```

```
      border-radius: 0 0 10px 10px;
```

```
      text-decoration: none;
```

```
      font-size: 22px;
```

```
      padding: 5px 50px;
```

```
    }  
  }  
}
```

```
.box-3{  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  padding-top: 15px;
```

```
  img{  
    width: 25%;  
    border-radius: 50%;  
    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);  
  }  
}  
}
```

@media only screen and (max-device-width: 900px) and (orientation: landscape) {

```
  header{  
    .box-1{  
      img{  
        width: 75%;  
      }  
    }  
  }
```

```
  .box-2{  
  
    .for-dad{
```

```
    a{  
        font-size: 18px !important;  
        padding: 5px 20px !important;  
    }  
}  
}
```

```
.box-3{  
  
    img{  
        width: 30%;  
    }  
}  
}
```

```
.main-container{  
    width: 100%;  
  
}
```

```
.status-container{  
    img{  
        width: 90px !important;  
    }  
}
```

```
.swiper-container{
```

```
margin-top: 5% !important;
```

```
.name-title{  
    font-size: 14px !important;  
}  
}
```

```
.modal-tutorial{  
    width: 580px !important;
```

```
img{  
    width: 400px !important;  
}
```

```
.layer{  
    width: 550px !important;  
}
```

```
.pointer{  
    width: 5px !important;  
    height: 5px !important;  
    border-radius: 50%;  
    border: 1px solid #E72B81;  
    margin: 5px !important;
```

```
&.active{  
    background-color: #E72B81;  
}  
}
```

```
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
    margin-top: 10px;  
    font-size: 12px !important;  
}
```

```
}
```

```
.courses-web{  
    display: block;  
}
```

```
.courses-movil{  
    display: none;  
}
```

@media only screen

```
and (max-width : 1135px)  
and (orientation : portrait) {  
    .courses-web{  
        display: none;  
    }  
}
```

```
.courses-movil{  
  display: block;  
}
```

```
.box-1{  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
  img{  
    width: 100% !important;  
  }  
}
```

```
.box-3{  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  padding-top: 15px;
```

```
  img{  
    width: 50% !important;  
    border-radius: 50%;  
    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);  
  }  
}  
}
```



```
.tuto-container{  
  position: relative;  
  
  .layer{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background-color: rgba(0, 0, 0, 0.0);  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    color: black;  
    font-size: 50px;  
  }  
  
}
```

```
.btn-cancel{  
  color: white;  
  padding: 5px 20px;  
  border: none;  
  background-color: #E72B81;  
  border-radius: 10px;  
  cursor: pointer;
```

```
margin-top: 10px;
}
```

```
.modal-tutorial{
max-width: 800px !important;
```

```
img{
width: 800px;
}
```

```
.pointer{
width: 10px;
height: 10px;
border-radius: 50%;
border: 1px solid #E72B81;
margin: 10px;
```

```
&.active{
background-color: #E72B81;
}
```

```
}
```

```
}
```

```
</style>
```

```
<template>
```

```
<q-page v-if="user != null" class="content">
```

```
<header>
```

```
<div class="row">
```

```

<div class="col-2">
  <div class="box-1">
    
  </div>
</div>
<div class="col-8">
  <div class="box-2">
    <ButtonTickets />
    <ForFathers />
    <ButtonStickers />
  </div>
</div>
<div class="col-2">
  <div class="box-3">
    
  </div>
</div>
</div>
</header>

```

```

<!-- Courses web -->

```

```

<div class="main-container courses-web" v-if="courses.length != 0">
  <swiper ref="mySwiper" :options="swiperOptions">
    <swiper-slide>
      <div class="course-container" @click="goToDailyActivation">
        <div class="course-body" :style="{ backgroundColor: 'purple' }">
          
          <div class="layer"></div>

```

</div>

</div>

<p class="name-title">Activa tu energía</p>

</swiper-slide>

<swiper-slide v-for="(course, index) in courses" :key="index" v-if="course.id !=  
'aBupoFZjqjsdFXTDjvsH'">

<div class="course-container" @click="goToLessons(course, index)">

<div class="course-body" :style="{ backgroundColor: course.color }">



<div class="layer"></div>

</div>

</div>

<p class="name-title">{{ course.name }}</p>

</swiper-slide>

</swiper>

</div>

<!-- Courses movil -->

<div class="main-container courses-movil" v-if="courses.length != 0">

<div class="flex flex-col items-center justify-center">

<div>

<div class="course-container" @click="goToDailyActivation">

<div class="course-body" :style="{ backgroundColor: 'purple' }">



<div class="layer"></div>

</div>

</div>

<p class="name-title">Activa tu energía</p>

</div>

<div v-for="(course, index) in courses" :key="index" v-if="course.id !=  
'aBupoFZjqjsdFXTDjvsH'">

<div class="course-container" @click="goToLessons(course, index)">

<div class="course-body" :style="{ backgroundColor: course.color }">



<div class="layer"></div>

</div>

</div>

<p class="name-title">{{ course.name }}</p>

</div>

</div>

</div>

<div class="footer">

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sunt repudiandae reprehenderit voluptatibus suscipit dolor minus corporis, officia culpa, nesciunt error ratione quos neque unde quod quas deleniti consequatur eaque? Necessitatibus.</p>

</div>

<q-dialog v-model="user.tutorial" persistent>

<q-card class="modal-tutorial">

<q-card-section class="row">

<div class="tuto-container" style="width: 100%;">

<div class="flex justify-center items-center">

```

        
    </div>

    <div class="layer">

        <q-icon class="cursor-pointer" name="chevron_left" @click="indexMinus" />

        <q-icon class="cursor-pointer" name="chevron_right" @click="indexPlus" />

    </div>

</div>

<div class="flex justify-center items-center" style="width: 100%; margin-top: 10px;">

    <span class="pointer" :class="[indexImage == 0 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 1 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 2 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 3 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 4 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 5 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 6 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 7 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 8 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 9 ? 'active' : '']"></span>

    <span class="pointer" :class="[indexImage == 10 ? 'active' : '']"></span>

</div>

<div class="flex justify-center items-center" style="width: 100%; flex-direction:
column;">

    <button @click="updateTutorial" class="btn-cancel">No volver a mostrar</button>

    <q-btn flat label="Cerrar" color="primary" style="margin-top: 5px;" v-close-popup />

</div>

</q-card-section>

</q-card>

</q-dialog>

```

```
</q-page>
</template>
```

```
<script>
import { db, storage } from 'boot/firebase'
const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'

import VueSlickCarousel from 'vue-slick-carousel'
import 'vue-slick-carousel/dist/vue-slick-carousel.css'
// optional style for arrows & dots
import 'vue-slick-carousel/dist/vue-slick-carousel-theme.css'
import ButtonStickers from './components/ButtonStickers'
import ButtonTickets from './components/ButtonTickets'
import ForFathers from './components/ForFathers'
import { Swiper, SwiperSlide, directive } from 'vue-awesome-swiper'
// import style (>= Swiper 6.x)
import 'swiper/swiper-bundle.css'

export default {
  name: 'dashboard',

  components: {
    VueSlickCarousel,
    ButtonStickers,
    ButtonTickets,
    ForFathers,
```

```
    Swiper,  
    SwiperSlide  
  },
```

```
  directives: {  
    swiper: directive  
  },
```

```
  data(){  
    return{  
      indexImage: 0,  
      numberImage: [  
        '/tutorial/0.png',  
        '/tutorial/1.png',  
        '/tutorial/2.png',  
        '/tutorial/3.png',  
        '/tutorial/4.png',  
        '/tutorial/5.png',  
        '/tutorial/6.png',  
        '/tutorial/8.png',  
        '/tutorial/9.png',  
        '/tutorial/10.png',  
        '/tutorial/11.png',  
      ],  
      swiperOptions: {  
        slidesPerView: 4,  
        spaceBetween: 10  
      },  
      settings: {
```



```
"arrows": false,  
"dots": false,  
"focusOnSelect": false,  
"infinite": false,  
"speed": 500,  
"slidesToShow": 3,  
"touchThreshold": 5,  
"centerMode": false,  
"centerPadding": "20px",  
},
```

```
daily_background: "",
```

```
status: [  
  {  
    ico: 'Enojado.png',  
    title: 'Enojado'  
  },  
  {  
    ico: 'Triste.png',  
    title: 'Triste'  
  },  
  {  
    ico: 'Cansado.png',  
    title: 'Cansado'  
  },  
  {  
    ico: 'Feliz.png',  
    title: 'Feliz'  }  
]
```

```
    },
    {
      ico: 'Emocionado.png',
      title: 'Emocionado'
    },
  ],

  handleClick: true,
}
},

computed: {
  ...mapState('auth', ['user', 'courses', 'daily_activation']),

  backgroundImage(){
    if (this.user) {
      return this.user.background_image
    }else{
      return 'https://wallpapercave.com/wp/wp3118534.jpg'
    }
  },

  swiper() {
    return this.$refs.mySwiper.$swiper
  }
},

mounted(){
```

```

        window.screen.orientation.lock('landscape')
    },

    created() {
        this.getDailyBG()
    },

    methods: {
        ...mapActions('auth', ['closeSesion']),

        async updateTutorial(){
            try {
                this.user.tutorial = false

                let response = await db.collection('users').doc(this.user.uid).update({ tutorial: false })

                this.$store.commit('auth/updateTutorialUser', false)
            } catch (error) {
                console.log(error);
            }
        },

        indexMinus(){
            if (this.indexImage > 0) {
                this.indexImage--
            }
        },

        indexPlus(){
            if (this.indexImage <= 9) {
                this.indexImage++
            }
        }
    }
}

```

```

    }
  },

  onSwiper(swiper){
    console.log(swiper);
  },

  onSlideChange(){
    console.log('slide change');
  },

  async getDailyBG(){
    try {
      let response = await db.collection('daily_background').doc('current').get()

      this.daily_background = response.data()
    } catch (error) {
      console.log(error);
    }
  },

  goToDailyActivation(){
    this.$router.push('/daily-activation')
  },

  handleUp(){
    this.handleClick = false
    console.log(this.handleClick);
  },

```

```
handleDown(){  
  this.handleClick = true  
  console.log(this.handleClick);  
},
```

```
skipStatus(){  
  try {  
    this.$q.dialog({  
      title: 'Estado de animo',  
      message: 'Estado de animo omitido.'  
    }).onOk(() => {  
  
    })  
    this.$store.commit('auth/updateCheckStatus', true)  
  
    this.$router.push('/daily-activation')  
  } catch (error) {  
    console.log(error);  
  }  
},
```

```
goToLessons(item){  
  this.$router.push(`/lessons/${item.id}`)  
},
```

```
async checkStatus(index){  
  try {  
    let mood = {
```

```

        user: this.user.uid,
        mood: index,
        created_at: new Date(),
        created_at_unix: moment().unix(),
    }

    let response = await db.collection('mood').add(mood)

    let updateStatus = await db.collection('users').doc(this.user.uid).update({ check_status:
false })

    this.$q.dialog({
        title: 'Estado de animo',
        message: 'Estado de animo registrado.'
    }).onOk(() => {

    })

    this.$store.commit('auth/updateCheckStatus', false)
    this.$router.push('/daily-activation')
} catch (error) {
    console.log(error);
}
},

logout(){
    this.closeSesion()
}
},
}
</script>

```

## Forfathers.txt

```
<style lang="scss" scoped>
```

```
.for-dad{  
  button{  
    background-color: #F29305;  
    color: white;  
    border: none;  
    border-radius: 0 0 10px 10px;  
    text-decoration: none;  
    font-size: 22px;  
    padding: 5px 50px;  
  }  
  
}
```

```
.answers{  
  display: flex;  
  justify-content: space-around;  
  align-items: center;
```

```
.option{  
  width: 45px;  
  height: 45px;  
  border-radius: 50%;  
  border: 1px solid black;  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
color: black;

font-size: 15px;

cursor: pointer;

box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);


&.active{

    background-color: #F29305;

    color: white;

}

}

}
```

```
.btn-login{

    background-color: #A1C62E;

    color: white;

    border-radius: 20px;

    width: 120px;

    padding: 5px;

    display: inline-block;

    border: none !important;

}
```

```
@media only screen

and (max-width : 1135px)

and (orientation : portrait) {

    .for-dad{

        button{

            margin-left: 15%;

            width: 80%;

        }

    }

}
```



```
background-color: #F29305;

color: white;

border: none;

border-radius: 0 0 10px 10px;

text-decoration: none;

font-size: 12px;

padding: 5px 10px;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

}

}
```

</style>

<template>

<div class="for-dad">

<button @click="goToDashboard">



Para Papás

</button>

<q-dialog v-model="showModal" persistent>

<q-card>

<q-card-section>

```

    <div class="text-h6 text-center" style="font-size: 25px;">Pregunta a tus padres</div>
  </q-card-section>

  <q-card-section class="q-pt-none">

    <p class="text-center" style="font-size: 22px;">¿Cual es el resultado de esta
operación?</p>

    <p class="text-center" style="font-size: 19px;">{{ number1 }} * {{ number2 }} = </p>

    <div class="answers">

      <div class="option" :class="[index_selected == index ? 'active' : '']" v-for="(option,
index) in options" :key="index" @click="selectOption(option, index)">

        {{ option }}

      </div>

    </div>

  </q-card-section>

  <q-card-actions align="right">

    <button class="btn-login" style="background-color: #0085ff;" @click="showModal =
false">Regresar</button>

    <button class="btn-login" style="margin-left: 5px;"
@click="confirmAnswer">Confirmar</button>

  </q-card-actions>

</q-card>

</q-dialog>

</div>

</template>

<script>
export default {
  name: 'ForFathers',

```

```
data(){  
  return{  
    showModal: false,  
    number1: 0,  
    number2: 0,  
    answer: 0,  
  
    options: [],  
    index_selected: null,  
    option_selected: null,  
  }  
},
```

```
methods: {  
  goToDashboard(){  
    this.options = []  
    this.showModal = true  
  
    this.number1 = Math.round(Math.random() * 100)  
    this.number2 = Math.round(Math.random() * 100)  
  
    this.answer = this.number1 * this.number2  
    this.options.push(this.answer)  
  
    do {  
      let random = Math.round(Math.random() * 500)  
      this.options.push(random)  
    } while (this.options.length <= 3);
```

```

        this.options = this.options.sort(()=> Math.random() - 0.5);
    },

    selectOption(option, index){
        this.index_selected = index
        this.option_selected = option
    },

    confirmAnswer(){
        if (this.option_selected == this.answer) {
            window.open('https://dashboard.kumari.mx', '_blank');
            this.showModal = false
            this.index_selected = null
            this.option_selected = null
        }else{
            this.showModal = false
            this.index_selected = null
            this.option_selected = null
        }
    }
}
}
</script>

```

### Lesson.txt

```

<style scoped lang="scss">
    .content{
        position: relative;
    }

```

```
min-width: 100vw;  
min-height: 100vh;  
background-color: rgba(0, 0, 0, 0.5);
```

```
.main-body{  
  position: relative;  
  min-width: 100vw;  
  min-height: 100vh;  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
  align-items: center;
```

```
.header{  
  width: 100%;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 10px;
```

```
.title{  
  width: 100%;  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
p{  
  color: white;  
  font-size: 3em;
```

```
    }  
  }  
  
  .logo{  
    width: 100%;  
    display: flex;  
    justify-content: flex-start;  
    align-items: center;  
  }  
  
  .logout{  
    width: 100%;  
    display: flex;  
    justify-content: flex-end;  
    align-items: center;  
  
    img{  
      width: 50px;  
    }  
  }  
}
```

```
.main-container{  
  width: 100%;  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
.video-container{  
    position: relative;  
    width: 100%;  
    padding: 30px;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    border-radius: 60px;  
    background-color: #E93081;  
    object-fit: cover;  
    object-position: center;
```

```
.video_layer{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    z-index: 1000;  
}
```

```
video{  
    width: 100%;  
    object-fit: cover;  
    object-position: center;  
}  
}
```

```
.evidence-container{
```

```
width: 100%;  
height: 100%;  
display: flex;  
justify-content: center;  
align-items: center;
```

```
.btn-material-img{  
  width: 80px;  
  margin: 10px;  
  border-radius: 50%;  
  box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);  
}  
}  
}
```

```
.footer{  
  width: 100%;  
  padding: 10px;
```

```
& > div{  
  flex: 1;  
}
```

```
.controls-container{  
  width: 100%;  
}
```

```
.controls{  
  display: flex;
```



```
justify-content: center;
```

```
align-items: center;
```

```
.btn-control-img{
```

```
    width: 50px;
```

```
    border-radius: 50%;
```

```
    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);
```

```
}
```

```
Button{
```

```
    background-color: transparent;
```

```
    border: none;
```

```
}
```

```
}
```

```
.info{
```

```
    display: flex;
```

```
    justify-content: flex-end;
```

```
    align-items: center;
```

```
.btn-info-img{
```

```
    width: 50px;
```

```
    margin: 10px;
```

```
    border-radius: 50%;
```

```
    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);
```

```
}
```

```
Button{
```

```
    background-color: transparent;
```

```
        border: none;
    }
}
}
```

  

```
}
```

```
.main-loading{
    min-width: 100vw;
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
.check-status{
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.9);
    position: absolute;
    top: 0;
    left: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    color: white;
    z-index: 100000;
```

```
.mood{
  display: flex;
  justify-content: center;
  align-items: center;

  .mood-image{
    padding: 5px;
  }

  .status-container{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;

    img{
      width: 150px;
    }

  }
}

header{
  width: 100%;
```

```
.box-1{  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
  img{  
    width: 55%;  
  }  
}
```

```
.box-2{  
  display: flex;  
  justify-content: space-around;  
  align-items: flex-start;
```

```
.title{  
  width: 100%;  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
  p{  
    color: white;  
    font-size: 3em;  
  }  
}  
}
```

```
.box-3{
```

```
display: flex;

justify-content: center;

align-items: center;

padding-top: 15px;

text-align: center;


a{

  img{

    width: 50%;

    border-radius: 50%;

    box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);

  }

}

}

}

}

@media only screen and (max-device-width: 900px) and (orientation: landscape) {

  header{

    .box-1{

      img{

        width: 75%;

      }

    }

    .box-2{

      .title{
```

```
p{
  color: white;
  margin-top: 10px;
  font-size: 1.5em !important;
}
}
```

```
.box-3{

  a{
    img{
      width: 30%;
    }
  }
}
```

```
.main-container{
  width: 100%;
```

```
.video-container{
  width: 100% !important;
  height: 180px !important;
  padding: 10px !important;
  border-radius: 0 !important;
```

```
.video_layer{
```

```
        height: 180px !important;
    }
}
```

```
video{
    width: 100%;
    height: 160px !important;
}
}
```

```
.evidence-container{
    .btn-material-img{
        width: 50px !important;
        margin: 10px;
    }
}
}
```

```
.footer{
    padding: 10px;
```

```
.controls{

    .btn-control-img{
        width: 40px !important;
    }
}
```

```
.info{
    margin-top: 15px !important;
```

```
.btn-info-img{  
  
    width: 40px !important;  
}  
}  
}
```

```
.status-container{  
    img{  
        width: 90px !important;  
    }  
}  
}
```

```
@media only screen  
and (max-width : 1135px)  
and (orientation : portrait) {  
    .mood{  
        flex-direction: column;  
  
        .mood-image{  
            padding: 0px;  
        }  
    }
```

```
.status-container{  
    img{  
        width: 70px !important;  
    }  
}
```



```
}  
}
```

```
.box-1{  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  
  img{  
    width: 100% !important;  
  }  
}
```

```
.box-2{  
  .title{  
    p{  
      font-size: 1.5em !important;  
    }  
  }  
}
```

```
.main-container{  
  width: 100%;  
  
  .video-container{  
    width: 100% !important;  
    height: 180px !important;  
    padding: 10px !important;  
    border-radius: 0 !important;
```

```
.video_layer{  
    height: 180px !important;  
}
```

```
video{  
    width: 100%;  
    height: 160px !important;  
}  
}
```

```
.evidence-container{  
    .btn-material-img{  
        width: 50px !important;  
        margin: 10px;  
    }  
}  
}
```

```
.info{  
    width: 100% !important;  
    margin-top: 15px !important;  
    text-align: center;  
    justify-content: center !important;
```

```
.btn-info-img{
```

```
}
```

```
    }  
  }  
</style>
```

```
<template>
```

```
  <q-page class="content">
```

```
    <div v-show="!loading" class="main-body">
```

```
      <header>
```

```
        <div class="row">
```

```
          <div class="col-2">
```

```
            <div class="box-1">
```

```
              
```

```
            </div>
```

```
          </div>
```

```
        <div class="col-8">
```

```
          <div class="box-2">
```

```
            <div class="title">
```

```
              <p>{{ lesson.name }}</p>
```

```
            </div>
```

```
          </div>
```

```
        </div>
```

```
      <div class="col-2">
```

```
        <div class="box-3">
```

```
          <router-link :to="`/lessons/${lesson.course}`">
```

```
            
```

```
          </router-link>
```

```
        </div>
```

```
      </div>
```

```
    </div>
```

</header>

<div class="main-container">

<div class="row" style="width: 100%; padding: 0 5%;">

<div class="col-12 col-sm-6">

<div class="video-container">

<div class="video\_layer"></div>

<video width="100%" ref="video\_player" controlsList="nodownload"  
:src="lesson.url"></video>

</div>

</div>

<div class="col-12 col-sm-6">

<div class="evidence-container">

<a :href="lesson.material" download target="\_blank">



</a>

<div class="input-file cursor-pointer">

<label v-if="\$q.platform.is.desktop" for="material-upload" class="cursor-  
pointer">



</label>

<label v-if="\$q.platform.is.mobile" class="cursor-pointer"  
@click="captureImage">



</label>

<input @change="getImageHomework" id="material-upload" type="file"  
style="display: none;" accept="capture=camera,image/\*">

</div>



```

        </div>

    </div>

</div>

</div>

<div class="footer">

    <div class="row" style="width: 100%; padding: 0 5%;">

        <div class="col-12 col-sm-6">

            <div class="controls-container">

                <!-- <q-slider style="border-radius: 0px; width: 100%;" v-model="seekRange"
:min="0" :max="50" @input="getSeek" ref="seek"/> -->

                <input style="border-radius: 0px; width: 100%;" v-model="seekRange"
@input="getSeek" class="" ref="seek" type="range" name="" id="">

                <div class="controls">

                    <div @click="playVideo" class="btn-play" style="margin: 5px;">

                        <button class="cursor-pointer">

                            

                        </button>

                    </div>

                    <div @click="pauseVideo" class="btn-play" style="margin: 5px;">

                        <button class="cursor-pointer">

                            

                        </button>

                    </div>

                    <div @click="fullScreen" class="btn-play" style="margin: 5px;">

                        <button class="cursor-pointer">

                            

                        </button>

                    </div>

                </div>

            </div>

        </div>

    </div>

```

```

    </div>
  </div>
  <div class="col-12 col-sm-6">
    <div class="info">
      <a :href="lesson.details" download target="_blank">
        
      </a>
      <button @click="finishLesson" class="cursor-pointer">
        
      </button>
      <!-- <button @click="binnie" class="cursor-pointer">
        
      </button> -->
    </div>
  </div>
</div>
</div>
</div>
<div v-show="loading" class="main-loading">
  
</div>
<q-dialog v-model="lesson_complete">
  <q-card style="z-index: 10000;">
    <q-card-section class="q-pt-none" style="">
      <video ref="video_complete" style="z-index: 10000;" width="100%"
src="~/assets/completado.mp4"></video>
    </q-card-section>
  </q-card>
</q-dialog>

```

```

<div class="check-status" v-if="check_status">
  <p>¿Como te sientes?</p>
  <div class="mood">
    <div class="mood-image" v-for="(item, index) in status" :key="index">
      <div class="status-container" @click="checkStatus(index)">
        
        <span>{{ item.title }}</span>
      </div>
    </div>
  </div>
</div>
</q-page>
</template>

```

```

<script>
import { db, storage, firebase } from 'boot/firebase'
const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'

export default {
  name: 'Lesson',

  props: [
    'id',
    'next_lesson',
    'finished_lessons'

```

],

data(){

  return{

    isPlaying: false,

    isMuted: false,

    volume: 0.5,

    seekDuration: 0,

    videoDuration: 0,

    videoTimeElapsed: 0,

    videoPlayer: null,

    loading: true,

    seekRange: 0,

    videoFinish: false,

    lesson: {

      url: "

    },

    homework: null,

    lesson\_homework: null,

    lesson\_complete: false,

    check\_status: false,

    status1: null,

    status2: null,

    imageSrc: "",

    status: [



```
{
  ico: 'Enojado.png',
  title: 'Enojado'
},
{
  ico: 'Triste.png',
  title: 'Triste'
},
{
  ico: 'Cansado.png',
  title: 'Cansado'
},
{
  ico: 'Feliz.png',
  title: 'Feliz'
},
{
  ico: 'Emocionado.png',
  title: 'Emocionado'
},
],
}
},

mounted(){
  window.screen.orientation.lock('landscape')

},
```

```
created() {  
  this.getLesson()  
  this.getHomework()  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ])  
},
```

```
watch: {  
  loading(){  
    if (this.loading) {  
      this.$q.loading.show({  
        message: 'Some important process is in progress. Hang on...'  
      })  
    }else{  
      this.$q.loading.hide()  
    }  
  },
```

```
videoDuration(){  
  console.log('Cambioooooo: ', this.videoDuration);  
  this.videoPlayer.currentTime = 0  
},
```

```
videoTimeElapsed(){  
  if (this.videoTimeElapsed == this.videoDuration) {
```

```
        this.videoFinish = true
    }else{
        this.videoFinish = false
    }
},
```

```
videoFinish(){
    if (this.videoFinish) {
        this.pauseVideo()
        this.videoPlayer.currentTime = 0
    }
},
```

```
lesson_complete(){
    if (!this.lesson_complete) {
        this.$router.push(`/lessons/${this.lesson.course}`)
    }
}
},
```

```
methods: {
    playModal(){
        this.$refs.video_complete.play()
    },
```

```
getImageHomework(e){
    console.log(e.target.files[0]);
    this.homework = e.target.files[0]
},
```

```

captureImage () {

    navigator.camera.getPicture((imageData ) => {

        this.homework = imageData

    }, () => {

        this.$q.notify('Could not access device camera.')

    }, { quality: 50,

        destinationType: Camera.DestinationType.DATA_URL });

},

async checkStatus(index){

    try {

        if (this.status1 == null){

            this.status1 = index

            this.check_status = false

        }else{

            this.status2 = index

            this.check_status = false

            this.registerLesson()

        }

    } catch (error) {

        console.log(error);

    }

},

```

```

async getHomework(){
  try {
    let response = await db.collection('homework')
      .where('user', '==', this.user.uid)
      .where('lesson', '==', this.id)
      .get()
      .then(query => {
        query.forEach(doc => {
          this.lesson_homework = doc.data()
        })
      })

    if (this.lesson_homework == null) this.check_status = true
  } catch (error) {
    console.log(error);
  }
},

async uploadPDF(data){
  try {
    let metadata = {
      contentType: "application/pdf",
      contentLanguage: "es",
    }

    let homeworkID = this.generateUUID()
    const refImg = ref.child('homework/' + homeworkID + '.jpg')

    if (this.$q.platform.is.mobile) {
      return refImg.putString(data, 'base64')
    }
  }
}

```

```

        .then(e => {
            console.log(e)
            return ref.child('homework/' + homeworkID + '.jpg').getDownloadURL()
            .then(async (url) => {
                return url
            })
        })

    })

    .catch(error => console.log(error))
}

return refImg.put(data, metadata)

.then(e => {
    console.log(e)
    return ref.child('homework/' + homeworkID + '.jpg').getDownloadURL()
    .then(async (url) => {
        return url
    })
})

})

.catch(error => console.log(error))
} catch (error) {
    console.log(error);
}
},

finishLesson(){

```

```
if (this.lesson_homework != null) {  
  this.$q.dialog({  
    title: 'Finalizado',  
    message: 'Ya terminaste esta lección.'  
  }).onOk(() => {  
  
  })  
  
  return  
}  
  
this.$q.dialog({  
  title: 'Finalizar clase',  
  message: '¿Quieres finalizar esta clase?',  
  cancel: true,  
  persistent: true  
}).onOk(async () => {  
  try {  
    this.check_status = true  
  
  
  } catch (error) {  
    console.log(error);  
  }  
});  
  
},
```

```

binnie(){
  this.lesson_complete = true
  setTimeout(() => {
    console.log(this.$refs.video_complete);
    this.$refs.video_complete.play()
  }, 100);
},

setTimeRedirect(){
  setTimeout(() => {
    this.lesson_complete = false
  }, 2000);
},

async registerLesson(){
  try {
    let found = this.finished_lessons.lessons.find(doc => doc.id == this.id)

    if (!found) {
      this.lesson_complete = true
      setTimeout(() => {
        console.log(this.$refs.video_complete);
        this.$refs.video_complete.play()

        this.setTimeRedirect()
      }, 100);

      if (this.next_lesson == undefined) {
        this.$q.dialog({

```



```

        title: 'Finalizado',
        message: 'Terminaste el curso.'
    }).onOk(() => {

    })
} else {
    let response = await db.collection('finished_courses')
        .doc(this.user.uid)
        .collection('courses')
        .doc(this.lesson.course)
        .update({ lessons:
firebase.firestore.FieldValue.arrayUnion(this.next_lesson.id) })
    }

}

this.registerLessonsActivity()
} catch (error) {
    console.log(error);
}
},

async registerLessonsActivity(){
    try {
        let materialUrl = null

        if (this.homework == null) {
            materialUrl = 'https://firebasestorage.googleapis.com/v0/b/kumari-
d465a.appspot.com/o/clases%2F016e7de3-b48d-4e31-8066-
0888014f75fb.jpg?alt=media&token=92e803cf-837d-4921-a271-32938b2959a2'

```

```

}else{

    materialUrl = await this.uploadPDF(this.homework)

    this.homework = null

}

let data = {

    user: this.user.uid,

    lesson: this.id,

    course: this.lesson.course,

    homework: materialUrl,

    date: new Date(),

    dateUnix: moment().unix(),

    status1: this.status1,

    status2: this.status2

}

let sticker = {

    clase: this.id,

    user: this.user.uid

}

let response = await db.collection('homework')

    .add(data)

let stickerResponse = await db.collection('stickers')

    .add(sticker)

this.lesson_homework = data

} catch (error) {

```

$$\left. \begin{array}{l} \} \\ \} \end{array} \right\}$$

```
async getLesson(){
  try {
    let response = await db.collection('classes')
      .doc(this.id)
      .get()
```

```
this.lesson = response.data()
```

```
this.initializeAll()
```

```
} catch (error) {  
    console.log(error);  
}
```

 $\},$ 

```
initializeAll(){
  this.videoPlayer = this.$refs.video_player

  this.videoPlayer.addEventListener('loadedmetadata', () => {
    console.log(this.videoPlayer.duration);

    if (this.videoPlayer.duration == Infinity) {
      console.log('binnieeeeeeeeeeeeeee');
      this.videoPlayer.currentTime = 10000;
      this.videoPlayer.ontimeupdate = () => {
        this.ontimeupdate = () => {
          return;
        }
      }
    }
  })
}
```

```

    }

    this.initializeVideo()

    console.log('daleeeee');

    return;
  }
}else{
  this.initializeVideo()
}
})

this.videoPlayer.addEventListener('timeupdate', this.updateTimeElapsed);
this.videoPlayer.addEventListener('timeupdate', this.updateProgress);
},

```

```

initializeVideo() {
  let videoDuration = Math.round(this.videoPlayer.duration)

  let time = this.formatTime(videoDuration)

  this.videoDuration = `${time.minutes}:${time.seconds}`

  this.$refs.seek.setAttribute('max', this.videoPlayer.duration);
  // this.$refs.progress_bar.setAttribute('max', this.videoPlayer.duration);

  this.videoPlayer.volume = 0.2

  this.loading = false
},

```

```

updateTimeElapsed() {
  let time = this.formatTime(Math.round(this.videoPlayer.currentTime));

  this.videoTimeElapsed = `${time.minutes}:${time.seconds}`;
},

```

```
updateProgress() {  
    this.$refs.seek.value = Math.floor(this.videoPlayer.currentTime);  
    // if (this.videoTimeElapsed == this.videoDuration) {  
    //     alert('termino el video')  
    // }  
    // this.$refs.progress_bar.value = Math.floor(this.videoPlayer.currentTime);  
},
```

```
undoSkip(){  
    this.videoPlayer.currentTime = this.videoPlayer.currentTime - 10  
},
```

```
getSeek($event){  
    this.videoPlayer.currentTime = $event.target.value  
    this.$refs.seek.value = $event.target.value  
    // this.$refs.progress_bar.value = $event.target.value  
},
```

```
skip(){  
    this.videoPlayer.currentTime = this.videoPlayer.currentTime + 10  
},
```

```
playVideo(){  
  
    this.videoPlayer.play()  
    this.isPlaying = true  
},
```

```
clickVideo(){
```

```
    if (this.isPlaying == false) {  
        this.playVideo()  
    }else{  
        this.pauseVideo()  
    }  
},
```

```
pauseVideo(){  
    this.videoPlayer.pause()  
    this.isPlaying = false  
},
```

```
mutedVideo(){  
    this.videoPlayer.muted = true  
    this.isMuted = true  
},
```

```
unmutedVideo(){  
    this.videoPlayer.muted = false  
    this.isMuted = false  
},
```

```
fullScreen(){  
    if (document.fullscreenElement) {  
        document.exitFullscreen();  
    } else if (document.webkitFullscreenElement) {  
        // Need this to support Safari  
        document.webkitExitFullscreen();  
    } else if (this.videoPlayer.webkitRequestFullscreen) {
```

```
// Need this to support Safari
this.videoPlayer.webkitRequestFullscreen();

} else {

    this.videoPlayer.webkitEnterFullscreen();

}

},


changeVol(){

    this.videoPlayer.volume = this.volume

},


formatTime(timeInSeconds) {

    const result = new Date(timeInSeconds * 1000).toISOString().substr(11, 8);
    return {

        minutes: result.substr(3, 2),

        seconds: result.substr(6, 2),

    };

},


//Generar UUID
generateUUID(){

    var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];

    var k = ['x','x','x','x','x','x','x','x','-','x','x','x','x','-','4','x','x','x','-', 'y','x','x','x','-', 'x','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];

    var u = "", i=0, rb=Math.random()*0xffffffff|0;

    while(i++<36) {

        var c=k[i-1], r=rb&0xf, v=c=='x'?r:(r&0x3|0x8);

        u+=(c=='-' || c=='4')?c:h[v]; rb=i%8==0?Math.random()*0xffffffff|0:rb>>4

    }

}
```

```
        return u
    },

}

}

</script>
```

```
<style>
```

```
</style>
```

[Lessons.txt](#)

```
<style scoped lang="scss">
```

```
.content{
    position: relative;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    background-color: rgba(0, 0, 0, 0.5);
```

```
.header{
    width: 100%;
```

```
.for-dad{
    width: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
```

```
    button{
```



```
background-color: orange;

color: white;

border: none;

border-radius: 0 0 10px 10px;


font-size: 22px;

padding: 5px 50px;

}

}
```

```
.header-menu{

width: 100%;

display: flex;

justify-content: space-between;

align-items: center;
```

```
.btn-logout{

a{

width: 80px;

height: 80px;

display: flex;

justify-content: center;

align-items: center;

border-radius: 50%;

margin-right: 10px;
```

```
img{

width: 50px;

}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
.main-container{
```

```
width: 100%;
```

```
overflow-x: hidden;
```

```
.course-container{
```

```
display: flex;
```

```
justify-content: center;
```

```
align-items: center;
```

```
padding: 10px;
```

```
cursor: pointer;
```

```
.course-body{
```

```
display: flex;
```

```
justify-content: center;
```

```
align-items: center;
```

```
background-color: white;
```

```
border-radius: 10px;
```

```
aspect-ratio: 16/9;
```

```
position: relative;
```

```
object-fit: cover;
```

```
object-position: center;
```

```
img{  
  width: 80%;  
}
```

```
.layer{  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
  border-radius: 10px;  
  background-color: rgba(0, 0, 0, 0.0);
```

```
&>div{  
  width: 100%;  
  text-align: right;  
  display: flex;  
  justify-content: flex-end;  
  align-items: flex-end;
```

```
  img{  
    width: 15%;  
    margin: 2%;  
  }  
}
```

```
.layer-disabled{  
  background-color: rgba(0, 0, 0, 0.7);
```

```
}  
}
```

```
}
```

```
.name-title{  
    text-align: center;  
    font-size: 22px;  
    font-weight: bold;  
    color: white;  
}
```

```
.slick-current{  
    padding-bottom: 5px;  
    border-bottom: 2px solid #006ab2;  
    color: blue;  
}  
}
```

```
.footer{  
    visibility: hidden;  
}
```

```
.check-status{  
    width: 100%;  
    height: 100%;  
    background-color: rgba(0, 0, 0, 0.9);  
    position: absolute;
```

```
top: 0;
left: 0;
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
color: white;
```

```
.mood{
  display: flex;
  justify-content: center;
  align-items: center;
```

```
  .mood-image{
    padding: 10px;
    img{
      width: 90px;
    }
  }
}
```

```
header{
  width: 100%;
```

```
.box-1{
  display: flex;
  justify-content: center;
```

```
align-items: center;
```

```
img{  
    width: 55%;  
}  
}
```

```
.box-2{  
    display: flex;  
    justify-content: space-around;  
    align-items: flex-start;  
}
```

```
.box-3{  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    padding-top: 15px;  
    text-align: center;
```

```
a{  
    img{  
        width: 50%;  
        border-radius: 50%;  
        box-shadow: 5px 5px 5px rgba(46, 46, 46, 0.377);  
    }  
}  
}
```

```
@media only screen and (max-device-width: 900px) and (orientation: landscape) {
```

```
  header{
```

```
    .box-1{
```

```
      img{
```

```
        width: 75%;
```

```
      }
```

```
    }
```

```
    .box-2{
```

```
      .for-dad{
```

```
        a{
```

```
          font-size: 18px !important;
```

```
          padding: 5px 20px !important;
```

```
        }
```

```
      }
```

```
    }
```

```
    .box-3{
```

```
      a{
```

```
        img{
```

```
          width: 30%;
```

```
        }
```

```
      }
```

```
    }  
  }  
  .main-container{  
    width: 100%;  
  }  
  
  .swiper-container{  
    margin-top: 5% !important;  
  
    .name-title{  
      font-size: 14px !important;  
    }  
  }  
}
```

```
.courses-web{  
  display: block;  
}
```

```
.courses-movil{  
  display: none;  
}
```

```
@media only screen  
  and (max-width : 1135px)  
  and (orientation : portrait) {  
    .courses-web{  
      display: none;  
    }  
  }
```





```

        <ForFathers />

        <ButtonStickers />
    </div>
</div>
<div class="col-2">
    <div class="box-3">
        <router-link to="/dashboard">
            
        </router-link>
    </div>
</div>
</div>
</div>
</header>
<div class="main-container courses-web">
    <swiper ref="mySwiper" :options="swiperOptions">
        <swiper-slide v-for="(item, index) in orderLessons" :key="index">
            <div @click="goToLesson(item, index)" class="course-container">
                <div class="course-body" :style="{ backgroundColor: course.color }">
                    
                    <div class="layer" :class="[finished_lessons.lessons.includes(item.id) || item.order
== 1 ? '' : 'layer-disabled']">
                        <div v-if="haveSticker(item.id)">
                            
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <p class="name-title" style="color: white;">{{ item.name }}</p>

```

```

        </swiper-slide>
    </swiper>
</div>

<div class="main-container courses-movil">
    <div>
        <div v-for="(item, index) in orderLessons" :key="index">
            <div @click="goToLesson(item, index)" class="course-container">
                <div class="course-body" :style="{ backgroundColor: course.color }">
                    
                    <div class="layer" :class="['finished_lessons.lessons.includes(item.id) || item.order
== 1 ? '' : 'layer-disabled']">
                        <div v-if="haveSticker(item.id)">
                            
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="name-title" style="color: white;">{{ item.name }}</div>
    </div>
</div>
</div>

<div class="footer">
    <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Possimus, beatae accusantium
natus atque maiores suscipit quam veniam eaque excepturi, dolorem incidunt esse, nesciunt nam
ullam in ipsum ducimus soluta fugit?</p>
</div>

```

```
</q-page>
</template>
```

```
<script>
import { db, storage } from 'boot/firebase'
const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'
import VueSlickCarousel from 'vue-slick-carousel'
import 'vue-slick-carousel/dist/vue-slick-carousel.css'
// optional style for arrows & dots
import 'vue-slick-carousel/dist/vue-slick-carousel-theme.css'
import ButtonStickers from './components/ButtonStickers'
import ButtonTickets from './components/ButtonTickets'
import ForFathers from './components/ForFathers'
import { Swiper, SwiperSlide, directive } from 'vue-awesome-swiper'
// import style (>= Swiper 6.x)
import 'swiper/swiper-bundle.css'

let divs = document.getElementsByClassName('slick-current')
console.log(divs);
for (let item of divs) {
  console.log(item);
}

export default {
```

```
name: 'Lessons',
```

```
components: {
```

```
  VueSlickCarousel,
```

```
  ButtonStickers,
```

```
  ButtonTickets,
```

```
  ForFathers,
```

```
  Swiper,
```

```
  SwiperSlide
```

```
},
```

```
props: [
```

```
  'id',
```

```
],
```

```
data(){
```

```
  return{
```

```
    showModalTotorial: false,
```

```
    swiperOptions: {
```

```
      slidesPerView: 4,
```

```
      spaceBetween: 10
```

```
    },
```

```
    settings: {
```

```
      "arrows": false,
```

```
      "dots": false,
```

```
      "focusOnSelect": true,
```

```
      "infinite": false,
```

```
      "speed": 500,
```

```
    "slidesToShow": 3,  
    "touchThreshold": 5,  
    "centerMode": false,  
    "centerPadding": "20px",  
  },
```

```
    course: null,  
    lessons: [],  
    finished_lessons: {  
      lessons: []  
    },  
    handleClick: true,  
    stickers: [],  
  }  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ]),
```

```
  orderLessons(){  
    return this.lessons.sort((a,b) => (a.order > b.order) ? 1 : ((b.order > a.order) ? -1 : 0))  
  }  
},
```

```
mounted(){  
  window.screen.orientation.lock('landscape')  
},
```

```
created() {
```

```

    this.getCourse()
  },

  methods: {
    haveSticker(id){
      let found = this.stickers.find(doc => doc.clase == id)

      if (found) {
        return true
      }else{
        return false
      }
    },

    handleUp(){
      this.handleClick = false
      console.log(this.handleClick);
    },

    handleDown(){
      this.handleClick = true
      console.log(this.handleClick);
    },

    async getCourse(){
      try {
        let response = await db.collection('courses')
          .doc(this.id)
          .get();
      }
    }
  }
}

```

```

    this.course = response.data()

    this.getLessons()
  } catch (error) {
    console.log(error);
  }
},

goToLesson(item, index){
  setTimeout(() => {
    try {
      if (this.handleClick) {
        let found = this.finished_lessons.lessons.find(doc => doc.id === item.id)

        if (found || item.order === 1) {
          console.log(index);

          let next_lesson = this.orderLessons[index + 1]

          this.$router.push({ name: 'Lesson', params: { id: item.id, next_lesson: next_lesson,
finished_lessons: this.finished_lessons } })
        }else{
          this.$q.dialog({
            title: 'Lección bloqueada',
            message: 'Esta Lección se encuentra bloqueada.'
          }).onOk(() => {

          })
        }
      }
    }
  })
}

```



```

    }

    } catch (error) {
        console.log(error);
    }
}, 100)
},

async getLessons(){
    try {
        let response = await db.collection('clases')
            .where('course', '==', this.id)
            .get()
            .then(query => {
                query.forEach(doc => {
                    let lesson = doc.data()

                    lesson.id = doc.id

                    this.lessons.push(lesson)
                })
            })

        this.getFinishedLessons()
        this.getStickers()
    } catch (error) {
        console.log(error);
    }
},

```

```

async getStickers(){
  try {
    let response = await db.collection('stickers')
      .where('user', '==', this.user.uid)
      .get()
      .then(query => {
        query.forEach(doc => {
          let sticker = doc.data()

          this.stickers.push(sticker)
        })
      })
  } catch (error) {
    console.log(error);
  }
},

```

```

async getFinishedLessons(){
  try {
    let response = await db.collection('finished_courses')
      .doc(this.user.uid)
      .collection('courses')
      .doc(this.id)
      .get()

    this.finished_lessons = response.data()
  } catch (error) {
    console.log(error);
  }
}

```

```

    }
  },

},
}
</script>

```

### Actions.txt

```

import { auth, db } from 'boot/firebase'

import { Notify } from 'quasar'

export async function loginUser ({commit, dispatch}, payload) {
  console.log('Desde vuex: ', payload);
  try {
    return new Promise((resolve, reject) =>{
      auth.signInWithEmailAndPassword(payload.email, payload.password)
        .then(response => {
          resolve(response)
        }).catch(error => {
          reject(error)
        })
    })
  } catch (error) {
    console.log(error);
  }
}

export async function registerUser ({commit}, payload) {
  console.log('Desde vuex: ', payload);

```

```

try {
  return new Promise((resolve, reject) =>{
    auth.createUserWithEmailAndPassword(payload.email, payload.password)
      .then(async (userCredential) => {
        console.log(userCredential);
        let user = {
          name: payload.name,
          email: payload.email,
          uid: userCredential.user.uid,
          active: payload.active,
          role: payload.role,
          photo:
'https://i.pinimg.com/originals/87/6f/38/876f386e2ce509c094b932e34211ea14.jpg',
        }
        await db.collection('users').doc(userCredential.user.uid).set(user)

        await auth.signInWithEmailAndPassword(payload.email, payload.password)
          .then(response => {
            console.log('Fierestore dice: ', response);
          }).catch(error => {
            console.log(error);
          })

        resolve(response)
      })
    .catch((error) => {
      console.log(error);
    })
  })
}

```

```

        reject(error)
    });
})
} catch (error) {
    console.log(error);
}
}

```

```

export async function getDailyActivation({commit, dispatch}, payload){
    try {
        let response = await db.collection('daily_activation')
            .doc('current')
            .get()

        commit('updateDailyActivation', response.data())
    } catch (error) {
        console.log(error);
    }
}

```

```

export async function getUser({commit, dispatch}, payload){
    try {
        //Obtenemos la activacion diaria
        dispatch('getDailyActivation')

        let document = await db.collection('users').doc(payload).get()

        if (document.exists) {

```

```
let user = document.data()

console.log('Este es el user: ', user);

if (user.role !== 'user' ) {
  Notify.create({
    message: 'Parece que no tienes los permisos necesarios.',
    color: 'negative'
  })

  auth.signOut()
  commit('updateUser', null)
  return
}

commit('updateUser', user)

dispatch('getCourses', user.uid)

this.$router.replace('/dashboard')

}
} catch (error) {
  console.log(error)
}
}

export async function getCourses({ commit }, payload) {
```

```

try {
  let list_courses = []

  let response = await db.collection('shopping')
    .where('user', '==', payload)
    .get()
    .then(query => {
      query.forEach(async course => {
        let co = await db.collection('courses')
          .doc(course.data().course)
          .get()

        if (co.exists) {
          let cour = co.data()
          cour.id = course.data().course
          list_courses.push(cour)
        }

      })
    })

  commit('updateListCourses', list_courses)
} catch (error) {
  console.log(error);
}

export function closeSesion({ commit }) {
  auth.signOut()
}

```

```
    commit('updateUser', null)

    this.$router.replace('/')
  }
}
```

### Mutations.txt

```
export function updateUser (state, payload) {
  state.user = payload
}
```

```
export function updateListCourses (state, payload) {
  state.courses = payload
}
```

```
export function updateCheckStatus (state, payload) {
  state.user.check_status = payload
}
```

```
export function updateDailyActivation (state, payload) {
  state.daily_activation = payload
}
```

```
export function updateTutorialUser (state, payload) {
  state.user.tutoriañ = payload
}
```

### Routes.txt

```
const routes = [
  {
    path: '/',
    component: () => import('layouts/AuthLayout.vue'),
    children: [
```



```

    { path: '', component: () => import('pages/auth/Login.vue') }
  ],
},
{
  path: '/register',
  component: () => import('layouts/AuthLayout.vue'),
  children: [
    { path: '', component: () => import('pages/auth/Register.vue') }
  ],
},
{
  path: '/user',
  component: () => import('layouts/MainLayout.vue'),
  children: [
    { path: '/daily-activation', component: () => import('src/pages/user/DailyActivation.vue') },
    { path: '/dashboard', component: () => import('pages/user/Dashboard.vue') },
    { path: '/lessons/:id', component: () => import('pages/user/Lessons.vue'), props: true },
    { name: 'Lesson', path: '/lesson/watch/:id', component: () =>
import('pages/user/Lesson.vue'), props: true },
  ],
},

// Always leave this as last one,
// but you can also remove it
{
  path: '*',
  component: () => import('pages/Error404.vue')
}
]

```

export default routes

### state.txt

```
export default function () {  
  return {  
    user: null,  
    courses: null,  
    daily_activation: null,  
  }  
}
```

## Web

### Admin

#### Courses.txt

```
<style scoped lang="scss">  
  
  .main-container{  
    background-color: white;  
    border-radius: 10px;  
    padding: 20px;  
  }  
  
  .main-title{  
    font-size: 22px;  
    font-weight: bold;  
  }  
  
  .btn-options{  
    border: none;
```

```
margin: 5px;
font-size: 18px;
color: white;
border-radius: 5px;
cursor: pointer;
}
```

```
.btn-modal{
border: none;
background-color: #24B8CB;
color: white;
border-radius: 15px;
padding: 10px 15px;
font-weight: bold;
font-size: 17px;
}
```

```
.form-group{
margin-top: 20px;

label{
font-weight: bold;
font-size: 18px;
}
}
```

```
.input-text{
width: 100%;
border: 2px solid #666666;
```

```
border-radius: 10px;  
padding: 5px;  
}
```

```
.input-text:focus{  
    outline: none;  
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;  
}
```

```
.badge-color{  
    width: 20px;
```

```
height: 20px;

border-radius: 50%;

margin-right: 10px;

cursor: pointer;

}

.active{

  box-shadow: 1px 1px 10px rgba(0, 0, 0, 0.8);

}

.layer-loading{

  width: 100%;

  height: 100%;

  position: absolute;

  top: 0;

  left: 0;

  background-color: rgba(0, 0, 0, 0.8);

  z-index: 10000%;

  display: flex;

  justify-content: center;

  align-items: center;

}

</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<q-table
```

```

    title="Cursos"

    :data="courses"

    :columns="columns"

    :filter="filter"

    row-key="name"

    flat

>

<template v-slot:top>

    <div class="flex justify-between items-center" style="width: 100%;">

        <span style="font-size: 25px; font-weight: bold;">Cursos</span>

        <button class="btn-modal cursor-pointer" @click="modalCourse = true">Crear
curso</button>

        <button class="btn-modal cursor-pointer" @click="modalAllCourses = true">Editar
todos los cursos</button>

    </div>

    <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">

        <template v-slot:append>

            <q-icon name="search" />

        </template>

    </q-input>

</div>

</template>

<template v-slot:body-cell-instructor="props">

    <q-td :props="props">

        {{ props.row.user.name }}

    </q-td>

</template>

```

```

<template v-slot:body-cell-clases="props">
  <q-td :props="props">
    {{ props.row.clases.length }}
  </q-td>
</template>

<template v-slot:body-cell-status="props">
  <q-td :props="props">
    <span v-if="props.row.status">Activo</span>
    <span v-else>No activo</span>
  </q-td>
</template>

<template v-slot:body-cell-options="props">
  <q-td :props="props">
    <button class="btn-options" style="background-color: #55B995;"
      @click="modalEditCourse = true, editCourse = props.row">
      <q-icon name="edit" />
    </button>
    <button class="btn-options" style="background-color: #F3920B;"
      @click="deleteCourse(props.row)">
      <q-icon name="delete" />
    </button>
  </q-td>
</template>
</q-table>
</div>

<q-dialog v-model="modalAllCourses" persistent v-if="allCourses != null">
  <q-card style="min-width: 800px;">
    <q-card-section class="row">
      <div class="col-12">

```

```

<q-form
  @submit="addAllCoursesEdit"
  class="q-gutter-md"
>
  <div class="row">
    <div class="col-6 q-pa-md">
      <div class="form-group">
        <label for="">Curso</label> <br>
        <input class="input-text" type="text" v-model="allCourses.name">
      </div>
      <div class="form-group">
        <label for="">Descripción</label> <br>
        <textarea class="input-text" v-model="allCourses.description"></textarea>
      </div>
      <div class="form-group">
        <label for="">Price</label> <br>
        <input class="input-text" type="text" v-model="allCourses.price">
      </div>
      <div class="form-group">
        <input type="file" @change="getImageAllCourses($event)"
accept="image/">
      </div>
    </div>
    <div class="col-6 q-pa-md">
      <div class="form-group">
        <div>
          <label for="">Cursos</label> <br>
          <div v-for="(course, index) in courses" :key="index">

```



```

        <label style="font-weight: normal;">
            <input v-model="allCourses.courses" :value="course.id"
type="checkbox" name="interes"> {{ course.name }}
        </label>
        <br>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalAllCourses =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>

<div class="layer-loading" v-if="isLoading">
    
</div>
</q-card>
</q-dialog>

<q-dialog v-model="modalCourse" persistent>
    <q-card style="min-width: 800px;">
        <q-card-section>
            <p class="main-title">Agregar nuevo curso</p>

```

```

</q-card-section>

<q-card-section class="row">

  <div class="col-12">

    <q-form
      @submit="addCourse"
      class="q-gutter-md"
    >

      <div class="row">

        <div class="col-6 q-pa-md">

          <div class="form-group">

            <label for="">Curso</label> <br>

            <input class="input-text" type="text" v-model="newCourse.name">

          </div>

          <div class="form-group">

            <label for="">Descripción</label> <br>

            <textarea class="input-text" v-model="newCourse.description"></textarea>

          </div>

          <div class="form-group">

            <label for="">Instructor</label>

            <br>

            <q-select
              borderless
              class="input-text"
              v-model="newCourse.instructor"
              use-input
              input-debounce="0"
              :options="optionsUsers"
              @filter="filterFnUser"
              behavior="dialog"
            >

```

```

        option-label="name"
        option-value="uid"
        emit-value
        map-options
    >
    <template v-slot:no-option>
    <q-item>
        <q-item-section class="text-grey">
            Sin resultados
        </q-item-section>
    </q-item>
    </template>
</q-select>
</div>
<div class="form-group">
    <vue-tags-input
        placeholder="Edades"
        v-model="tagAge"
        :tags="newCourse.age"
        @tags-changed="newTagsAge => newCourse.age = newTagsAge"
    />
</div>

<div>
    <vue-tags-input
        placeholder="Palabras clave"
        v-model="tag"
        :tags="newCourse.clave_words"
        @tags-changed="newTags => newCourse.clave_words = newTags"
    >

```

```

        />
    </div>

</div>

<div class="col-6 q-pa-md">
    <div class="form-group">
        <input type="file" @change="getImage($event)" accept="image/">
    </div>
    <div class="form-group">
        <label for="">Color de Acento</label>
        <div class="flex">
            <div class="badge-color" @click="newCourse.color = item" :class="[
newCourse.color == item ? 'active' : '' ]" :style="{ backgroundColor: item }" v-for="(item, index) in
palette" :key="index">

                </div>
            </div>
        </div>
        <div class="form-group">
            <label for="">Estatus</label>
            <div>
                <select class="input-text" v-model="newCourse.status">
                    <option value="Activo">Activo</option>
                    <option value="Pausado">Pausado</option>
                </select>
            </div>
        </div>
    </div>
</div>
<div class="q-pa-md flex justify-end">

```

```
        <button class="btn-cancel" type="button" @click="modalCourse =
false">Cerrar</button>
```

```
        <button class="btn-submit" type="submit">Guardar</button>
```

```
    </div>
```

```
</q-form>
```

```
</div>
```

```
</q-card-section>
```

```
</q-card>
```

```
</q-dialog>
```

```
<q-dialog v-model="modalEditCourse" persistent v-if="editCourse != null">
```

```
<q-card style="min-width: 800px;">
```

```
<q-card-section>
```

```
<p class="main-title">Editar curso</p>
```

```
</q-card-section>
```

```
<q-card-section class="row">
```

```
<div class="col-12">
```

```
<q-form
```

```
    @submit="addCourseEdit"
```

```
    class="q-gutter-md"
```

```
>
```

```
<div class="row">
```

```
<div class="col-6 q-pa-md">
```

```
<div class="form-group">
```

```
<label for="">Curso</label> <br>
```

```
<input class="input-text" type="text" v-model="editCourse.name">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Descripción</label> <br>
```

```

        <textarea class="input-text" v-model="editCourse.description"></textarea>
    </div>
    <div class="form-group">
        <label for="">Instructor</label>
        <br>
        <q-select
            borderless
            class="input-text"
            v-model="editCourse.instructor"
            use-input
            input-debounce="0"
            :options="optionsUsers"
            @filter="filterFnUser"
            behavior="dialog"
            option-label="name"
            option-value="uid"
            emit-value
            map-options
        >
            <template v-slot:no-option>
                <q-item>
                    <q-item-section class="text-grey">
                        Sin resultados
                    </q-item-section>
                </q-item>
            </template>
        </q-select>
    </div>
    <div class="form-group">

```

```

<label for="">Edades</label> <br>

<vue-tags-input
  placeholder="Palabras clave"
  v-model="tag"
  :tags="editCourse.age"
  @tags-changed="newTags => editCourse.age = newTags"
/>

```

```

</div>

```

```

<div>
  <vue-tags-input
    placeholder="Palabras clave"
    v-model="tag"
    :tags="editCourse.clave_words"
    @tags-changed="newTags => editCourse.clave_words = newTags"
  />
</div>

```

```

</div>

```

```

<div class="col-6 q-pa-md">

```

```

  <div class="form-group">

```

```

    <input type="file" @change="getImage($event)" accept="image/">

```

```

  </div>

```

```

  <div class="form-group">

```

```

    <label for="">Color de Acento</label>

```

```

    <div class="flex">

```

```

      <div class="badge-color" @click="editCourse.color = item" :class="[
editCourse.color == item ? 'active' : '' ]" :style="{ backgroundColor: item }" v-for="(item, index) in
palette" :key="index">

```

```

        </div>
    </div>
</div>
<div class="form-group">
    <label for="">Estatus</label>
    <div>
        <select class="input-text" v-model="editCourse.status">
            <option value="Activo">Activo</option>
            <option value="Pausado">Pausado</option>
        </select>
    </div>
</div>
<div class="form-group">
    <label for="">Precio</label>
    <div>
        <input type="number" v-model="editCourse.price" class="input-text">
    </div>
</div>
</div>
<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalEditCourse =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>

```



```

        </q-card>
    </q-dialog>
</q-page>
</template>

<script>
import { db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import VueTagsInput from '@johmun/vue-tags-input';

export default {
  name: 'feed',

  components: {
    VueTagsInput,
  },

  data(){
    return{
      isLoading: false,

      newCourse: {
        name: "",
        age: [],
        clave_words: [],
        photo: "",
        color: "",

```

```
    status: null,
  },

  allCourses: null,

  users: [],
  optionsUsers: null,

  palette: [
    '#019a9d', '#d9b801', '#e8045a', '#b2028a',
    '#2a0449',
  ],

  tag: "",
  tagAge: "",
  modalCourse: false,
  modalAllCourses: false,
  modalEditCourse: false,
  editCourse: null,

  courseImage: null,
  allCourseImage: null,

  courses: [],
  filter: "",
  columns: [
    {
      name: 'name',
      required: true,
```

```

        label: 'Nombre',
        align: 'left',
        field: row => row.name,
        format: val => `${val}`,
        sortable: true
    },
    { name: 'instructor', align: 'left', label: 'Instructor', field: 'instructor', sortable: true },
    { name: 'price', align: 'left', label: 'Precio', field: 'price', sortable: true },
    { name: 'clases', align: 'left', label: 'Clases', field: 'clases', sortable: true },
    { name: 'status', align: 'left', label: 'Estatus', field: 'status' },
    { name: 'options', label: 'Opciones', field: 'options' }
],
}
},

```

```

filters: {
    formatDate(args){
        try {
            let date = moment(args.toDate()).fromNow();
            return date
        } catch (error) {
            let date = moment(args).fromNow();
            return date
        }
    }
},

```

```

created() {
    this.getCourses()
}

```

```

    this.getAllCourses()
    this.getUsers()
  },

  computed: {
    ...mapState('auth', [ 'user' ])
  },

  methods: {
    truncateString(str, num) {
      if (str.length <= num) {
        return str
      }

      return str.slice(0, num) + '...'
    },

    filterFnUser (val, update) {
      if (val === '') {
        update(() => {
          this.optionsUsers = this.users.filter(user => user.role !== 'instructor' || user.role !==
'admin')
        })
        return
      }

      update(() => {
        const needle = val.toLowerCase()
        this.optionsUsers = this.users.filter(v => v.name.toLowerCase().indexOf(needle) > -1)

```

```

    })
  },

  async getUsers(){
    try {
      let response = await db.collection('users')
        .get()
        .then(query => {
          query.forEach(doc => {
            let user = doc.data()
            user.id = doc.id

            this.users.push(user)
          })
        })
    } catch (error) {
      console.log(error);
    }
  },

```

```

  async addCourse(){
    try {
      this.$q.dialog({
        title: 'Agregar nuevo curso',
        message: '¿Quieres registrar este curso?',
        cancel: true,
        persistent: true
      }).onOk(async () => {
        let url = null

```

```

    if (this.courseImage == null) {
        url = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-jihan-cover-680x384.jpg'
    }else{
        url = await this.uploadImage()
        this.courseImage = null
    }
    this.newCourse.photo = url
    this.newCourse.instructor = this.user.uid
    this.newCourse.price = 500

    this.newCourse.created_at = moment().format()
    this.newCourse.price = 100

    this.newCourse.created_atUnix = moment(this.newCourse.created_at).unix()

    let reponse = await db.collection('courses')
        .add(this.newCourse)
        .then(query => {
            this.newCourse.id = query.id
            let course = JSON.parse(JSON.stringify(this.newCourse))
            course.clases = []

            this.courses.push(course)

            this.courseImage = null

            this.modalCourse = false

```

```
        this.newCourse = {  
            name: "",  
            age: "",  
            clave_words: [],  
            photo: "",  
            color: "",  
            status: null,  
        }  
    })  
}
```

```
        this.$q.notify({  
            message: 'Curso registrado',  
            color: 'positive'  
        })  
    })  
}
```

```
    } catch (error) {  
        console.log(error);  
    }  
},  
  
isObject(obj) {  
    return Object.prototype.toString.call(obj) === '[object Object]';  
},  
  
async addCourseEdit(){  
    try {  
        if (this.courselImage == null) {  
            this.editCourse.photo = this.editCourse.photo
```

```

    }else{

        let url = await this.uploadImage()

        this.courserImage = null

        this.editCourse.photo = url

    }

    if (this.isObject(this.editCourse.user)) {

        this.editCourse.user = this.editCourse.user.uid

    }

    let response = await db.collection('courses')

        .doc(this.editCourse.id)

        .update(this.editCourse)

    this.modalEditCourse = false

    this.$q.notify({

        message: 'Curso editado',

        color: 'positive'

    })

} catch (error) {

    console.log(error);

}

},

```

```

async addAllCoursesEdit(){

    this.isLoading = true

    try {

        if (this.allCourseImage == null) {

```



```

        this.allCourses.photo = this.allCourses.photo
    }else{
        let url = await this.uploadImageAllCourses()
        console.log(url);
        this.allCourseImage = null
        this.allCourses.photo = url
    }

    let response = await db.collection('all-courses')
        .doc('course')
        .update(this.allCourses)

    this.modalAllCourses = false

    this.$q.notify({
        message: 'Todos los cursos editado',
        color: 'positive'
    })

    this.isLoading = false
    } catch (error) {
        this.isLoading = false
        console.log(error);
    }
},

async getCourses(){
    this.courses = []

```

```

try {
  let response = await db.collection('courses')
    .get()
  .then(query => {
    query.forEach(async doc => {
      let course = doc.data()
      course.id = doc.id
      course.clases = []

      let user = await db.collection('users')
        .doc(course.instructor)
        .get()

      let clases = await db.collection('clases')
        .where('course', '==', doc.id)
        .get()
        .then(query => {
          query.forEach(doc => {
            let clase = doc.data()
            clase.id = doc.id

            course.clases.push(clase)

          })
        })

      course.user = user.data()
      this.courses.push(course)
    })
  })
}

```

```
    } catch (error) {  
        console.log(error);  
    }  
},
```

```
async getAllCourses(){
```

```
    try {  
        let response = await db.collection('all-courses')  
            .doc('course')  
            .get()  
  
        if (response.exists) {  
            this.allCourses = response.data()  
        }  
    } catch (error) {  
        console.log(error);  
    }  
},
```

```
async deleteCourse(course){
```

```
    try {  
        this.$q.dialog({  
            title: 'Eliminar curso',  
            message: '¿Estas seguro de eliminar este curso?',  
            cancel: true,  
            persistent: true  
        }).onOk(async () => {  
            let index = this.courses.findIndex(doc => doc.id == course.id)
```

```

    let response = await db.collection('courses').doc(course.id).delete()

    this.$q.notify({
      message: 'Curso eliminado',
      color: 'positive'
    })

    this.courses.splice(index, 1)
  })

} catch (error) {
  console.log(error);
}
},

```

```

//Generar UUID
generateUUID(){
  var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
  var k = ['x','x','x','x','x','x','x','x','-','x','x','x','x','-','4','x','x','x','x','-','y','x','x','x','x','-','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];
  var u = "",i=0,rb=Math.random()*0xffffffff|0;
  while(i++<36) {
    var c=k[i-1],r=rb&0xf,v=c=='x'?r:(r&0x3|0x8);
    u+=(c=='-' || c=='4')?c:h[v];rb=i%8==0?Math.random()*0xffffffff|0:rb>>4
  }
  return u
},

```

```

async uploadImageAllCourses(){

```

```

// this.isLoading = true;

try {
  let metadata = {
    contentType: "image/jpeg",
    contentLanguage: "es",
  }

  let fotold = this.generateUUID()

  const refImg = ref.child('courses/' + fotold + '.jpg')

  return refImg.put(this.allCourseImage, metadata)

  .then(e => {
    console.log(e)

    return ref.child('courses/' + fotold + '.jpg').getDownloadURL()

    .then(async (url) => {
      return url

    })

    // this.getResizePath(fotold)

  })

  .catch(error => console.log(error))
} catch (error) {
  console.log(error);
}

},

```

```

async uploadImage(){
  // this.isLoading = true;

  try {
    let metadata = {
      contentType: "image/jpeg",

```

```

        contentLanguage: "es",
    }
    let fotold = this.generateUUID()
    const reflmg = ref.child('courses/' + fotold + '.jpg')
    return reflmg.put(this.courseImage, metadata)
    .then(e => {
        console.log(e)
        return ref.child('courses/' + fotold + '.jpg').getDownloadURL()
        .then(async (url) => {
            return url
        })
        // this.getResizePath(fotold)
    })
    .catch(error => console.log(error))
} catch (error) {
    console.log(error);
}
},

```

```

getImageAllCourses(e){
    this.allCourseImage = e.target.files[0]
},

```

```

getImage(e){
    this.courseImage = e.target.files[0]
    console.log(this.courseImage);
},

```

```
},  
}  
</script>
```

### Instructors.txt

```
<style scoped lang="scss">
```

```
.main-container{  
  background-color: white;  
  border-radius: 10px;  
  padding: 20px;  
}
```

```
.main-title{  
  font-size: 22px;  
  font-weight: bold;  
}
```

```
.btn-options{  
  border: none;  
  margin: 5px;  
  font-size: 18px;  
  color: white;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

```
.btn-modal{  
  border: none;  
  background-color: #24B8CB;  
  color: white;
```

```
border-radius: 15px;  
padding: 10px 15px;  
font-weight: bold;  
font-size: 17px;  
}
```

```
.form-group{  
  margin-top: 20px;  
  
  label{  
    font-weight: bold;  
    font-size: 18px;  
  }  
}
```

```
.input-text{  
  width: 100%;  
  border: 2px solid #666666;  
  border-radius: 10px;  
  padding: 5px;  
}
```

```
.input-text:focus{  
  outline: none;  
}
```

```
.btn-cancel{  
  color: white;  
  padding: 5px 20px;
```



```
border: none;

background-color: #E72B81;

border-radius: 10px;

cursor: pointer;

}
```

```
.btn-submit{

color: white;

padding: 5px 20px;

border: none;

background-color: #24B8CB;

border-radius: 10px;

margin-left: 10px;

cursor: pointer;

}
```

```
.badge-color{

width: 20px;

height: 20px;

border-radius: 50%;

margin-right: 10px;

cursor: pointer;

}
```

```
.active{

box-shadow: 1px 1px 10px rgba(0, 0, 0, 0.8);

}
```

```
.form-group{  
    padding: 10px 5px;  
}
```

```
label{  
    font-size: 18px;
```

```
    input{  
        margin-right: 10px;  
    }  
}
```

```
input[type=text]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
input[type=date]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
input[type=password]{
```

```
width: 100%;  
  
border-radius: 10px;  
  
border: none;  
  
border: 1px solid black;  
  
padding: 5px;  
}
```

```
select{  
  
width: 100%;  
  
border-radius: 10px;  
  
border: none;  
  
border: 1px solid black;  
  
padding: 5px;  
}
```

```
</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<q-table
```

```
title="Instructores"
```

```
:data="users"
```

```
:columns="columns"
```

```
:filter="filter"
```

```
row-key="name"
```

```
flat
```

```
>
```

```
<template v-slot:top>
```

```
<div class="flex justify-between items-center" style="width: 100%;">
```

```

        <span style="font-size: 25px; font-weight: bold;">Instructores</span>

        <button class="btn-modal cursor-pointer" @click="modalUser = true">Crear
instructor</button>

        <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">

            <template v-slot:append>

                <q-icon name="search" />

            </template>

        </q-input>

    </div>

</template>

<template v-slot:body-cell-education="props">

    <q-td :props="props">

        {{ props.row.education }}

    </q-td>

</template>

<template v-slot:body-cell-courses="props">

    <q-td :props="props">

        {{ props.row.courses.length }}

    </q-td>

</template>

<template v-slot:body-cell-students="props">

    <q-td :props="props">

        4

    </q-td>

</template>

<template v-slot:body-cell-status="props">

    <q-td :props="props">

        <span v-if="props.row.status">Activo</span>

```

```

        <span v-else>No activo</span>
    </q-td>
</template>
<template v-slot:body-cell-options="props">
    <q-td :props="props">
        <button class="btn-options" style="background-color: #55B995;"
        @click="modalEditUser = true, editUser = props.row">
            <q-icon name="edit" />
        </button>
        <button class="btn-options" style="background-color: #F3920B;"
        @click="deleteUser(props.row)">
            <q-icon name="delete" />
        </button>
    </q-td>
</template>
</q-table>
</div>

```

```

<q-dialog v-model="modalUser" persistent>
    <q-card style="min-width: 600px;">
        <q-card-section class="row">
            <div class="col-12">
                <q-form
                    @submit="addUser"
                    class="q-gutter-md"
                >
                    <div class="row">
                        <div class="col-12" style="padding: 20px;">
                            <div class="form-group">
                                <label for="">Nombre</label>

```

```

        <br>

        <input v-model="newUser.name" type="text">
    </div>

    <div class="form-group">
        <label for="">Rol</label>

        <br>

        <select v-model="newUser.role">
            <option value="admin">Administrador</option>
            <option value="instructor">Instructor</option>
        </select>
    </div>

    <div class="form-group">
        <label for="">Email</label>

        <br>

        <input v-model="newUser.email" type="text">
    </div>

    <div class="form-group">
        <label for="">Contraseña</label>

        <br>

        <input v-model="newUser.password" type="password">
    </div>
</div>

<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalUser =
false">Cerrar</button>

    <button class="btn-submit" type="submit">Guardar</button>
</div>

```

```

        </q-form>
    </div>
</q-card-section>
</q-card>
</q-dialog>

<q-dialog v-model="modalEditUser" persistent v-if="editUser != null">
    <q-card style="min-width: 600px;">
        <q-card-section class="row">
            <div class="col-12">
                <q-form
                    @submit="addUserEdit"
                    class="q-gutter-md">

                    <div class="row">
                        <div class="col-12" style="padding: 20px;">
                            <div class="form-group">
                                <label for="">Nombre</label>
                                <br>
                                <input v-model="editUser.name" type="text">
                            </div>
                            <div class="form-group">
                                <label for="">Rol</label>
                                <br>
                                <select v-model="editUser.role">
                                    <option value="admin">Administrador</option>
                                    <option value="instructor">Instructor</option>
                                </select>
                            </div>
                        </div>
                    </div>
                </q-form>
            </div>
        </q-card-section>
    </q-card>
</q-dialog>

```

```

        <div class="form-group">
            <label for="">Email</label>
            <br>
            <input v-model="editUser.email" type="text">
        </div>
        <div class="form-group">
            <label for="">Contraseña</label>
            <br>
            <input v-model="editUser.password" type="password">
        </div>
    </div>
</div>

<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalEditUser =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>
</q-card>
</q-dialog>
</q-page>
</template>

<script>
import { db, storage } from 'boot/firebase'

const ref = storage.ref()

```



```
import { mapState, mapActions } from 'vuex'

import moment from 'moment-timezone'

import VueTagsInput from '@johmun/vue-tags-input';

export default {
  name: 'instructors',

  components: {
    VueTagsInput,
  },

  data(){
    return{
      newUser: {
        role: 'instructor',
        complete_profile: false,
        check_status: true,
        background_image: 'Azul.jpg',
        photo: 'https://firebasestorage.googleapis.com/v0/b/kumari-
d465a.appspot.com/o/user_image%2Fimagen_2022-03-
15_210459.png?alt=media&token=6d8c0cf3-7e89-4949-888b-35a8c8c294fe'
      },

      modalUser: false,
      modalEditUser: false,
      editUser: null,

      courseImage: null,

      users: [],
```

```

filter: "",
columns: [
  {
    name: 'name',
    required: true,
    label: 'Nombre',
    align: 'left',
    field: row => row.name,
    format: val => `${val}`,
    sortable: true
  },
  { name: 'email', align: 'left', label: 'Email', field: 'email', sortable: true },
  { name: 'education', align: 'left', label: 'Educacion', field: 'education', sortable: true },
  { name: 'courses', align: 'left', label: 'Cursos', field: 'courses' },
  { name: 'students', label: 'Estudiantes', field: 'students' },
  { name: 'status', label: 'Estatus', field: 'status' },
  { name: 'options', label: 'Opciones', field: 'options' }
],
}
},

```

```

filters: {
  formatDate(args){
    try {
      let date = moment(args.toDate()).fromNow();
      return date
    } catch (error) {
      let date = moment(args).fromNow();
      return date
    }
  }
}

```

```

    }
  }
},

created() {
  this.getUsers()
},

computed: {
  ...mapState('auth', [ 'user' ])
},

methods: {
  truncateString(str, num) {
    if (str.length <= num) {
      return str
    }

    return str.slice(0, num) + '...'
  },

  async addUser(){
    try {
      this.$q.dialog({
        title: 'Agregar nuevo instructor',
        message: '¿Quieres registrar este instructor?',
        cancel: true,
        persistent: true
      }).onOk(async () => {

```

```

        let response = await db.collection('temporary').add(this.newUser)
    })

    } catch (error) {
        console.log(error);
    }
},

```

```

async addUserEdit(){
    try {
        let response = await db.collection('users')
            .doc(this.editUser.uid)
            .update(this.editUser)

        this.$q.notify({
            message: 'Instructor editado',
            color: 'positive'
        })

        this.modalEditUser = false
    } catch (error) {
        console.log(error);
    }
},

```

```

async getUsers(){
    this.users = []

    try {

```

```

let response = await db.collection('users')
    .where('role', '==', 'instructor')
    .get()
    .then(query => {
        query.forEach(async doc => {
            let user = doc.data()
            user.id = doc.id

            user.courses = []

            let courses = await db.collection('courses')
                .where('instructor', '==', user.id)
                .get()
                .then(query => {
                    query.forEach(dat => {
                        let course = dat.data()
                        course.id = dat.id

                        user.courses.push(course)

                    })
                })

            this.users.push(user)
        })
    })
} catch (error) {
    console.log(error);
}

```

```
async deleteUser(user){
  try {
    this.$q.dialog({
      title: 'Eliminar instructor',
      message: '¿Estas seguro de eliminar este instructor?',
      cancel: true,
      persistent: true
    }).onOk(async () => {

    })

  } catch (error) {
    console.log(error);
  }
},
```

```
//Generar UUID  
generateUUID(){  
  
    var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];  
  
    var k = ['x','x','x','x','x','x','x','x','-','x','x','x','x','-','4','x','x','x','-','y','x','x','x','-'  
,'x','x','x','x','x','x','x','x','x','x','x','x'];  
  
    var u = "", i=0, rb=Math.random()*0xffffffff|0;  
  
    while(i++<36) {  
  
        var c=k[i-1], r=rb&0xf, v=c=='x'?r:(r&0x3|0x8);  
  
        u+=(c=='-' || c=='4')?c:h[v]; rb=i%8==0?Math.random()*0xffffffff|0:rb>>4  
    }  
  
    return u
```

```
},
```

```
async uploadImage(){
  // this.isLoading = true;
  try {
    let metadata = {
      contentType: "image/jpeg",
      contentLanguage: "es",
    }
    let fotold = this.generateUUID()
    const refImg = ref.child('courses/' + fotold + '.jpg')
    return refImg.put(this.courseImage, metadata)
      .then(e => {
        console.log(e)
        return ref.child('courses/' + fotold + '.jpg').getDownloadURL()
          .then(async (url) => {
            return url
          })
      })
    // this.getResizePath(fotold)
  })
  .catch(error => console.log(error))
} catch (error) {
  console.log(error);
}
},
```

```
getImage(e){
  this.courseImage = e.target.files[0]
```

```
        console.log(this.courseImage);
    },

    },
}
</script>
```

### Lessons.txt

```
<style scoped lang="scss">
```

```
.main-container{
    background-color: white;
    border-radius: 10px;
    padding: 20px;
}
```

```
.main-title{
    font-size: 22px;
    font-weight: bold;
}
```

```
.btn-options{
    border: none;
    margin: 5px;
    font-size: 18px;
    color: white;
    border-radius: 5px;
    cursor: pointer;
}
```

```
.btn-modal{
```



```
border: none;
background-color: #24B8CB;
color: white;
border-radius: 15px;
padding: 10px 15px;
font-weight: bold;
font-size: 17px;
}
```

```
.form-group{
  margin-top: 20px;

  label{
    font-weight: bold;
    font-size: 18px;
  }
}
```

```
.input-text{
  width: 100%;
  border: 2px solid #666666;
  border-radius: 10px;
  padding: 5px;
}
```

```
.input-text:focus{
  outline: none;
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;  
}
```

```
.layer{  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    top: 0;  
    left: 0;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;
```

```
background-color: rgba(0, 0, 0, 0.5);  
color: white;  
}  
</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<q-table
```

```
  :data="clases"
```

```
  :columns="columns"
```

```
  :filter="filter"
```

```
  row-key="name"
```

```
  flat
```

```
>
```

```
<template v-slot:top>
```

```
<div class="flex justify-between items-center" style="width: 100%;">
```

```
<span style="font-size: 25px; font-weight: bold;">Clases</span>
```

```
<button class="btn-modal cursor-pointer" @click="modalClase = true">Crear  
clase</button>
```

```
<q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"  
color="primary" v-model="filter">
```

```
<template v-slot:append>
```

```
<q-icon name="search" />
```

```
</template>
```

```
</q-input>
```

```
</div>
```

```
</template>
```

```
<template v-slot:body-cell-course="props">
```

```

    <q-td :props="props">
      {{ returnCourse(props.row.course) }}
    </q-td>
  </template>
  <template v-slot:body-cell-video="props">
    <q-td :props="props">
      <span class="cursor-pointer" @click="modalVideo = true, claseSelect =
props.row">Ver</span>
    </q-td>
  </template>
  <template v-slot:body-cell-material="props">
    <q-td :props="props">
      <a target="_blank" :href="props.row.material">Ver</a>
    </q-td>
  </template>
  <template v-slot:body-cell-details="props">
    <q-td :props="props">
      <a target="_blank" :href="props.row.details">Ver</a>
    </q-td>
  </template>
  <template v-slot:body-cell-active="props">
    <q-td :props="props">
      <span v-if="props.row.active">Activo</span>
      <span v-else>No activo</span>
    </q-td>
  </template>
  <template v-slot:body-cell-options="props">
    <q-td :props="props">
      <button class="btn-options" style="background-color: #2CB79F;"
@click="modalEditClase = true, editClase = props.row">

```

```

        <q-icon name="edit" />
      </button>

      <button class="btn-options" style="background-color: #DC3545;"
@click="deleteClase(props.row)">

        <q-icon name="delete" />
      </button>
    </q-td>
  </template>
</q-table>
</div>

```

```

<q-dialog v-model="modalClase" persistent>
  <q-card style="min-width: 800px; position:relative">

```

```

    <q-card-section class="row">
      <div class="col-12">
        <q-form
          @submit="addClase"
          class="q-gutter-md"
        >
          <div class="row">
            <div class="col-6 q-pa-md">
              <div class="form-group">
                <label for="">Clase</label> <br>
                <input class="input-text" type="text" v-model="newClase.name">
              </div>
              <div class="form-group">
                <label for="">Curso</label> <br>
                <select class="input-text" v-model="courseSelected">

```

```

        <option :value="course" v-for="(course, index) in courses" :key="index">{{
course.name }}</option>

    </select>

</div>

<div class="row form-group">

    <div class="col-6">

        <label for="">PDF Materiales</label>

        <input type="file" @change="getMaterial($event)"
accept="application/pdf">

        <br>

        <label for="">PDF Detalles</label>

        <input type="file" @change="getDetails($event)"
accept="application/pdf">

        <br>

        <label for="">Sticker</label>

        <input type="file" @change="getSticker($event)" accept="image/*">

    </div>

</div>

<div class="col-6 q-pa-md">

    <div class="row form-group">

        <div class="col-6">

            <label for="">Banner</label>

            <input type="file" @change="getImage($event)" accept="image/*">

            <br>

            <label for="">Video</label>

            <input ref="inputVideo" type="file" @change="getVideo($event)">

        </div>

    </div>

</div>

```

```

    <div class="form-group">
      <label for="">Estatus</label>
      <select class="input-text" v-model="newClase.status">
        <option value="Activo">Activo</option>
        <option value="Pausado">Pausado</option>
      </select>
    </div>
    <div class="form-group">
      <label for="">Orden</label> <br>
      <input class="input-text" type="text" v-model="newClase.order">
    </div>
  </div>
</div>
<div class="q-pa-md flex justify-end">
  <button class="btn-cancel" type="button" @click="modalClase =
false">Cerrar</button>
  <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>

<div v-if="uploading" class="layer">
  <q-spinner
    color="primary"
    size="3em"
  />
  <p>Creando registro...</p>
</div>

```

```
</q-card>
```

```
</q-dialog>
```

```
<q-dialog v-model="modalEditClase" persistent v-if="editClase != null">
```

```
<q-card style="min-width: 800px; position:relative">
```

```
<q-card-section class="row">
```

```
<div class="col-12">
```

```
<q-form
```

```
@submit="addClaseEdit"
```

```
class="q-gutter-md"
```

```
>
```

```
<div class="row">
```

```
<div class="col-6 q-pa-md">
```

```
<div class="form-group">
```

```
<label for="">Clase</label> <br>
```

```
<input class="input-text" type="text" v-model="editClase.name">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Curso</label> <br>
```

```
<select class="input-text" v-model="editClase.course">
```

```
<option :value="course.id" v-for="(course, index) in courses"
:key="index">{{ course.name }}</option>
```

```
</select>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Clase URL</label> <br>
```

```
<input class="input-text" type="text" v-model="editClase.url">
```

```
</div>
```



```

<div class="row form-group">
  <div class="col-6">
    <label for="">PDF Materiales</label>
    <input type="file" @change="getMaterial($event)"
accept="application/pdf">
    <br>
    <label for="">PDF Detalles</label>
    <input type="file" @change="getDetails($event)"
accept="application/pdf">
    <br>
    <label for="">Sticker</label>
    <input type="file" @change="getSticker($event)" accept="image/*">
  </div>
</div>
</div>
<div class="col-6 q-pa-md">
  <div class="row form-group">
    <div class="col-6">
      <label for="">Banner</label>
      <input type="file" @change="getImage($event)" accept="image/*">
      <br>
      <label for="">Video</label>
      <input ref="inputVideo" type="file" @change="getVideo($event)">
    </div>
  </div>
</div>

<div class="form-group">
  <label for="">Estatus</label>
  <select class="input-text" v-model="editClase.status">
    <option value="Activo">Activo</option>

```

```

        <option value="Pausado">Pausado</option>
    </select>
</div>
<div class="form-group">
    <label for="">Orden</label> <br>
    <input class="input-text" type="text" v-model="editClase.order">
</div>
</div>
</div>
<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalEditClase =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>

<div v-if="uploading" class="layer">
    <q-spinner
        color="primary"
        size="3em"
    />
    <p>Actualizando...</p>
</div>
</q-card>
</q-dialog>

<q-dialog v-model="modalVideo" persistent>

```

```

<q-card>
  <q-card-section>
    <div class="text-h6">{{ claseSelect.name }}</div>
  </q-card-section>

  <q-card-section class="q-pt-none">
    <video controls controlsList="nodownload" width="100%"
:src="claseSelect.url"></video>
  </q-card-section>

  <q-card-actions align="right">
    <q-btn flat label="Cerrar" color="primary" v-close-popup />
  </q-card-actions>
</q-card>
</q-dialog>

  <ProgressComponent :upload="upload" :progress="progress" :progress-label="progressLabel"
:file="claseVideo" />
</q-page>
</template>

<script>
//Firebase
import { firebase, db, storage } from 'boot/firebase'
const ref = storage.ref()
import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import VueTagsInput from '@johmun/vue-tags-input';
import ProgressComponent from '../instructor/ProgressComponent.vue'

```

```
export default {
  name: 'feed',

  components: {
    VueTagsInput,
    ProgressComponent
  },

  data(){
    return{
      uploading: false,

      newClase: {
        name: "",
        photo: "",
        status: null,
        course: "",
        video: "",
        order: 1,
        url: "",
      },

      courses: [],

      modalVideo: false,
      modalClase: false,
      modalEditClase: false,
      editClase: null,
```

```
classImage: null,
classVideo: null,
classMaterial: null,
classDetails: null,
classSticker: null,

courseSelected: null,

classSelect: {},

classes: [],
filter: "",
columns: [
    {
        name: 'name',
        required: true,
        label: 'Nombre',
        align: 'left',
        field: row => row.name,
        format: val => `${val}`,
        sortable: true
    },
    { name: 'course', align: 'left', label: 'Curso', field: 'course', sortable: true },
    { name: 'video', align: 'left', label: 'Video', field: 'video', sortable: true },
    { name: 'material', align: 'left', label: 'Material', field: 'material', sortable: true },
    { name: 'details', align: 'left', label: 'Detalles', field: 'details', sortable: true },
    { name: 'status', align: 'left', label: 'Estatus', field: 'status' },
    { name: 'options', label: 'Opciones', field: 'options' }
],
```

```
    upload: false,  
    progress: null,  
    progressLabel: "",  
  }  
},
```

```
filters: {  
  formatDate(args){  
    try {  
      let date = moment(args.toDate()).fromNow();  
      return date  
    } catch (error) {  
      let date = moment(args).fromNow();  
      return date  
    }  
  }  
},
```

```
created() {  
  this.getClases()  
  this.getCourses()  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ])  
},
```

```
methods: {
```

```
truncateString(str, num) {  
  if (str.length <= num) {  
    return str  
  }  
  
  return str.slice(0, num) + '...'  
},
```

```
async addClase(){  
  try {  
    this.$q.dialog({  
      title: 'Agregar nuevo clase',  
      message: '¿Quieres registrar este clase?',  
      cancel: true,  
      persistent: true  
    }).onOk(async () => {  
      this.uploading = true  
  
      let url = null  
      let materialUrl = null  
      let detailsUrl = null  
      let stickerUrl = null  
  
      if (this.claseImage == null) {  
        url = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-jihan-cover-680x384.jpg'  
      }else{  
        url = await this.uploadImage(this.claseImage)  
        this.claseImage = null  
      }  
    })  
  }  
}
```

```
}
```

```
if (this.claseMaterial == null) {
```

```
    materialUrl = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-  
jihan-cover-680x384.jpg'
```

```
}else{
```

```
    materialUrl = await this.uploadPDF(this.claseMaterial)
```

```
    this.claseMaterial = null
```

```
}
```

```
if (this.claseDetails == null) {
```

```
    detailsUrl = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-  
jihan-cover-680x384.jpg'
```

```
}else{
```

```
    detailsUrl = await this.uploadPDF(this.claseDetails)
```

```
    this.claseDetails = null
```

```
}
```

```
if (this.claseSticker == null) {
```

```
    stickerUrl =  
'http://pm1.narvii.com/7584/8ef97fd2da9ef17bc680cb3691cc767d079873f3r1-574-  
569v2_uhq.jpg'
```

```
}else{
```

```
    stickerUrl = await this.uploadImage(this.claseSticker)
```

```
    this.claseSticker = null
```

```
}
```

```
this.newClase.course = this.courseSelected.id
```

```
this.newClase.instructor = this.courseSelected.instructor
```



```

this.newClase.photo = url

this.newClase.material = materialUrl

this.newClase.details = detailsUrl

this.newClase.sticker = stickerUrl

if (this.newClase.url == '') {

    this.newClase.url = 'https://firebasestorage.googleapis.com/v0/b/kumari-
d465a.appspot.com/o/videos%2FSigrid%20-
%20Burning%20Bridges.mp4?alt=media&token=20dabddb-3494-403b-8040-99d210bd5052'

}

this.newClase.created_at = moment().format()

this.newClase.created_atUnix = moment(this.newClase.created_at).unix()

let reponse = await db.collection('clases')
    .add(this.newClase)
    .then(query => {
        this.newClase.id = query.id
        let course = JSON.parse(JSON.stringify(this.newClase))

        this.clases.push(course)

        this.claseImage = null

        this.modalClase = false

        this.newClase = {
            name: "",
            photo: "",
            status: null,

```

```

        course: "",
    }

    this.uploading = false
  })
  this.$refs.inputVideo.value = ""
  this.$q.notify({
    message: 'Clase registrado',
    color: 'positive'
  })
})

} catch (error) {
  console.log(error);
  this.uploading = false
}
},

returnCourse(id){
  let found = this.courses.find(doc => doc.id == id)

  if (found) {
    return found.name
  }

},

async addClaseEdit(){
  try {

```

```
this.uploading = true
```

```
if (this.clasImage == null) {  
    this.editClase.photo = this.editClase.photo  
}else{  
    let url = await this.uploadImage(this.clasImage)  
    this.clasImage = null  
    this.editClase.photo = url  
}
```

```
if (this.claseMaterial == null) {  
    this.editClase.material = this.editClase.material  
}else{  
    let materialUrl = await this.uploadPDF(this.claseMaterial)  
    this.claseMaterial == null  
    this.editClase.material = materialUrl  
}
```

```
if (this.claseDetails == null) {  
    this.editClase.details = this.editClase.details  
}else{  
    let detailsURL = await this.uploadPDF(this.claseDetails)  
    this.claseDetails == null  
    this.editClase.details = detailsURL  
}
```

```
if (this.claseSticker == null) {  
    this.editClase.sticker = this.editClase.sticker  
}else{
```

```

        let stickerUrl = await this.uploadImage(this.claseSticker)

        this.claseSticker == null

        this.editClase.sticker = stickerUrl
    }

    let response = await db.collection('clases')
        .doc(this.editClase.id)
        .update(this.editClase)

    this.modalEditClase = false

    this.$q.notify({
        message: 'Clase editado',
        color: 'positive'
    })

    this.uploading = false
} catch (error) {
    console.log(error);
    this.uploading = false
}
},

async getCourses(){
    this.courses = []

    try {
        let response = await db.collection('courses')
            .get()
    }
}

```

```

        .then(query => {
            query.forEach(async doc => {
                let course = doc.data()
                course.id = doc.id

                this.courses.push(course)
            })
        })
    } catch (error) {
        console.log(error);
    }
},

```

```

async getClases(){
    this.clases = []

    try {
        let response = await db.collection('clases')
            .get()
            .then(query => {
                query.forEach(async doc => {
                    let clase = doc.data()
                    clase.id = doc.id

                    this.clases.push(clase)
                })
            })
    } catch (error) {
        console.log(error);
    }
}

```

```

    }
  },

  async deleteClase(clase){
    try {
      this.$q.dialog({
        title: 'Eliminar clase',
        message: '¿Estas seguro de eliminar este clase?',
        cancel: true,
        persistent: true
      }).onOk(async () => {
        let index = this.clases.findIndex(doc => doc.id == clase.id)
        let response = await db.collection('clases').doc(clase.id).delete()

        this.$q.notify({
          message: 'Clase eliminado',
          color: 'positive'
        })

        this.clases.splice(index, 1)
      })

    } catch (error) {
      console.log(error);
    }
  },

  //Generar UUID
  generateUUID(){

```

```

var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];

var k = ['x','x','x','x','x','x','x','x','-','x','x','x','x','-','4','x','x','x','x','-','y','x','x','x','x','-','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];

var u = "",i=0,rb=Math.random()*0xffffffff|0;

while(i++<36) {

    var c=k[i-1],r=rb&0xf,v=c=='x'?r:(r&0x3|0x8);

    u+=(c=='-' || c=='4')?c:h[v];rb=i%8==0?Math.random()*0xffffffff|0:rb>>4

}

return u

},

```

```

async uploadVideo(file){

    this.upload = true

    let ext = file.name.substring(file.name.lastIndexOf('.')+1, file.name.length) || file.name

    try {

        let metadata = {

            contentType: file.type,

            contentLanguage: "es",

        }

        let fileId = this.generateUUID()

        const refFile = ref.child('vid/' + fileId + '.' + ext)

        var task = refFile.put(file, metadata);

        task.on('state_changed',

            (snapshot) => {

                // Observe state change events such as progress, pause, and resume

                // Get task progress, including the number of bytes uploaded and the total number
of bytes to be uploaded

                this.progress = ((snapshot.bytesTransferred / snapshot.totalBytes) * 100) / 100;

                this.progressLabel = Math.trunc((this.progress * 100)) + '%'

                console.log('Upload is ' + (this.progress * 100) + '% done');

```

```

switch (snapshot.state) {
case firebase.storage.TaskState.PAUSED: // or 'paused'
    console.log('Upload is paused');
    break;
case firebase.storage.TaskState.RUNNING: // or 'running'
    console.log('Upload is running');
    break;
}
},
(error) => {
    // Handle unsuccessful uploads
},
() => {

    // Handle successful uploads on complete
    // For instance, get the download URL: https://firebasestorage.googleapis.com/...
    task.snapshot.ref.getDownloadURL().then(async (downloadURL) => {
        console.log('File available at', downloadURL);

        this.newClase.url = downloadURL

        if (this.editClase != null) {
            this.editClase.url = downloadURL
        }
        this.upload = false
    });
}
);

```



```
    } catch (error) {  
      console.log(error);  
    }  
  },
```

```
  async uploadImage(image){
```

```
    try {  
      let metadata = {  
        contentType: "image/jpeg",  
        contentLanguage: "es",  
      }  
  
      let fotold = this.generateUUID()  
  
      const refImg = ref.child('clases/' + fotold + '.jpg')  
  
      return refImg.put(image, metadata)  
  
      .then(e => {  
        console.log(e)  
  
        return ref.child('clases/' + fotold + '.jpg').getDownloadURL()  
  
        .then(async (url) => {  
          return url  
  
        })  
  
      })  
  
    })  
  
    .catch(error => console.log(error))  
  } catch (error) {  
    console.log(error);  
  }  
},
```

```

async uploadPDF(data){

  try {

    let metadata = {

      contentType: "application/pdf",

      contentLanguage: "es",

    }

    let pdfId = this.generateUUID()

    const refImg = ref.child('PDFs/' + pdfId + '.pdf')

    return refImg.put(data, metadata)

    .then(e => {

      console.log(e)

      return ref.child('PDFs/' + pdfId + '.pdf').getDownloadURL()

      .then(async (url) => {

        return url

      })

    })

    .catch(error => console.log(error))

  } catch (error) {

    console.log(error);

  }

},

```

```

getMaterial(e){

  this.claseMaterial = e.target.files[0]

  console.log(this.claseMaterial);

```

```

    },

    getDetails(e){
        this.claseDetails = e.target.files[0]
        console.log(this.claseDetails);
    },

    getSticker(e){
        this.claseSticker = e.target.files[0]
        console.log(this.claseSticker);
    },

    getImage(e){
        this.claseImage = e.target.files[0]
        console.log(this.claseImage);
    },

    getVideo(e){
        this.claseVideo = e.target.files[0]
        console.log(this.claseVideo);

        this.uploadVideo(this.claseVideo)
    },

    },
}
</script>
Orders.txt
<style scoped lang="scss">

```

```
.main-container{  
    background-color: white;  
    border-radius: 10px;  
    padding: 20px;  
}
```

```
.main-title{  
    font-size: 22px;  
    font-weight: bold;  
}
```

```
.btn-options{  
    border: none;  
    margin: 5px;  
    font-size: 18px;  
    color: white;  
    border-radius: 5px;  
    cursor: pointer;  
}
```

```
.btn-modal{  
    border: none;  
    background-color: #24B8CB;  
    color: white;  
    border-radius: 15px;  
    padding: 10px 15px;  
    font-weight: bold;  
    font-size: 17px;  
}
```

```
.form-group{  
  margin-top: 20px;  
  
  label{  
    font-weight: bold;  
    font-size: 18px;  
  }  
}
```

```
.input-text{  
  width: 100%;  
  border: 2px solid #666666;  
  border-radius: 10px;  
  padding: 5px;  
}
```

```
.input-text:focus{  
  outline: none;  
}
```

```
.btn-cancel{  
  color: white;  
  padding: 5px 20px;  
  border: none;  
  background-color: #E72B81;  
  border-radius: 10px;  
  cursor: pointer;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;  
}
```

```
.badge-color{  
    width: 20px;  
    height: 20px;  
    border-radius: 50%;  
    margin-right: 10px;  
    cursor: pointer;  
  
}
```

```
.active{  
    box-shadow: 1px 1px 10px rgba(0, 0, 0, 0.8);  
}
```

```
.form-group{  
    padding: 10px 5px;  
}
```

```
label{
```

```
font-size: 18px;
```

```
input{  
    margin-right: 10px;  
}  
}
```

```
input[type=text]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
input[type=date]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
input[type=password]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;
```

```
}
```

```
select{  
  width: 100%;  
  border-radius: 10px;  
  border: none;  
  border: 1px solid black;  
  padding: 5px;  
}
```

```
.rate-card{  
  padding: 5px;  
  border: 1px solid gray;  
  margin-bottom: 10px;  
  line-height: 10px;
```

```
p:nth-child(1){  
  font-size: 17px;  
  font-weight: bold;  
}
```

```
p:nth-child(3){  
  font-size: 15px;  
  font-weight: 400;  
}
```

```
}
```

```
.layer-buy{  
  position: absolute;
```



```
top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.8);
```

```
display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

color: white;
```

```
img{

  width: 30%;

}

}
```

```
</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<q-table
```

```
  title="Pedidos"
```

```
  :data="orders"
```

```
  :columns="columns"
```

```
  :filter="filter"
```

```
  row-key="name"
```

```
  flat
```

>

```
<template v-slot:top>

  <div class="flex justify-between items-center" style="width: 100%;">

    <span style="font-size: 25px; font-weight: bold;">Pedidos</span>

    <button class="btn-modal cursor-pointer" @click="modalOrder = true">Nuevo
pedido</button>

    <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">

      <template v-slot:append>

        <q-icon name="search" />

      </template>

    </q-input>

  </div>

</template>

<template v-slot:body-cell-name="props">

  <q-td :props="props">

    {{ props.row.user.name }}

  </q-td>

</template>

<template v-slot:body-cell-email="props">

  <q-td :props="props">

    {{ props.row.user.email }}

  </q-td>

</template>

<template v-slot:body-cell-direction="props">

  <q-td :props="props">

    {{ props.row.user.direction }}

  </q-td>

</template>
```

```

<template v-slot:body-cell-city="props">
  <q-td :props="props">
    {{ props.row.user.city }}
  </q-td>
</template>

<template v-slot:body-cell-date="props">
  <q-td :props="props">
    {{ props.row.date | formatDate }}
  </q-td>
</template>

<template v-slot:body-cell-course="props">
  <q-td :props="props">
    {{ props.row.course }}
  </q-td>
</template>

<template v-slot:body-cell-status="props">
  <q-td :props="props">
    <span v-if="props.row.status">Entregado</span>
    <span v-else>Pendiente</span>
  </q-td>
</template>

<template v-slot:body-cell-options="props">
  <q-td :props="props">
    <div v-if="props.row.external">
      <button class="btn-options" style="background-color: #A5A5A5;" disabled>
        <q-icon name="print" />
      </button>
      <button class="btn-options" style="background-color: #55B995;"
        @click="modalEditOrder = true, editOrder = JSON.parse(JSON.stringify(props.row))">

```

```

        <q-icon name="edit" />
    </button>

    <button class="btn-options" style="background-color: #F3920B;"
@click="deleteOrder(props.row)">

        <q-icon name="delete" />
    </button>

    <button class="btn-options" style="background-color: #A5A5A5;" disabled>

        <q-icon name="local_shipping" />
    </button>
</div>

<div v-else>

    <button v-if="props.row.generate_shipment" class="btn-options"
style="background-color: #2C5CB4;" @click="getGuide(props.row)">

        <q-icon name="print" />
    </button>

    <button v-else class="btn-options" style="background-color: #A5A5A5;" disabled>

        <q-icon name="print" />
    </button>

    <button class="btn-options" style="background-color: #55B995;"
@click="modalEditOrder = true, editOrder = JSON.parse(JSON.stringify(props.row))">

        <q-icon name="edit" />
    </button>

    <button class="btn-options" style="background-color: #F3920B;"
@click="deleteOrder(props.row)">

        <q-icon name="delete" />
    </button>

    <button v-if="props.row.generate_shipment" class="btn-options"
style="background-color: #A5A5A5;" disabled>

        <q-icon name="local_shipping" />
    </button>

```

```
        <button v-else class="btn-options" style="background-color: #972121;"
@click="generateShipment(props.row)">
```

```
            <q-icon name="local_shipping" />
```

```
        </button>
```

```
    </div>
```

```
</q-td>
```

```
</template>
```

```
</q-table>
```

```
</div>
```

```
<q-dialog v-model="modalRates" position="right" persistent v-if="shipment != null">
```

```
  <q-card style="width: 350px; max-height: 80vh;">
```

```
    <q-card-section class="">
```

```
      <div class="rate-card" v-for="(rate, index) in shipment.included" :key="index" v-
if="rate.type == 'rates'">
```

```
        <p>{{ rate.attributes.provider }}</p>
```

```
        <p>${{{ rate.attributes.total_pricing }}}</p>
```

```
        <p>{{ rate.attributes.service_level_name }}</p>
```

```
      <button class="btn-modal"
```

```
@click="createLabelSkydrox(rate.id)">Seleccionar</button>
```

```
    </div>
```

```
  </q-card-section>
```

```
</q-card>
```

```
</q-dialog>
```

```
<q-dialog v-model="modalOrder" persistent>
```

```
  <q-card style="min-width: 500px; border-radius: 50px;position: relative;">
```

```
    <q-card-section class="row">
```

```
      <div class="col-12">
```

```

<q-form
  @submit="addOrder"
  class="q-gutter-md"
>
  <div class="row">
    <div class="col-12">
      <div class="form-group">
        <label for="">Usuario</label>
        <br>
        <q-select
          v-model="newOrder.user"
          use-input
          input-debounce="0"
          :options="optionsUsers"
          @filter="filterFnUser"
          behavior="dialog"
          option-label="name"
          option-value="uid"
          emit-value
          map-options
        >
          <template v-slot:no-option>
            <q-item>
              <q-item-section class="text-grey">
                Sin resultados
              </q-item-section>
            </q-item>
          </template>
        </q-select>

```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="">Curso</label>
```

```
  <br>
```

```
  <q-select
```

```
    v-model="newOrder.course"
```

```
    use-input
```

```
    input-debounce="0"
```

```
    :options="optionsCourses"
```

```
    @filter="filterFnCourse"
```

```
    behavior="dialog"
```

```
    option-label="name"
```

```
    option-value="name"
```

```
    emit-value
```

```
    map-options
```

```
  >
```

```
    <template v-slot:no-option>
```

```
      <q-item>
```

```
        <q-item-section class="text-grey">
```

```
          Sin resultados
```

```
        </q-item-section>
```

```
      </q-item>
```

```
    </template>
```

```
  </q-select>
```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="">Fecha de pago</label>
```

```
  <br>
```

```
  <q-input filled v-model="newOrder.date" mask="date" :rules="['date']">
```

```

        <template v-slot:append>
            <q-icon name="event" class="cursor-pointer">
                <q-popup-proxy ref="qDateProxy" cover transition-show="scale"
transition-hide="scale">
                    <q-date v-model="newOrder.date">
                        <div class="row items-center justify-end">
                            <q-btn v-close-popup label="Close" color="primary" flat />
                        </div>
                    </q-date>
                </q-popup-proxy>
            </q-icon>
        </template>
    </q-input>
</div>
<div class="form-group">
    <label for="">Estatus</label>
    <br>
    <q-radio v-model="newOrder.status" :val="true" label="Entregado" />
    <br>
    <q-radio v-model="newOrder.status" :val="false" label="Pendiente" />
</div>
<div class="form-group">
    <label for="">Envío por fuera de Skydropx</label>
    <br>
    <q-checkbox right-label v-model="generateOrder" label="Si" />
</div>
</div>
</div>
<div class="q-pa-md flex justify-end">

```



```
        <button class="btn-cancel" type="button" @click="modalOrder =
false">Cerrar</button>
```

```
        <button class="btn-submit" type="submit">Guardar</button>
```

```
    </div>
```

```
  </q-form>
```

```
</div>
```

```
</q-card-section>
```

```
<div v-if="isLoading" class="layer-buy">
```

```
  
```

```
  <p>Espere un momento...</p>
```

```
</div>
```

```
</q-card>
```

```
</q-dialog>
```

```
<q-dialog v-model="modalEditOrder" persistent v-if="editOrder != null">
```

```
  <q-card style="min-width: 500px; border-radius: 50px;">
```

```
    <q-card-section class="row">
```

```
      <div class="col-12">
```

```
        <q-form
```

```
          @submit="addOrderEdit"
```

```
          class="q-gutter-md">
```

```
            <div class="row">
```

```
              <div class="col-12">
```

```
                <div class="form-group">
```

```
                  <label for="">Usuario</label>
```

```
                  <br>
```

```

<q-select
  v-model="editOrder.user"
  use-input
  input-debounce="0"
  :options="optionsUsers"
  @filter="filterFnUser"
  behavior="dialog"
  option-label="name"
  option-value="uid"
  emit-value
  map-options
  readonly
>
  <template v-slot:no-option>
    <q-item>
      <q-item-section class="text-grey">
        Sin resultados
      </q-item-section>
    </q-item>
  </template>
</q-select>
</div>
<div class="form-group">
  <label for="">Curso</label>
  <br>
  <q-select
    v-model="editOrder.course"
    use-input
    input-debounce="0"

```

```

:options="optionsCourses"
@filter="filterFnCourse"
behavior="dialog"
option-label="name"
option-value="name"
emit-value
map-options
readonly
>
<template v-slot:no-option>
<q-item>
  <q-item-section class="text-grey">
    Sin resultados
  </q-item-section>
</q-item>
</template>
</q-select>
</div>
<div class="form-group">
  <label for="">Fecha de pago</label>
  <br>
  <q-input filled v-model="editOrder.date" mask="date" :rules="['date']"
readonly>
    <template v-slot:append>
      <q-icon name="event" class="cursor-pointer">
      <q-popup-proxy ref="qDateProxy" cover transition-show="scale"
transition-hide="scale">
        <q-date v-model="editOrder.date" readonly>
        <div class="row items-center justify-end">
          <q-btn v-close-popup label="Close" color="primary" flat />

```

```

        </div>

        </q-date>

    </q-popup-proxy>

    </q-icon>

</template>

</q-input>
</div>

<div class="form-group">
    <label for="">Estatus</label>

    <br>

    <q-radio v-model="editOrder.status" :val="true" label="Entregado" />

    <br>

    <q-radio v-model="editOrder.status" :val="false" label="Pendiente" />

</div>

</div>

</div>

<div class="q-pa-md flex justify-end">

    <button class="btn-cancel" type="button" @click="modalEditOrder =
false">Cerrar</button>

    <button class="btn-submit" type="submit">Guardar</button>

</div>

</q-form>

</div>

</q-card-section>

</q-card>

</q-dialog>

</q-page>

</template>

```

```
<script>

import { db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'

import moment from 'moment-timezone'

import VueTagsInput from '@johmun/vue-tags-input';

import { getCourses } from 'src/store/auth/actions'

import axios from 'axios'

export default {
  name: 'feed',

  components: {
    VueTagsInput,
  },

  data(){
    return{
      newOrder: {},

      modalRates: false,

      modalOrder: false,
      modalEditOrder: false,
      editOrder: null,

      shipment: null,
      label: null,
```

users: [],

optionsUsers: null,

courses: [],

optionsCourses: null,

orders: [],

filter: "",

columns: [

{

name: 'name',

required: true,

label: 'Nombre',

align: 'left',

field: row => row.name,

format: val => `\${val}`,

sortable: true

},

{ name: 'email', align: 'left', label: 'Email', field: 'email', sortable: true },

{ name: 'direction', align: 'left', label: 'Direccion', field: 'direction', sortable: true },

{ name: 'city', align: 'left', label: 'Ciudad', field: 'city', sortable: true },

{ name: 'date', align: 'left', label: 'Fecha de Pedido', field: 'date' },

{ name: 'course', label: 'Curso', field: 'course' },

{ name: 'status', label: 'Estatus', field: 'status' },

{ name: 'options', label: 'Opciones', field: 'options' }

],

isLoading: false,

isNew: false,

```
    allCourses: null,  
    generateOrder: false,
```

```
    selectedUser: null,  
  }  
},
```

```
filters: {  
  formatDate(args){  
    return args  
  }  
},
```

```
created() {  
  this.getOrders()  
  this.getUsers()  
  this.getCourses()  
  this.getAllCourses()
```

```
  //Skydropx  
  this.getOrdersSky()  
},
```

```
computed: {  
  ...mapState('auth', [ 'user']),
```

```
  allCoursesFilter(){  
    return this.courses.filter(doc => this.allCourses.courses.includes(doc.id))
```

```
},  
},
```

```
methods: {  
  async getAllCourses(){  
  
    try {  
      let response = await db.collection('all-courses')  
        .doc('course')  
        .get()  
  
      if (response.exists) {  
        this.allCourses = response.data()  
  
        this.courses.push(this.allCourses)  
      }  
    } catch (error) {  
      console.log(error);  
    }  
  },
```

```
//Skydropx  
getGuide(data){  
  console.log(data);  
  const options = {  
    method: 'GET',  
    url: 'https://api.skydropx.com/v1/labels/' + data.id_label,  
    headers: {
```



```
        cookie:
'_session_id=L2FmRSt1ZnAvWXYvQkY5cjBjV1AyNG9zbkptNUIzUkxGNFBwb0gvNUIBVkMzZEIRdGgv
emlpVzJ2NTJVOdhTN1I4NnUrV3AxeUJ0dG95Rm5lb1YxSVE9PS0tT1E5N2VQZEIvQmRPSWJ1dUptR1
RUUT09--62f681d59b964c054af17ef3153e74714501eebe',
```

```
        Authorization: 'Token token=GPhwp2G1TvttooGTQzrfxDYwlaCRB4ZxrpUyWJS5bg1Qt',
        'Content-Type': 'application/json'
    }
};
```

```
axios.request(options).then((response) => {
```

```
    window.open(response.data.data.attributes.label_url, "_blank");
}).catch(function (error) {
    console.error(error);
});
},
```

```
getOrdersSky(){
    try {
        const options = {
            method: 'GET',
            url: 'https://api.skydropx.com/v1/shipments',
            headers: {
                cookie:
'_session_id=ZDFpVXh1SnBXVG8xNEJ3WW4vbXIES0VVN0RWW05uV0lVQ05lOWZ6bzhuUnpXVEdp
UmlkZVRyUE8wT1FqcmpxeU9TV3J1YmInQlZBejVoelNpZ0NZQnc9PS0tc0dOcDNOWXZCeGpQVkJ6Z
nhYchlQT09--fb483433eef563d61a546ac8f8200869961853da',
```

```
                Authorization: 'Token token=GPhwp2G1TvttooGTQzrfxDYwlaCRB4ZxrpUyWJS5bg1Qt',
                'Content-Type': 'application/json'
            }
        };
    }
```

```

    axios.request(options).then(function (response) {
      console.log(response.data.data);
    }).catch(function (error) {
      console.error(error);
    });
  } catch (error) {
    console.log(error);
  }
},

```

```

//End Skydropx

```

```

truncateString(str, num) {
  if (str.length <= num) {
    return str
  }

  return str.slice(0, num) + '...'
},

```

```

filterFnUser (val, update) {
  if (val === '') {
    update(() => {
      this.optionsUsers = this.users.filter(user => user.role == 'user')
    })
    return
  }
}

```

```

update(() => {

```

```

        const needle = val.toLowerCase()

        this.optionsUsers = this.users.filter(v => v.name.toLowerCase().indexOf(needle) > -1)
    })
},

```

```

filterFnCourse (val, update) {
    if (val === '') {
        update(() => {
            this.optionsCourses = this.courses.filter(course => course.active == true)
        })
        return
    }
}

```

```

        update(() => {
            const needle = val.toLowerCase()

            this.optionsCourses = this.courses.filter(v => v.name.toLowerCase().indexOf(needle) > -1)
        })
    },

```

```

async getUsers(){
    try {
        let response = await db.collection('users')

            .get()
            .then(query => {
                query.forEach(doc => {
                    let user = doc.data()
                    user.id = doc.id

                    this.users.push(user)
                })
            })
    } catch (err) {
        console.log(err)
    }
}

```

```

        })
    })
} catch (error) {
    console.log(error);
}
},

```

```

async getCourses(){
    try {
        let response = await db.collection('courses')
            .get()
            .then(query => {
                query.forEach(doc => {
                    let course = doc.data()
                    course.id = doc.id

                    this.courses.push(course)
                })
            })
    }
}

```

```

} catch (error) {
    console.log(error);
}
},

```

```

async addOrder(){
    try {
        this.$q.dialog({

```

```

        title: 'Agregar nuevo orden',
        message: '¿Quieres registrar este orden?',
        cancel: true,
        persistent: true
    }).onOk(async () => {
        this.isNew = true
        this.isLoading = true

        if (this.generateOrder) {
            this.makeOrder()
            return
        }
        this.registerSkydropx()

    })

} catch (error) {
    console.log(error);
}
},

async makeOrder(){

    try {
        if (!this.generateOrder) {
            this.newOrder.id_skydropx = this.shipment.data.id
            this.newOrder.id_label = this.label.data.id
            this.newOrder.generate_shipment = true
        }
    }

```

```

    this.newOrder.external = true

    let response = await db.collection('orders').add(this.newOrder)

    this.modalOrder = false

    this.getOrders()

    this.submitPayment()
  } catch (error) {
    console.log(error);
    this.isLoading = false

    this.$q.notify({
      message: 'Ha ocurrido un error',
      color: 'negative'
    })
  }
},

registerSkydropx(data = null){
  let user = {}
  console.log(this.isNew);
  if (this.isNew) {
    user = this.users.find(doc => doc.uid == this.newOrder.user)
  }else{
    user = this.users.find(doc => doc.uid == data.user.uid)
    this.$q.loading.show()
  }
  this.selectedUser = user

```

```
try {  
  const options = {  
    method: 'POST',  
    url: 'https://api.skydropx.com/v1/shipments',  
    headers: {  
      cookie:  
'_session_id=aVYyalB4VUhnN29taWRHSnVDbElTY0FLNUN4THdOeE4yOW8vRTI5ZXNDamNIWIptL0  
JsaG1uUm9Mb0Y1Y0lTOEpmam5saDFVWGIb2lMYkk3RIZURmc9PS0tbjJRRUY1WUZKS3NmUVVuT  
3RSdURHdz09--81173d5e35ec68da2c55e95703149a7e1fe5acd3',  
      'Content-Type': 'application/json',  
      Authorization: 'Token token=GPhwp2G1TvttooGTQzrfxDYwlaCRB4ZxrpUyWJS5bg1Qt'  
    },  
    data: {  
      address_from: {  
        province: 'Chihuahua',  
        city: 'Chihuahua',  
        name: 'Paola Quezada',  
        zip: '31205',  
        country: 'MX',  
        address1: 'Blvd. Ortiz Mena 1834-C, 31205 Chihuahua, Chih.',  
        company: 'Kumari',  
        address2: 'Centro',  
        phone: '6144632764',  
        email: 'admin@kumari.mx'  
      },  
      parcels: [  
        {  
          weight: 3,  
          distance_unit: 'CM',  
          mass_unit: 'KG',
```

```

        height: 12,
        width: 35,
        length: 35
    }
],
    address_to: {
        province: user.state,
        city: user.city,
        name: user.name,
        zip: user.cp,
        country: 'MX',
        address1: user.direction,
        company: '-',
        address2: '-',
        phone: '6144632764',
        email: user.email,
        reference: 'Kumari',
        contents: 'Caja de actividades Kumari'
    },
    consignment_note_class_code: '53131600',
    consignment_note_packaging_code: '1H1',
    // consignment_note_class_code: '11646',
    // consignment_note_packaging_code: '4G',
}
};

```

```

axios.request(options).then((response) => {
    this.$q.loading.hide()
    console.log(response.data);

```



```
this.shipment = response.data
```

```
this.modalRates = true
```

```
}).catch((error) => {
```

```
  console.error(error);
```

```
  this.$q.loading.hide()
```

```
  this.isLoading = false
```

```
  this.$q.notify({
```

```
    message: 'Ha ocurrido un error',
```

```
    color: 'negative'
```

```
  })
```

```
});
```

```
} catch (error) {
```

```
  console.log(error);
```

```
  this.$q.loading.hide()
```

```
}
```

```
},
```

```
createLabelSkydrox(id){
```

```
  try {
```

```
    const options = {
```

```
      method: 'POST',
```

```
      url: 'https://api.skydropx.com/v1/labels',
```

```
      headers: {
```

```
        cookie:
```

```
'_session_id=aVYyaIB4VUhnN29taWRHSnVDbElTY0FLNUN4THdOeE4yOW8vRTI5ZXNDamNIWlptL0JsaG1uUm9Mb0Y1Y0lTOEpmam5saDFVWGIBb2lMYkk3RlZURmc9PS0tbjJRRUY1WUZKS3NmUVVuT3RSdURHdz09--81173d5e35ec68da2c55e95703149a7e1fe5acd3',
```

```
    'Content-Type': 'application/json',  
    Authorization: 'Token token=GPhwp2G1TvttooGTQzrfxDYwlaCRB4ZxrpUyWJS5bg1Qt'  
  },  
  data: {rate_id: parseInt(id), label_format: 'pdf'}  
};
```

```
axios.request(options).then((response) => {  
  console.log('Esta es la respuesta: ', response.data.data.attributes.status);
```

```
  
  if (response.data.data.attributes.status == 'ERROR') {  
    this.$q.notify({  
      message: 'Esta paqueteria no esta disponible',  
      color: 'negative'  
    })  
    this.modalRates = false  
    this.isLoading = false  
    this.$q.loading.hide()  
    return  
  }  
}
```

```
  this.label = response.data
```

```
  this.modalRates = false
```

```
  if (this.isNew) {  
    this.makeOrder()  
    this.getOrdersSky()
```

```
  return
```

```

    }

    this.editOrder.id_skydropx = this.shipment.data.id
    this.editOrder.id_label = this.label.data.id
    this.editOrder.tracking_number = this.label.data.attributes.tracking_number
    this.editOrder.tracking_url_provider = this.label.data.attributes.tracking_url_provider
    this.editOrder.generate_shipment = true
    this.$q.loading.hide()

    //Create mail
    this.sendEmail(this.editOrder.tracking_number, this.editOrder.tracking_url_provider)
    this.addOrderEdit()

  }).catch((error) => {
    console.error(error);
    this.isLoading = false
    this.$q.loading.hide()
    this.modalRates = false
    this.$q.notify({
      message: 'Esta paqueteria no esta disponible',
      color: 'negative'
    })
  });
} catch (error) {
  console.log(error);
  this.$q.loading.hide()
}
},

```

```

async submitPayment(){
  try {
    let user = this.users.find(doc => doc.uid == this.newOrder.user)

    let course = this.courses.find(doc => doc.name == this.newOrder.course)

    let date1 = moment(new Date()).unix();
    let date2 = moment(new Date()).add(6, 'months').unix();

    let mail = {
      to: user.email,
      subject: 'Compra KUMARI',
      message: {
        text: '¡Gracias por tu compra!',
        html: `
          

          <p>¡Gracias por tu compra!</p>

          <br />

          <p>Puedes consultar tu recibo en <a
href="https://dashboard.kumari.mx">KUMARI</a></p>

          <p>¿Estas list@ para vivir la experiencia Kumari?</p>

          <p>Descarga la aplicación oficial en tu dispositivo o ingresa en el siguiente link de
acceso en tu computadora: <a href="https://app.kumari.mx">KUMARI</a></p>

          <p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>

          <p>Tu pedido será procesado en las siguientes 24 horas, te avisaremos cuando tu
K-box haya sido enviada.</p>

```

<p>En caso de requerir factura envíanos tu recibo y datos fiscales a facturación@kumari.mx</p>

<p><strong>Archivos adjuntos:</strong></p>

<ul>

<li>

<p><strong>Instructivo Kumari:</strong>

Por favor, léelo con atención para que puedas estar bien informado acerca del sistema en general.

<a href="https://firebasestorage.googleapis.com/v0/b/kumari-d465a.appspot.com/o/Instructivo%20Kumari%20(2).pdf?alt=media&token=d37dcb52-2307-4a81-896d-659f3003788d">Instructivo</a>

</p>

<p>¡Empieza la diversión!</p>

<p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>

</li>

</ul>



},

}

let receipt = {

user: user.uid,

amount: !course.hasOwnProperty('id') ? this.allCourses.price : course.price,

name: user.name,

email: user.email,

```
    created_at: new Date(),  
    charge: {  
      amount: !course.hasOwnProperty('id') ? this.allCourses.price : course.price,  
      created: date1  
    }  
  }  
}
```

```
await db.collection('mails').add(mail)  
  
await db.collection('users').doc(user.uid).update({ complete_profile: true,  
date_deactivate: date2 })  
  
await db.collection('receipts').add(receipt)
```

//Esto lo hago para saber si compro el curso "todos los cursos" ya q este no cuenta con ID

```
if (course) {  
  if (!course.hasOwnProperty('id')) {
```

```
    this.registerAllCourses()
```

```
    return
```

```
  }  
}
```

```
let data = {  
  course: course.id,  
  user: this.newOrder.user,  
  date_active: date1,  
  date_deactivate: date2  
}
```

```
let responseCourse = await db.collection('shopping')
    .add(data)
```

```
let firstLesson = await db.collection('classes')
    .where('course', '==', course.id)
    .where('order', '==', 1)
    .get()
    .then(query => {
        query.forEach(async doc => {
            let lesson = doc.data()
            lesson.id = doc.id

            let responseFinished = await db.collection('finished_courses')
                .doc(this.newOrder.user)
                .collection('courses')
                .doc(course.id)
                .add({ lessons: [lesson.id], name: lesson.name })
        })
    })
```

```
this.startCourseRegister(course)
```

```
} catch (error) {
    this.$q.loading.hide()
    console.log(error);
}
```

```

    }
  },

  async startCourseRegister(course){
    try {
      let dataCourse = {
        name: course.name,
        lessons: [],
      }

      let response = await db.collection('finished_courses')
        .doc(this.newOrder.user)
        .collection('courses')
        .doc(course.id)
        .set(dataCourse)

      this.newOrder = {}

      this.isLoading = false
    } catch (error) {
      console.log(error);
    }
  },

  isObject(obj) {
    return Object.prototype.toString.call(obj) === '[object Object]';
  },

  async addOrderEdit(){

```



```

try {
  console.log(this.editOrder);

  if (this.isObject(this.editOrder.user)) {
    this.editOrder.user = this.editOrder.user.uid
  }

  let response = await db.collection('orders')
    .doc(this.editOrder.id)
    .update(this.editOrder)

  this.$q.notify({
    message: 'Orden editada',
    color: 'positive'
  })

  this.editOrder = null
  this.modalEditOrder = false

  // this.getOrders()
} catch (error) {
  console.log(error);
}
},

async getOrders(){
  this.orders = []

  try {
    let response = await db.collection('orders')

```

```

        .get()
        .then(query => {
            query.forEach(async doc => {
                let order = doc.data()
                order.id = doc.id

                let user = await db.collection('users')
                    .doc(order.user)
                    .get()

                order.user = user.data()

                this.orders.push(order)
            })
        })
    } catch (error) {
        console.log(error);
    }
},

```

```

async deleteOrder(order){
    try {
        this.$q.dialog({
            title: 'Eliminar orden',
            message: '¿Estas seguro de eliminar este orden?',
            cancel: true,
            persistent: true
        }).onOk(async () => {

```

```

    let response = await db.collection('orders').doc(order.id).delete()

    this.$q.notify({
      message: 'Orden eliminada',
      color: 'positive'
    })

    this.getOrders()
  })

} catch (error) {
  console.log(error);
}
},

generateShipment(data){
  console.log(data);
  this.isNew = false
  this.editOrder = data
  this.registerSkydropx(data)
},

async registerAllCourses(){
  try {
    let bar = new Promise((resolve, reject) => {
      this.allCoursesFilter.forEach(async (course, index, array) => {

        // let found = this.courses.find(doc => doc.id == course.id)

```

```
// if (!found) {
```

```
let date1 = moment(new Date()).unix();
```

```
let date2 = moment(new Date()).add(6, 'months').unix();
```

```
let data = {
```

```
  course: course.id,
```

```
  user: this.newOrder.user,
```

```
  date_active: date1,
```

```
  date_deactivate: date2
```

```
}
```

```
let responseCourse = await db.collection('shopping')
```

```
  .add(data)
```

```
let firstLesson = await db.collection('clases')
```

```
  .where('course', '==', course.id)
```

```
  .where('order', '==', 1)
```

```
  .get()
```

```
  .then(query => {
```

```
    query.forEach(async doc => {
```

```
      let lesson = doc.data()
```

```
      lesson.id = doc.id
```

```
let responseFinished = await db.collection('finished_courses')
```

```
  .doc(this.newOrder.user)
```

```
  .collection('courses')
```

```
  .doc(course.id)
```

```

        lesson.name })
    })
})

```

```

let dataCourse = {
  name: course.name,
  lessons: [],
}

```

```

await db.collection('finished_courses')
    .doc(this.newOrder.user)
    .collection('courses')
    .doc(course.id)
    .set(dataCourse)

// }

```

```

    if (index === array.length -1) resolve();
  })
});

```

```

bar.then(() => {
  this.newOrder = {}

  this.isLoading = false
});

```

```

    } catch (error) {
        console.log(error);
    }
},

async sendEmail(number, url){
    console.log('enviar email');
    try {
        let mail = {
            to: this.selectedUser.email,
            subject: 'Compra KUMARI',
            message: {
                text: '¡Tu K-Box esta en camino!',
                html: `
                    

                    <br />

                    <p>¡Felicidades! Este es tu número de orden: No. De ORDEN. ${number}</p>

                    <p>Tu K-Box esta en camino, puedes consultar el estatus de tu paquete después de
24 horas de tu compra en el siguiente link: <a href="${url}">LINK</a>.</p>

                    

                    `
            }
        }

        await db.collection('mails').add(mail)
    }
}

```

```

    } catch (error) {
        console.log(error);
    }
},

//Generar UUID
generateUUID(){
    var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
    var k = ['x','x','x','x','x','x','x','x','-','x','x','x','x','-','4','x','x','x','x','-','y','x','x','x','x','-','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];
    var u = "",i=0,rb=Math.random()*0xffffffff|0;
    while(i++<36) {
        var c=k[i-1],r=rb&0xf,v=c=='x'?r:(r&0x3|0x8);
        u+=(c=='-' || c=='4')?c:h[v];rb=i%8==0?Math.random()*0xffffffff|0:rb>>4
    }
    return u
},

},
}
</script>

```

## Questions.txt

```
<style scoped lang="scss">
```

```

.main-container{
    padding: 20px;
}

```

```

.main-title{
    font-size: 22px;
    font-weight: bold;
}

```

```
}
```

```
.form-question{  
  border-radius: 10px;  
}
```

```
.question{  
  border-radius: 10px;  
}
```

```
.question-header{  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
.question-header p:nth-child(1){  
  font-size: 18px;  
  font-weight: bold;  
}
```

```
.formulario{  
  
  button{  
    color: white;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;
```



```
}  
}
```

```
</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<p class="main-title">Preguntas Frecuentes</p>
```

```
<div class="row">
```

```
<div class="col-4">
```

```
<div class="row">
```

```
<div class="col-12 q-pa-md q-mt-sm form-question shadow-4">
```

```
<q-form
```

```
@submit="addQuestion"
```

```
class="q-gutter-md formulario"
```

```
>
```

```
<div class="row">
```

```
<div class="col-12">
```

```
<h5>Nueva Pregunta</h5>
```

```
</div>
```

```
<div class="col-12">
```

```
<q-input v-model="newQuestion.question" type="text" label="Pregunta" />
```

```
<q-input v-model="newQuestion.answer" type="textarea"  
label="Respuesta" />
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<q-btn label="Registrar" type="submit" />
```

```

        </div>
    </q-form>
</div>
</div>
</div>
<div class="col-8 q-pl-md q-pr-md">
    <div class="row">
        <div class="col-12 q-pa-md question shadow-4 q-mt-sm" v-for="(question, index) in
questions" :key="index">
            <div class="question-header">
                <p>{{ question.question }}</p>
                <div style="display: flex;">
                    <q-btn style="background-color: #1976D2; color: white;" label="Editar"
@click="modalEditQuestion = true, editQuestion = question" />
                    <q-btn color="red" class="q-ml-sm" label="Eliminar"
@click="deleteQuestion(question)" />
                </div>
            </div>
            <p>{{ question.answer }}</p>
        </div>
    </div>
</div>
</div>
</div>

<!-- modales -->
<q-dialog v-model="modalEditQuestion" persistent>
    <q-card v-if="editQuestion != null" style="min-width: 500px;">
        <q-card-section>
            <p class="main-title">Editar pregunta</p>
        </q-card-section>
    </q-card>
</q-dialog>

```

```

<q-card-section class="row">
  <div class="col-12">
    <q-form
      @submit="addQuestionEdit"
      class="q-gutter-md"
    >
      <q-input v-model="editQuestion.question" type="text" label="Pregunta" />
      <q-input v-model="editQuestion.answer" type="textarea" label="Respuesta" />
      <div>
        <q-btn label="Actualizar" type="submit" color="primary" v-close-popup/>
        <q-btn label="Cancelar" type="reset" color="primary" flat class="q-ml-sm" v-
close-popup />
      </div>
    </q-form>
  </div>
</q-card-section>
</q-card>
</q-dialog>
</div>
</q-page>
</template>

```

```

<script>
//Firebase
import { firebase, db, firestore, storage } from 'boot/firebase'
const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
var moment = require('moment');

```

```

export default {
  name: 'questions',

  data(){
    return{
      questions: [],
      questions: [],

      newQuestion: {
        question: "",
        answer: "",
      },

      modalSpace: false,
      modalEditQuestion: false,
      editQuestion: null,
    }
  },

  created() {
    this.getQuestions()
  },

  filters: {
    formatDay(time){
      let date = time.toDate()

      moment.locale('es');
    }
  }
}

```

```

        return moment(date).format('LLL');
    }
},

computed: {
    ...mapState('auth', [ 'user' ])
},

methods: {
    async addQuestion(){
        try {
            try {
                this.$q.dialog({
                    title: 'Agregar nueva pregunta',
                    message: '¿Quieres registrar esta pregunta?',
                    cancel: true,
                    persistent: true
                }).onOk(async () => {
                    let reponse = await db.collection('questions').add(this.newQuestion)
                    this.newQuestion.question = ""
                    this.newQuestion.answer = ""

                    this.$q.notify({
                        message: 'Pregunta registrada',
                        color: 'positive'
                    })
                })
            }

        } catch (error) {

```

```

        console.log(error);
    }

    } catch (error) {
        console.log(error);
    }
},

async getQuestions(){
    try {
        let response = await db.collection('questions')
            .onSnapshot(query => {
                this.questions = []
                let question
                query.forEach(doc => {
                    question = doc.data()

                    Object.defineProperty(question, 'id', {
                        enumerable: true,
                        configurable: true,
                        writable: true,
                        value: doc.id
                    });

                    this.questions.push(question)
                })
            })
    } catch (error) {
        console.log(error);
    }
}

```

```

    }
  },

  async addQuestionEdit(){
    try {
      try {
        let response = await db.collection('questions')
          .doc(this.editQuestion.id)
          .update(this.editQuestion)

        this.$q.notify({
          message: 'Pregunta editada',
          color: 'positive'
        })
      } catch (error) {
        console.log(error);
      }

    } catch (error) {
      console.log(error);
    }
  },

  async deleteQuestion(question){
    try {
      this.$q.dialog({
        title: 'Eliminar pregunta',
        message: '¿Estas seguro de eliminar esta pregunta?',
        cancel: true,

```

```

        persistent: true
    }).onOk(async () => {
        let response = await db.collection('questions').doc(question.id).delete()
        this.editQuestion = null
        this.$q.notify({
            message: 'Pregunta eliminado',
            color: 'positive'
        })
    })

    } catch (error) {
        console.log(error);
    }
},

},

}
</script>

```

### Users.txt

```

<style scoped lang="scss">

    .main-container{

        background-color: white;

        border-radius: 10px;

        padding: 20px;

    }


    .main-title{

        font-size: 22px;

        font-weight: bold;
    }

```



```
}
```

```
.btn-options{  
  border: none;  
  margin: 5px;  
  font-size: 18px;  
  color: white;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

```
.btn-modal{  
  border: none;  
  background-color: #24B8CB;  
  color: white;  
  border-radius: 15px;  
  padding: 10px 15px;  
  font-weight: bold;  
  font-size: 17px;  
}
```

```
.form-group{  
  margin-top: 20px;  
  
  label{  
    font-weight: bold;  
    font-size: 18px;  
  }  
}
```

```
.input-text{  
    width: 100%;  
    border: 2px solid #666666;  
    border-radius: 10px;  
    padding: 5px;  
}
```

```
.input-text:focus{  
    outline: none;  
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;
```

```
}
```

```
.badge-color{  
  width: 20px;  
  height: 20px;  
  border-radius: 50%;  
  margin-right: 10px;  
  cursor: pointer;
```

```
}
```

```
.active{  
  box-shadow: 1px 1px 10px rgba(0, 0, 0, 0.8);  
}
```

```
.form-group{  
  padding: 10px 5px;  
}
```

```
label{  
  font-size: 18px;  
  
  input{  
    margin-right: 10px;  
  }  
}
```

```
input[type=text]{  
  width: 100%;
```

```
border-radius: 10px;  
  
border: none;  
  
border: 1px solid black;  
  
padding: 5px;  
}
```

```
input[type=email]{  
    width: 100%;  
  
    border-radius: 10px;  
  
    border: none;  
  
    border: 1px solid black;  
  
    padding: 5px;  
}
```

```
input[type=date]{  
    width: 100%;  
  
    border-radius: 10px;  
  
    border: none;  
  
    border: 1px solid black;  
  
    padding: 5px;  
}
```

```
input[type=password]{  
    width: 100%;  
  
    border-radius: 10px;  
  
    border: none;  
  
    border: 1px solid black;  
  
    padding: 5px;  
}
```

```

select{
  width: 100%;

  border-radius: 10px;

  border: none;

  border: 1px solid black;

  padding: 5px;
}
</style>

<template>
  <q-page>
    <div class="main-container">
      <q-table
        title="Usarios"
        :data="users"
        :columns="columns"
        :filter="filter"
        row-key="email"
        flat
      >
        <template v-slot:top>
          <div class="flex justify-between items-center" style="width: 100%;">
            <span style="font-size: 25px; font-weight: bold;">Usuarios</span>
            <button class="btn-modal cursor-pointer" @click="modalUser = true">Crear
usuario</button>
            <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">
          <template v-slot:append>
            <q-icon name="search" />

```

```
        </template>
    </q-input>
</div>
```

```
</template>
<template v-slot:body-cell-name="props">
    <q-td :props="props">
        {{ props.row.children }}
    </q-td>
</template>
<template v-slot:body-cell-tutor="props">
    <q-td :props="props">
        {{ props.row.name }}
    </q-td>
</template>
<template v-slot:body-cell-age="props">
    <q-td :props="props">
        {{ props.row.birthdate }}
    </q-td>
</template>
<template v-slot:body-cell-courses="props">
    <q-td :props="props">
        5
    </q-td>
</template>
<template v-slot:body-cell-refresh="props">
    <q-td :props="props">
        {{ formatDate(props.row.date_deactivate) }}
    </q-td>
```

```

</template>

<template v-slot:body-cell-status="props">

  <q-td :props="props">

    <span v-if="props.row.status">Activo</span>

    <span v-else>No activo</span>

  </q-td>

</template>

<template v-slot:body-cell-options="props">

  <q-td :props="props">

    <button class="btn-options" style="background-color: #55B995;"
@click="modalEditUser = true, editUser = props.row">

      <q-icon name="edit" />

    </button>

    <button class="btn-options" style="background-color: red;"
@click="deleteUser(props.row)">

      <q-icon name="delete" />

    </button>

    <button class="btn-options" style="background-color: #0B5CF3;"
@click="showTest(props.row)">

      <q-icon name="quiz" />

    </button>

    <button class="btn-options" style="background-color: #F3920B;"
@click="deleteCourses(props.row)">

      <q-icon name="delete_forever" />

    </button>

  </q-td>

</template>

</q-table>

<div class="row">

```

```

    <div class="col-12 col-md-6 q-pa-sm">

        <canvas id="myChart" width="400" height="400"></canvas>

    </div>

</div>

</div>

<!-- <button @click="sendMailTest">Email de prueba</button> -->

<q-dialog v-model="modalUser" persistent>

    <q-card style="min-width: 800px;">

        <q-card-section class="row">

            <div class="col-12">

                <q-form

                    @submit="addUser"

                    class="q-gutter-md"

                >

                    <div class="row">

                        <div class="col-6" style="padding: 20px;">

                            <div class="form-group">

                                <label for="">Nombre del niño</label>

                                <br>

                                <input v-model="newUser.children" type="text">

                                <!-- <div class="error" v-if="!$v.newUser.children.required">Field is
required</div> -->

                            </div>

                            <div class="form-group">

                                <label for="">Fecha de nacimiento</label>

                                <br>

                                <input v-model="newUser.birthdate" type="date">

```



```

</div>

<div class="form-group">

  <label for="">Intereses</label>

  <br>

  <div>

    <div v-for="(interest, index) in interests" :key="index">

      <label>

        <input v-model="newUser.interest" :value="interest"
type="checkbox" name="interes">{{ interest }}

      </label>

      <br>

    </div>

  </div>

</div>

<div class="form-group">

  <label for="">Grado Academico</label>

  <br>

  <select v-model="newUser.academy" name="grado" id="">

    <option value="1">1ro de Primaria</option>

    <option value="2">2o de Primaria</option>

    <option value="3">3ro de Primaria</option>

    <option value="4">4ro de Primaria</option>

    <option value="5">5ro de Primaria</option>

    <option value="6">6ro de Primaria</option>

  </select>

</div>

<div class="form-group">

  <label for="">Escuela</label>

  <br>

```

```
        <input v-model="newUser.school" type="text">
    </div>
</div>
<div class="col-6" style="padding: 20px;">
    <div class="form-group">
        <label for="">Nombre del Padre o Tutor</label>
        <br>
        <input v-model="newUser.name" type="text">
    </div>
    <div class="row">
        <div class="col-6">
            <div class="form-group">
                <label for="">Direccion</label>
                <br>
                <input v-model="newUser.direction" type="text">
            </div>
        </div>
        </div>
        <div class="col-6">
            <div class="form-group">
                <label for="">Codigo Postal</label>
                <br>
                <input v-model="newUser.cp" type="text">
            </div>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-6">
```

```
<div class="form-group">
  <label for="">Ciudad</label>
  <br>
  <input v-model="newUser.city" type="text">
</div>
```

```
</div>
```

```
<div class="col-6">
  <div class="form-group">
    <label for="">Estado</label>
    <br>
    <input v-model="newUser.state" type="text">
  </div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
  <label for="">Telefono</label>
  <br>
  <input v-model="newUser.telephone" type="text">
</div>
```

```
<div class="form-group">
  <label for="">Email</label>
  <br>
  <input v-model="newUser.email" type="email">
</div>
```

```
<div class="form-group">
  <label for="">Contraseña</label>
  <br>
```

```

        <input v-model="newUser.password" type="password">
      </div>
    </div>
  </div>
  <div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalUser =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
  </div>
</q-form>
</div>
</q-card-section>
</q-card>
</q-dialog>

<q-dialog v-model="modalEditUser" persistent v-if="editUser != null">
  <q-card style="min-width: 800px;">
    <q-card-section class="row">
      <div class="col-12">
        <q-form
          @submit="addUserEdit"
          class="q-gutter-md">
          <div class="row">
            <div class="col-6" style="padding: 20px;">
              <div class="form-group">
                <label for="">Nombre del niño</label>
                <br>
                <input v-model="editUser.children" type="text">
                <!-- <div class="error" v-if="!$v.editUser.children.required">Field is
required</div -->

```

```

</div>

<div class="form-group">
  <label for="">Fecha de nacimiento</label>

  <br>

  <input v-model="editUser.birthdate" type="date">
</div>

<div class="form-group">
  <label for="">Intereses</label>

  <br>

  <div>
    <div v-for="(interest, index) in interests" :key="index">
      <label>

        <input v-model="editUser.interest" :value="interest"
type="checkbox" name="interes">{{ interest }}

      </label>

      <br>
    </div>
  </div>
</div>

<div class="form-group">
  <label for="">Grado Academico</label>

  <br>

  <select v-model="editUser.academy" name="grado" id="">
    <option value="1">1ro de Primaria</option>
    <option value="2">2o de Primaria</option>
    <option value="3">3ro de Primaria</option>
    <option value="4">4ro de Primaria</option>
    <option value="5">5ro de Primaria</option>
    <option value="6">6ro de Primaria</option>
  </select>
</div>

```

```
        </select>
    </div>
    <div class="form-group">
        <label for="">Escuela</label>
        <br>
        <input v-model="editUser.school" type="text">
    </div>
</div>
<div class="col-6" style="padding: 20px;">
    <div class="form-group">
        <label for="">Nombre del Padre o Tutor</label>
        <br>
        <input v-model="editUser.name" type="text">
    </div>
    <div class="row">
        <div class="col-6">
            <div class="form-group">
                <label for="">Direccion</label>
                <br>
                <input v-model="editUser.direction" type="text">
            </div>
        </div>
        <div class="col-6">
            <div class="form-group">
                <label for="">Codigo Postal</label>
                <br>
                <input v-model="editUser.cp" type="text">
            </div>
        </div>
    </div>
</div>
```

```
</div>
```

```
</div>
```

```
<div class="row">
```

```
<div class="col-6">
```

```
<div class="form-group">
```

```
<label for="">Ciudad</label>
```

```
<br>
```

```
<input v-model="editUser.city" type="text">
```

```
</div>
```

```
</div>
```

```
<div class="col-6">
```

```
<div class="form-group">
```

```
<label for="">Estado</label>
```

```
<br>
```

```
<input v-model="editUser.state" type="text">
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Telefono</label>
```

```
<br>
```

```
<input v-model="editUser.telephone" type="text">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Email</label>
```

```
<br>
```

```

        <input v-model="editUser.email" type="email">
    </div>
    <div class="form-group">
        <label for="">Contraseña</label>
        <br>
        <input v-model="editUser.password" type="password">
    </div>
</div>
</div>
<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalEditUser =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>
</q-card>
</q-dialog>

<!-- <q-dialog v-model="modalTest" persistent>
    <q-card style="min-width: 800px;">
        <q-card-section class="row">
            <div class="col-12">
                <canvas id="myChart" width="400" height="400"></canvas>
                <div class="q-pa-md flex justify-end">
                    <button class="btn-cancel" type="button" @click="modalTest =
false">Cerrar</button>
                </div>
            </div>
        </div>
    </q-card>
</q-dialog>
-->

```



```

        </q-card-section>
    </q-card>
</q-dialog> -->
</q-page>
</template>

<script>
import { db, storage, auth } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import VueTagsInput from '@johmun/vue-tags-input';

export default {
  name: 'feed',

  components: {
    VueTagsInput,
  },

  data(){
    return{
      myChart: null,
      newUser: {
        children: "",
        interest: [],
        role: 'user',
        complete_profile: false,
        check_status: true,

```

```
background_image: 'Azul.jpg',  
photo: 'https://firebasestorage.googleapis.com/v0/b/kumari-  
d465a.appspot.com/o/user_image%2Fimagen_2022-03-  
15_210459.png?alt=media&token=6d8c0cf3-7e89-4949-888b-35a8c8c294fe'
```

```
},
```

```
modalUser: false,  
modalEditUser: false,  
modalTest: false,  
editUser: null,
```

```
courselImage: null,
```

```
users: [],
```

```
filter: "",
```

```
columns: [
```

```
{  
  name: 'name',  
  required: true,  
  label: 'Nombre',  
  align: 'left',  
  field: row => row.name,  
  format: val => `${val}`,  
  sortable: true
```

```
},
```

```
{ name: 'email', align: 'left', label: 'Email', field: 'email', sortable: true },
```

```
{ name: 'tutor', align: 'left', label: 'Tutor', field: 'tutor', sortable: true },
```

```
{ name: 'age', align: 'left', label: 'Edad', field: 'age', sortable: true },
```

```
{ name: 'courses', align: 'left', label: 'Cursos', field: 'courses' },
```

```
{ name: 'refresh', label: 'Renovacion', field: 'refresh' },
```

```
    { name: 'status', label: 'Estatus', field: 'status' },
    { name: 'options', label: 'Opciones', field: 'options' }
  ],

  interests: [
    'Ciencia',
    'Matemáticas',
    'Robótica',
    'Lectura',
    'Escritura',
    'Fotografía',
    'Artes',
    'Baile',
    'Geografía',
    'Programación',
    'Naturaleza',
    'Animales',
    'Animación',
    'Videojuegos'
  ]
}
},
```

```
created() {
  this.getUsers()
},
```

```
computed: {
  ...mapState('auth', [ 'user' ])
```

```
},
```

```
methods: {
```

```
  showTest(user){
```

```
    this.modalTest = true
```

```
    if (user.test) {
```

```
      let data = {
```

```
        labels: [
```

```
          'Intrapersonal',
```

```
          'Lingüística',
```

```
          'Interpersonal',
```

```
          'Lógico matemática',
```

```
          'Espacial',
```

```
          'Cinética',
```

```
          'Musical',
```

```
          'Naturalista'
```

```
        ],
```

```
        datasets: [{
```

```
          label: 'Mi inteligencia',
```

```
          data: user.test,
```

```
          fill: true,
```

```
          backgroundColor: 'rgba(255, 99, 132, 0.2)',
```

```
          borderColor: 'rgb(255, 99, 132)',
```

```
          pointBackgroundColor: 'rgb(255, 99, 132)',
```

```
          pointBorderColor: '#fff',
```

```
          pointHoverBackgroundColor: '#fff',
```

```
          pointHoverBorderColor: 'rgb(255, 99, 132)'
```

```
        }]
```

```

    };

    const ctx = document.getElementById(`myChart`);
    if (this.myChart) {
        this.myChart.destroy();
    }

    this.myChart = new Chart(ctx, {
        type: 'radar',
        data: data,
        options: {
            elements: {
                line: {
                    borderWidth: 3
                }
            },
            scale: {
                ticks: {
                    beginAtZero: true,
                    max: 100,
                    min: 0,
                    stepSize: 10
                }
            }
        },
    });
} else {
    alert('Sin test contestado')
}

```

```
},
```

```
formatDate(date){  
  try {  
    if (date) {  
      var new_date = new Date(date * 1000);  
  
      return moment(new_date).format('L')  
    }else{  
      return 'Sin registro'  
    }  
  } catch (error) {  
    return 'Sin registro'  
  }  
},
```

```
truncateString(str, num) {  
  if (str.length <= num) {  
    return str  
  }  
  
  return str.slice(0, num) + '...'  
},
```

```
async addUser(){  
  try {  
    this.$q.dialog({  
      title: 'Agregar nuevo usuario',  
      message: '¿Quieres registrar este usuario?',
```

```

cancel: true,

persistent: true

}).onOk(async () => {

  let response = await db.collection('temporary').add(this.newUser)


  let mail = {

    to: this.newUser.email,

    subject: 'Registro KUMARI',

    message: {

      text: '¡Bienvenido a la comunidad Kumari!',

      html: `

        

        <p>¡Bienvenido a la comunidad Kumari!</p>

        <br />

        <p>En este espacio seguro podrás encontrar sorpresas, nuevas formas de
aprender y sobre todo mucha diversión. Te invitamos a descargar la aplicación oficial en el
siguiente link: <a
href="https://play.google.com/store/apps/details?id=com.kumari.kumari">Link</a></p>

        <p><strong>¡Suscríbete y obtén tu K-Box hasta la puerta de tu
casa!</strong></p>

        <p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>

        

      `

    }

  }
}

```

```

        await db.collection('mails').add(mail)

        this.modalUser = false
    })

    } catch (error) {
        console.log(error);
    }
},

async addUserEdit(){
    try {
        let response = await db.collection('users')
            .doc(this.editUser.uid)
            .update(this.editUser)

        this.$q.notify({
            message: 'Usuario editado',
            color: 'positive'
        })

        this.modalEditUser = false
    } catch (error) {
        console.log(error);
    }
},

async getUsers(){

```



```

try {
  let response = await db.collection('users')
    .where('role', '==', 'user')
    .onSnapshot(query => {
      this.users = []

      query.forEach(async doc => {
        let user = doc.data()
        user.id = doc.id

        this.users.push(user)
      })
    })
} catch (error) {
  console.log(error);
}
},

async deleteCourses(user){
  try {
    this.$q.dialog({
      title: 'Eliminar cursos',
      message: '¿Estas seguro de eliminar los cursos de este usuario?',
      cancel: true,
      persistent: true
    }).onOk(async () => {
      let response = await db.collection('shopping').where('user','==',user.uid);

      response.get().then((querySnapshot) => {

```

```
    querySnapshot.forEach((doc) => {  
        doc.ref.delete();  
    });  
});
```

```
let receipts = await db.collection('receipts').where('user','==',user.uid);
```

```
receipts.get().then((querySnapshot) => {  
    querySnapshot.forEach((doc) => {  
        doc.ref.delete();  
    });  
});
```

```
let orders = await db.collection('orders').where('user','==',user.uid);
```

```
orders.get().then((querySnapshot) => {  
    querySnapshot.forEach((doc) => {  
        doc.ref.delete();  
    });  
});
```

```
await db.collection('finished_courses').doc(user.uid).delete()  
await db.collection('users').doc(user.uid).update({ date_deactivate: null,  
complete_profile: false })
```

```
this.$q.notify({  
    message: 'Informacion eliminada',  
    color: 'positive'  
})
```

```

    })

    } catch (error) {
        console.log(error);
    }
},

async deleteUser(user){
    try {
        this.$q.dialog({
            title: 'Eliminar usuario',
            message: '¿Estas seguro de eliminar este usuario?',
            cancel: true,
            persistent: true
        }).onOk(async () => {
            let response = await db.collection('shopping').where('user','==',user.uid);

            response.get().then((querySnapshot) => {
                querySnapshot.forEach((doc) => {
                    doc.ref.delete();
                });
            });

            let receipts = await db.collection('receipts').where('user','==',user.uid);

            receipts.get().then((querySnapshot) => {
                querySnapshot.forEach((doc) => {
                    doc.ref.delete();
                });
            });
        });
    }
}

```

$$\} );$$

```
let orders = await db.collection('orders').where('user','==',user.uid);
```

```
orders.get().then((querySnapshot) => {
```

```
querySnapshot.forEach((doc) => {
```

```
doc.ref.delete();
```

$$\}));$$
$$\}));$$

```
await db.collection('finished_courses').doc(user.uid).delete()
```

```
let userRef = await db.collection('users').doc(user.uid).delete()
```

```
let index = this.users.findIndex(doc => doc.uid == user.uid)
```

$$\})$$

```
} catch (error) {
```

```
console.log(error);
```

}

 $\},$ 

```
//Generar UUID
```

```
generateUUID(){
```

```
var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
```

```
var k = ['x','x','x','x','x','x','x','x','-', 'x','x','x','x','-', '4','x','x','x','-', 'y','x','x','x','-', 'x','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];
```

```
var u = "",i=0,rb=Math.random()*0xffffffff|0;
```

```
while(i++<36) {
```

```

    var c=k[i-1],r=rb&0xf,v=c=='x'?r:(r&0x3|0x8);
    u+=(c=='-'||c=='4')?c:h[v];rb=i%8==0?Math.random()*0xffffffff|0:rb>>4
  }
  return u
},

```

```

async uploadImage(){
  // this.isLoading = true;
  try {
    let metadata = {
      contentType: "image/jpeg",
      contentLanguage: "es",
    }
    let fotold = this.generateUUID()
    const refImg = ref.child('courses/' + fotold + '.jpg')
    return refImg.put(this.coursetImage, metadata)
    .then(e => {
      console.log(e)
      return ref.child('courses/' + fotold + '.jpg').getDownloadURL()
      .then(async (url) => {
        return url
      })
    })
    // this.getResizePath(fotold)
  })
  .catch(error => console.log(error))
} catch (error) {
  console.log(error);
}

```

```
},
```

```
getImage(e){
```

```
  this.courselImage = e.target.files[0]
```

```
  console.log(this.courselImage);
```

```
},
```

```
async sendMailTest(){
```

```
  try {
```

```
    let mail = {
```

```
      to: 'gera_conecta@hotmail.com',
```

```
      subject: 'Compra KUMARI',
```

```
      message: {
```

```
        text: '¡Gracias por tu compra!',
```

```
        html: `
```

```
          
```

```
          <p>¡Gracias por tu compra!</p>
```

```
          <br />
```

```
          <p>Puedes consultar tu recibo en <a
```

```
          href="https://dashboard.kumari.mx">KUMARI</a></p>
```

```
          <p>¿Estas list@ para vivir la experiencia Kumari?</p>
```

```
          <p>Descarga la aplicación oficial en tu dispositivo o ingresa en el siguiente link de
          acceso en tu computadora: <a href="https://app.kumari.mx">KUMARI</a></p>
```

```
          <p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>
```

<p>Tu pedido será procesado en las siguientes 24 horas, te avisaremos cuando tu K-box haya sido enviada.</p>

<p>En caso de requerir factura envíanos tu recibo y datos fiscales a facturación@kumari.mx</p>

<p><strong>Archivos adjuntos:</strong></p>

<ul>

<li>

<p><strong>Instructivo Kumari:</strong>

Por favor, léelo con atención para que puedas estar bien informado acerca del sistema en general.

<a href="https://firebasestorage.googleapis.com/v0/b/kumari-d465a.appspot.com/o/Instructivo%20Kumari%20(2).pdf?alt=media&token=d37dcb52-2307-4a81-896d-659f3003788d">Instructivo</a>

</p>

<p>¡Empieza la diversión!</p>

<p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>

</li>

</ul>



}

}

await db.collection('mails').add(mail)

console.log('enviado');

} catch (error) {

```
        console.log(error);
    }
}

},
}
</script>
```

## Auth

### Login.txt

```
<style scoped>
```

```
.card{
    width: 700px;
    border-radius: 20px;
}
```

```
.card .content{
    padding: 20px 40px;
}
```

```
.card .side{
    width: 200px;
    background-color: #F5F6F8;
}
```

```
.card .side img{
    width: 200px;
    object-fit: cover;
    object-position: center;
}
```



```
.card p{  
    line-height: 12px;  
}
```

```
.card p:nth-child(1){  
    font-weight: bold;  
    font-size: 22px;  
}
```

```
.card p:nth-child(2){  
    color: #676879;  
    font-size: 16px;  
}
```

```
.card input[type=text], input[type=password], input[type=email]{  
    width: 100%;  
    background-color: transparent;  
    padding: 5px;  
}
```

```
.card button{  
    width: 100%;  
    background-color: #A0C62E;  
    color: white;  
    margin-top: 10px;  
    padding: 10px 10px;  
    border: none;  
    border-radius: 10px;
```



```

<div class="col-12 col-md-7 img-container">

  

</div>

<div class="col-12 col-md-5 text-center" style="display: flex; flex-direction: column;
justify-content: center;">

  <input v-model="user.email" type="email" placeholder="Correo">

  <input class="q-mt-md" v-model="user.password" type="password"
placeholder="Contraseña">


  <button @click="onSubmit">Iniciar sesion</button>


  <router-link class="q-mt-md" to="/forgot-password">¿Olvidaste tu
contraseña?</router-link>

  <router-link class="q-mt-md" to="/complete-profile">¿No tienes una cuenta?
Registrate</router-link>

</div>

</div>

</div>

</div>

<div class="card" v-if="step == 2">

  <div class="row">

    <div class="col-12" style="text-align: center;">

      <q-spinner-ball

        color="primary"

        size="4em"

      />

      <q-tooltip :offset="[0, 8]">Wait a moment!</q-tooltip>

    </div>

  </div>

</div>

</div>

```

```

        </div>
    </div>
</div>
</q-page>
</template>

<script>
import { db } from 'boot/firebase'

import { mapState, mapActions } from 'vuex'
import { required, minLength, email } from 'vuelidate/lib/validators'

export default {
  name: 'Login',

  data(){
    return{
      isPwd: true,

      step: 1,

      user: {
        email: "",
        password: ""
      }
    }
  },

  validations: {

```

```

user: {
  email: {
    email,
    required,
  },
  password: {
    required,
  },
},
},

methods: {
  ...mapActions('auth', [ 'loginUser', 'closeSesion' ]),

  async onSubmit(){
    try {
      if (!this.$v.user.email.required || !this.$v.user.password.required) {
        this.$q.notify({
          message: 'The info is required',
          color: 'negative'
        })

        return
      }

      this.loginUser(this.user).then((response) => {
        console.log('binnieeeee: ', response);
      }).catch(error => {
        console.log(error);
      });
    }
  }
}

```

```
        this.$q.notify({
            message: error.message,
            color: 'negative'
        })
    })

    } catch (error) {
        console.log(error);
    }
}
},
}
</script>
```

### Profile.txt

```
<style scoped lang="scss">
```

```
.card{
    width: 900px;
    border-radius: 20px;
```

```
p{
    font-size: 18px;
}
```

```
.sec-1{
    margin-top: 20px;
}
```

```
.main-container{  
    position: relative;  
    background-color: white;  
    border-radius: 40px;  
    padding: 40px;  
    min-height: 55vh;  
    overflow: scroll;  
    display: flex;  
    justify-content: space-around;  
    align-items: center;  
    flex-direction: column;  
    box-shadow: 0px 0px 40px rgba(0, 0, 0, 0.589);  
  
    -ms-overflow-style: none; /* IE and Edge */  
    scrollbar-width: none; /* Firefox */
```

```
&::-webkit-scrollbar {  
    display: none;  
}
```

```
.monito_1{  
    position: absolute;  
    top: 30%;  
    left: -5%;  
    width: 20%;  
}
```

```
.monito_2{  
    position: absolute;
```

```
    top: 0;
    right: -5%;
    width: 20%;
}
```

```
.monito_3{
    position: absolute;
    top: 20%;
    left: -1%;
    width: 12%;
}
```

```
.monito_4{
    position: absolute;
    top: 25%;
    right: -7%;
    width: 22%;
}
```

```
}
```

```
.form-group{
    padding: 10px 5px;
}
```

```
label{
    font-size: 18px;
```

```
input{
```



```
        margin-right: 10px;
    }
}
```

```
input[type=text]{
    width: 100%;
    border-radius: 10px;
    border: none;
    border: 1px solid black;
    padding: 5px;
}
```

```
input[type=date]{
    width: 100%;
    border-radius: 10px;
    border: none;
    border: 1px solid black;
    padding: 5px;
}
```

```
input[type=password]{
    width: 100%;
    border-radius: 10px;
    border: none;
    border: 1px solid black;
    padding: 5px;
}
```

```
select{
```

```
width: 100%;  
  
border-radius: 10px;  
  
border: none;  
  
border: 1px solid black;  
  
padding: 5px;  
}
```

```
.btn-next{  
  
background-color: #A0C62E;  
  
min-width: 150px;  
  
color: white;  
  
border-radius: 10px;  
  
padding: 8px 20px;  
  
border: none;  
  
margin-top: 20px;  
  
cursor: pointer;  
  
text-align: center;  
}
```

```
.btn-back{  
  
width: 45px;  
  
height: 45px;  
  
border-radius: 50%;  
  
padding: 5px;  
  
border: none;  
  
margin-top: 20px;  
  
cursor: pointer;  
  
display: flex;  
  
justify-content: center;
```

```
align-items: center;
```

```
img{  
  width: 45px;  
}  
}
```

```
.title{  
  font-size: 2em;  
  font-weight: bold;  
  line-height: 20px;  
}  
}
```

```
.main{  
  background-image: url('/recursos_2/Verde.jpg');  
}
```

```
.pointer{  
  width: 10px;  
  height: 10px;  
  border-radius: 50%;  
  border: 1px solid #E72B81;  
  margin: 10px;
```

```
&.active{  
  background-color: #E72B81;  
}  
}
```

```
@media (max-width: 1023px) {  
  .card{  
    width: 100%;  
  }  
  
  .monito_1{  
    display: none;  
  }  
  
  .monito_2{  
    display: none;  
  }  
  
  .monito_3{  
    display: none;  
  }  
  
  .monito_4{  
    display: none;  
  }  
}  
</style>
```

```
<template>  
  <q-page  
    class="window-height window-width row justify-center items-center main"  
  >  
    <div class="column q-pa-lg">
```

```
<div class="row">
```

```
<div class="col-12">
```

```
<div class="card" v-if="step == 1">
```

```
<div class="main-container">
```

```

```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-12 text-center">
```

```
<p class="title">Creemos tu cuenta</p>
```

```
</div>
```

```
</div>
```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-12 col-md-3"></div>
```

```
<div class="col-12 col-md-6" style="padding: 20px;">
```

```
<div class="form-group">
```

```
<label for="">Email</label>
```

```
<br>
```

```
<input v-model="profile.email" type="text">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Contraseña</label>
```

```
<br>
```

```
<input v-model="profile.password" type="password">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Confirmar contraseña</label>
```

```
<br>
```

```
<input type="password">
```

```

        </div>
    </div>
    <div class="col-12 col-md-3"></div>
</div>
<div class="row" style="width: 100%;">
    <div class="col-3">
        <div class="flex justify-start items-center">

            </div>
        </div>
        <div class="col-6">
            <div class="flex justify-around items-center">
                <router-link to="/" class="btn-next" style="margin-right: 10px; text-decoration: none; background-color: #E72B81;">
                    Iniciar sesión
                </router-link>
                <button class="btn-next" @click="step = 2">Continuar</button>
            </div>
        </div>
    <div class="col-3"></div>
    <div class="col-12 flex justify-center items-center">
        <span class="pointer active"></span>
        <span class="pointer"></span>
        <span class="pointer"></span>
        <span class="pointer"></span>
        <span class="pointer"></span>
    </div>
</div>

```

```
</div>
```

```
</div>
```

```
<div class="card" v-if="step == 2">
```

```
<div class="main-container">
```

```

```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-12 text-center">
```

```
<p class="title">Datos del Niño(a)</p>
```

```
</div>
```

```
</div>
```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-12 col-md-2"></div>
```

```
<div class="col-12 col-md-4" style="padding: 20px;">
```

```
<div class="form-group">
```

```
<label for="">Nombre del niño</label>
```

```
<br>
```

```
<input v-model="profile.children" type="text">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Fecha de nacimiento</label>
```

```
<br>
```

```
<input v-model="profile.birthdate" type="date">
```

```
</div>
```

```
</div>
```

```
<div class="col-12 col-md-4" style="padding: 20px;">
```

```
<div class="form-group">
```

```
<label for="">Grado Academico</label>
```

```

<br>
<select v-model="profile.academy" name="grado" id="">
  <option value="1">1ro de Primaria</option>
  <option value="2">2o de Primaria</option>
  <option value="3">3ro de Primaria</option>
  <option value="4">4ro de Primaria</option>
  <option value="5">5ro de Primaria</option>
  <option value="6">6ro de Primaria</option>
</select>
</div>
<div class="form-group">
  <label for="">Escuela</label>
  <br>
  <input v-model="profile.school" type="text">

</div>
</div>
<div class="col-12 col-md-2"></div>
</div>
<div class="row" style="width: 100%;">
  <div class="col-3">
    <div class="flex justify-start items-center">
      <button class="btn-back" @click="step--">

      
    </button>
  </div>
</div>
<div class="col-6">

```



```

        <div class="flex justify-around items-center">
            <router-link to="/" class="btn-next" style="margin-right: 10px; text-
decoration: none; background-color: #E72B81;">
                Iniciar sesión
            </router-link>
            <button class="btn-next" @click="step = 3">Continuar</button>
        </div>
    </div>
<div class="col-3"></div>
<div class="col-12 flex justify-center items-center">
    <span class="pointer"></span>
    <span class="pointer active"></span>
    <span class="pointer"></span>
    <span class="pointer"></span>
    <span class="pointer"></span>
</div>
</div>

</div>
</div>

<div class="card" v-if="step == 3">
    <div class="main-container">
        
        <div class="row" style="width: 100%;">
            <div class="col-12 text-center">
                <p class="title">Intereses del Niño(a)</p>
            </div>
        </div>
    </div>
</div>

```

```

<div class="row" style="width: 100%;">

  <div class="col-12 col-md-1"></div>

  <div class="col-12 col-md-5" style="padding: 20px;">

    <div class="form-group">

      <div>

        <div v-for="(interest, index) in interests" :key="index">

          <label>

            <input v-model="profile.interest" :value="interest"
type="checkbox" name="interes">{{ interest }}

          </label>

          <br>

        </div>

      </div>

    </div>

    <div class="col-12 col-md-5" style="padding: 20px;">

      <div class="form-group">

        <div>

          <div v-for="(interest, index) in interests2" :key="index">

            <label>

              <input v-model="profile.interest" :value="interest"
type="checkbox" name="interes">{{ interest }}

            </label>

            <br>

          </div>

        </div>

      </div>

    </div>

  </div>

```

```

        <div class="col-12 col-md-1"></div>
    </div>
    <div class="row" style="width: 100%;">
        <div class="col-12 col-md-3">
            <div class="flex justify-start items-center">
                <button class="btn-back" @click="step--">
                    
                </button>
            </div>
        </div>
        <div class="col-12 col-md-6">
            <div class="flex justify-around items-center">
                <router-link to="/" class="btn-next" style="margin-right: 10px; text-decoration: none; background-color: #E72B81;">
                    Iniciar sesión
                </router-link>
                <button class="btn-next" @click="step = 4">Continuar</button>
            </div>
        </div>
        <div class="col-12 col-md-3"></div>
        <div class="col-12 flex justify-center items-center">
            <span class="pointer"></span>
            <span class="pointer"></span>
            <span class="pointer active"></span>
            <span class="pointer"></span>
            <span class="pointer"></span>
        </div>
    </div>
</div>

```

</div>

<div class="card" v-if="step == 4">

<div class="main-container">



<div class="row" style="width: 100%;">

<div class="col-12 text-center">

<p class="title">Datos para el envio del material</p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-12 col-md-2"></div>

<div class="col-12 col-md-4">

<div class="form-group">

<label for="">Nombre del Padre o Tutor</label>

<br>

<input v-model="profile.name" type="text">

</div>

<div class="form-group">

<label for="">Telefono</label>

<br>

<input v-model="profile.telephone" type="text">

</div>

</div>

<div class="col-12 col-md-2">

<div class="form-group">

<label for="">Direccion</label>

<br>

<input v-model="profile.direction" type="text">

```
</div>

<div class="form-group">
  <label for="">Ciudad</label>

  <br>

  <input v-model="profile.city" type="text">
</div>

</div>

<div class="col-12 col-md-2">
  <div class="form-group">
    <label for="">Codigo Postal</label>

    <br>

    <input v-model="profile.cp" type="text">
  </div>

  <div class="form-group">
    <label for="">Estado</label>

    <br>

    <input v-model="profile.state" type="text">
  </div>
</div>

<div class="col-12 col-md-2"></div>
</div>

<div class="row" style="width: 100%;">
  <div class="col-12 col-md-3">
    <div class="flex justify-start items-center">
      <button class="btn-back" @click="step--">
        
      </button>
    </div>
  </div>
</div>
```

```

<div class="col-12 col-md-6">
  <div class="flex justify-around items-center">
    <router-link to="/" class="btn-next" style="margin-right: 10px; text-
decoration: none; background-color: #E72B81;">
      Iniciar sesión
    </router-link>
    <button class="btn-next" @click="step = 5">Continuar</button>
  </div>
</div>
<div class="col-12 col-md-3"></div>
<div class="col-12 flex justify-center items-center">
  <span class="pointer"></span>
  <span class="pointer"></span>
  <span class="pointer"></span>
  <span class="pointer active"></span>
  <span class="pointer"></span>
</div>
</div>
</div>
</div>

```

```

<div class="card" v-if="step == 5">
  <div class="main-container">
    
    <div class="row" style="width: 100%;">
      <div class="col-12 text-center">
        <p class="title">Términos y condiciones de uso</p>
      </div>
    </div>
  </div>
</div>

```

</div>

</div>

<div class="row flex justify-center">

<div class="col-12" style="max-height: 30vh; width: 80%; overflow-y: scroll; border: 1px solid black; border-radius: 20px; padding: 20px;">

<h5 class="text-center">APRENDE CON KUMARI S.A. DE C.V.</h5>

<p>Bienvenido al sitio de internet <https://www.kumari.mx/> (en lo sucesivo el "sitio web") administrado por APRENDE CON KUMARI S.A. de C.V. (en lo sucesivo "Kumari") con domicilio fiscal en Blvd Ortiz Mena 1834, Los Sicomoros, Chihuahua.</p>

<p>El presente contrato establece los términos y condiciones generales (en lo sucesivo "términos y condiciones") aplicables al uso de los contenidos y/o información y/o productos y/o servicios que "Kumari" ofrece a todas las personas que utilicen y/o naveguen y/o visiten y/o accedan al "sitio web" (en lo sucesivo el "usuario" y/o los "usuarios").</p>

<p>Cualquier persona que desee acceder y/o utilizar y/o visitar y/o navegar y/u obtener y/o descargar y/o proporcionar información del "sitio web" en lo presente y/o futuro podrá hacerlo sujetándose a los "términos y condiciones" por lo que previamente deberá leerlos cuidadosamente.</p>

<p>Los servicios que son ofrecidos y/o publicados por el sitio web pueden ser creadas y/o cobradas y/o enviadas y/o presentadas por una página web tercera y en tal caso estarían sujetas a sus propios Términos y Condiciones. En algunos casos, para adquirir un servicio, será necesario el registro por parte del usuario, con ingreso de datos personales fidedignos y definición de una contraseña, mismos que deberán ser aprobados por el sitio web.</p>

<p>Las compras y transacciones que se lleven a cabo por medio del sitio web estarán sujetas a un proceso de confirmación y verificación, el cual podría incluir la verificación y disponibilidad de producto, validación de la forma de pago, validación de la factura (en caso de existir) y el cumplimiento de las condiciones requeridas por el medio de pago seleccionado. En algunos casos puede que se requiera una verificación por medio de correo electrónico.</p>

<p>El uso del "sitio web" constituye una aceptación de hecho de este acuerdo por lo que si cualquier "usuario" no acepta y/o no está de acuerdo con los "términos y condiciones" deberá de abstenerse de utilizar el "sitio web".</p>

<h6>1. Privacidad y/o datos personales</h6>

<p>“Kumari” a través del “sitio web” puede usar y/o captar información con la finalidad de brindar un mejor servicio a todos los “usuarios”. El manejo de la información del “usuario” se rige por el aviso de privacidad por lo que en caso de que el “usuario” proporcione algún dato personal se considerará como su absoluta y expresa aceptación a nuestro aviso de privacidad</p>

<p>El “usuario” acepta la responsabilidad en que pudiera incurrir derivado del incumplimiento a la privacidad y/o confidencialidad de datos y/o información de “Kumari” y/o de otro “usuario” y/o de algún tercero. En el supuesto de un requerimiento judicial de revelar información confidencial propiedad del “usuario”, “Kumari” notificará por escrito al “usuario” a fin de que se encuentre en aptitud de ejercer cualquier defensa a la que tuviere derecho.</p>

## <h2>2. Derechos de autor y propiedad industrial </h2>

<p>“Kumari” es titular de todo el contenido y/o derechos de propiedad industrial y/o derechos de propiedad intelectual del “sitio web” incluyendo sin limitar el texto y/o gráficos y/o logotipos y/o iconos y/o imágenes y/o grabaciones de audio y/o descargas digitales y/o compilaciones de datos y software y/o código fuente y/o marcas y/o nombres comerciales y/o avisos comerciales y/o combinaciones de colores y/o diseños y/o nombres de dominio y/o nombres comerciales y/o html y/o clips de video y por ello están protegidos por la legislación mexicana en materia de propiedad industrial e intelectual.</p>

<p>Queda expresamente prohibida la reproducción y/o modificación y/o distribución y/o difusión de cualquier contenido del “sitio web” sin la autorización previa y por escrito de “Kumari”. </p>

<p>Las marcas y/o nombres comerciales y/o avisos comerciales del “sitio web” no podrán ser utilizadas en relación con cualquier producto y/o servicio que no sea de “Kumari”.</p>

<p>Las marcas y/o nombres comerciales y/o avisos comerciales que no sean propiedad de “Kumari” que aparezcan en el “sitio web” son propiedad de sus respectivos dueños y se utilizan con permiso expreso; sin embargo “Kumari” no garantiza ni afirma que la utilización por parte de los “usuarios” de cualquier “material” de terceros publicado en el “sitio web” no infringe los derechos de los terceros por lo que el uso de cualquier “material” del “sitio web” es única y exclusivamente responsabilidad del “usuario”.</p>

<p>“Kumari” se reserva el derecho de revisar y/o complementar total o parcialmente el contenido del “sitio web”.</p>

## <h2>3. Reclamaciones de derechos de autor</h2>



<p>“Kumari” protege y respeta la propiedad industrial de terceros por lo que en caso de que el “usuario” crea que su obra ha sido copiada y/o es infringida por el contenido del “sitio web” podrá notificar de manera personal y por escrito a “Kumari” en el domicilio que se proporciona en los “términos y condiciones”. La notificación deberá cumplir con lo establecido en la Ley Federal del Derecho de Autor por lo que cualquier notificación que indique una supuesta infracción y que incumpla sustancialmente con las disposiciones legales no se considerará como un “conocimiento real” y/o “conocimiento de los hechos” por parte de “Kumari”.</p>

#### <h6>4. Uso</h6>

<p>Al utilizar y/o acceder y/o visitar y/o navegar y/o descargar y/u obtener y/o proporcionar información en el “sitio web” se considerará que los “usuarios” han leído y aceptado los “términos y condiciones” y han aceptado estar legalmente sujetos a los mismos incluida la Política de Privacidad.</p>

<p>“Kumari” se reserva el derecho de eliminar y/o impedir que cualquier persona y/o “usuario” acceda al “sitio web” a su entera discreción.</p>

<p>Los servicios ofrecidos en el “sitio web” son exclusivamente de uso personal, es decir, los “usuarios” no podrán modificar y/o distribuir y/o transmitir y/o presentar y/o ejecutar y/o publicar y/o autorizar y/o transferir y/o enajenar cualquier tipo de información y/o contenido y/o material obtenido del “sitio web”.</p>

<p>El acceso a la plataforma es de carácter interactivo por lo que es inmediato al concretar la compra, por lo que es requisito y se realiza mediante un mail para con ello realizar un perfil y/o usuario, así como la contraseña de dicho perfil y/o usuario.</p>

<p>El producto físico llegará en un plazo de cinco a ocho días hábiles, el tiempo de entrega está sujeto a cambios imprevistos y/o fortuitos que serán notificados de forma electrónica, dichos cambios imprevistos y/o fortuitos no son imputables a “Kumari” al ser atribuibles a terceros y/o diversas sociedades de las que “Kumari” utilice sus servicios para la entrega de sus productos.</p>

<p>Los precios de los servicios ofrecidos y/o publicados en este Sitio Web son válidos solamente en las compras realizadas en este sitio web.</p>

#### <h6>5. Modificación al “producto”</h6>

<p>El “usuario” no podrá realizar modificaciones al “producto” incluyendo sin limitar agregar y/o eliminar servicios y/o productos una vez que la orden sea confirmada por lo que el “usuario” es el único responsable de verificar la selección de servicios y/o planes funerarios del “sitio web”.</p>

## <h6>6. Precios</h6>

<p>Los precios de los servicios y/o productos del “sitio web” incluyen el impuesto al valor agregado.</p>

## <h6>7. Método de pago</h6>

<p>Los “usuarios” aceptan que los únicos métodos de pago de cualquier servicio y/o producto del “sitio web” serán mediante tarjeta de crédito o débito.</p>

<p>Una vez que el sistema de pagos de “Kumari” refleje el pago realizado por el “usuario” se enviará al “usuario” la confirmación del “producto”.</p>

<p>El “usuario” será el único responsable de realizar el pago del “producto” por lo que “Kumari” no se hace responsable si la tarjeta bancaria utilizada por el “usuario” en el “sitio web” fuere rechazada mencionando en forma enunciativa más no limitativa las posibles causas de rechazo: tarjeta expirada y/o fondos insuficientes y/o prevención antifraude y/o datos incorrectos proporcionados en el sistema de pagos del “sitio web”, por lo que el “usuario” deberá utilizar una forma de pago alterna.</p>

<p>“Kumari” no administra y/o almacena y/o procesa y/o trasmite información y/o datos relacionados a las transacciones y/o tarjetas bancarias de los “usuarios” por lo que los “usuarios” deslindan de cualquier responsabilidad y/o reclamo y/o daño a “Kumari” por robo y/o fraude y/o clonación y/o cualquier otra acción que afecte la cuenta bancaria del “usuario”.</p>

<p>El “sitio web” de “Kumari” únicamente acepta pagos de personas mayores de edad.</p>

## <h6>8. Reembolso y garantía</h6>

<p>En caso de estimarse de parte del usuario que el producto se recibió con fallas de origen y/o malas condiciones deberá enviar un correo electrónico a: servicio@kumari.mx dentro de los de 3 días hábiles siguientes a la recepción física del producto anexando al mismo evidencia fotografica y relatando las fallas y/o malas condiciones, por lo que se someterá a consideración y discreción de “Kumari” la aplicación de la garantía del producto físico.</p>

<p>La garantía no cubre averías y/o daños ocasionados por uso indebido de parte del usuario. Los términos de la garantía están asociados únicamente a fallas de origen y/o

funcionamiento en condiciones normales y/o naturales y/o indicadas del producto por lo que sólo se harán efectivos estos términos si este ha sido usado correcta y debidamente .</p>

## <h6>9. Comunicaciones electrónicas</h6>

<p>Al visitar y/o utilizar el “sitio web” y/o enviar correos electrónicos por medio del “sitio web”, los “usuarios” aceptan la comunicación electrónica con “Kumari” por lo que otorgan su consentimiento expreso para recibir comunicaciones electrónicas de parte de “Kumari” aceptando y reconociendo el “usuario” que dichas comunicaciones electrónicas satisfacen y se ajustan a cualquier requisito legal.</p>

## <h6>10. Terceros</h6>

<p>“Kumari” proporciona información y/o enlaces a otros sitios web sobre los que “Kumari” no tiene control alguno y/o que no han sido revisados por “Kumari” por lo que no se hace responsable de la disponibilidad y/o daño de esos sitios web y/o recursos externos y/o contenido disponible en esos sitios web por lo que en todo momento los “usuarios” deberán ejercer su propio juicio en cuanto a la idoneidad y adecuación de la información proporcionada por terceros.</p>

<p>“Kumari” no recomienda ni respalda a personas físicas y/o empresas y/o servicios y/o productos de terceros, a excepción de las declaraciones expresas hechas por “Kumari” en el “sitio web”.</p>

<p>“Kumari” proporciona a los “usuarios” herramientas de terceros que permiten un mayor funcionamiento del “sitio web” por lo que el uso por parte de los “usuarios” de esa funcionalidad se encuentra sujeto a los “términos y condiciones” del tercero.</p>

## <h6>11. Uso indebido</h6>

<p>“Kumari” se reserva el derecho de suspender y/o restringir el acceso a los “usuarios” que utilicen el “sitio web” con fines ilícitos y/o prohibidos de conformidad con la legislación mexicana y los “términos y condiciones”.</p>

<p>Queda estrictamente prohibido el uso por parte del “usuario” de herramientas diseñadas para comprometer la seguridad y/o alterar el “sitio web” por lo que si el “usuario” se ve involucrado en la violación a los sistemas de seguridad de “Kumari” los hechos se harán del conocimiento de la autoridad investigadora a fin de iniciar el procedimiento que la ley imponga.</p>

<p>“Kumari” se reserva el derecho de negar un servicio y/o eliminar comentarios y/o cancelar servicios y/o productos. </p>

## <h6>12. Modificaciones a los “términos y condiciones”</h6>

<p>“Kumari” se reserva el derecho de actualizar y/o modificar y/o reemplazar parcial y/o totalmente el contenido de los “términos y condiciones” mediante su publicación en el “sitio web” por lo que el “usuario” en cualquier momento podrá revisar la versión más reciente en el “sitio web” y el único responsable de revisar y/o supervisar los cambios en los “términos y condiciones” del “sitio web” será el “usuario”.</p>

<p>El acceso al “sitio web” posterior a la actualización y/o modificación de los “términos y condiciones” constituye el consentimiento del “usuario” a vincularse a los mismos.</p>

<p>Los “términos y condiciones” no podrán ser alterados ni modificados por el “usuario” por lo que tampoco podrán renunciar parcial y/o totalmente al contenido de los mismos.</p>

## <h6>13. Vigencia </h6>

<p>Los “términos y condiciones” se mantendrán vigentes por el tiempo necesario para su ejecución y el cumplimiento de los derechos y obligaciones de las partes.</p>

## <h6>14. Notificaciones y domicilios</h6>

<p>Los avisos y/o notificaciones relacionados con el “sitio web” se deberán realizar personalmente y/o por correo electrónico a “Kumari”.</p>

<p>

Aprende con Kumari S.A. de C.V.

<br>

Blvd Ortiz Mena 1834, Los Sicomoros, Chihuahua.

</p>

<p>

Teléfonos: 625-105-67-24

<br>

contacto@kumari.mx

</p>

## <h6>15. Legislación y tribunales</h6>

<p>"Kumari" se reserva el ejercicio de cualquier acción civil y/o penal por la utilización indebida del "sitio web" por cualquier persona y/o por el incumplimiento de los "términos y condiciones". </p>

<p>En relación a las actividades realizadas dentro del "sitio web", "los usuarios" se someterán en su interpretación, ejecución, cumplimiento y aplicación de manera expresa a las leyes y a los tribunales competentes de la Ciudad de Chihuahua, Chihuahua, renunciando expresamente a cualquier fuero o jurisdicción que por razón de sus domicilios presentes y/o futuros, y/o por cualquier otra causa pudiere corresponderles a los "usuarios".</p>

## <h6>16. Dudas y aclaraciones</h6>

<p>Para dudas y aclaraciones de los "términos y condiciones" y el "aviso de privacidad", el "usuario" podrá comunicarse con "Kumari" a través del correo electrónico y/o a los teléfonos y/o al domicilio indicados en los "términos y condiciones". </p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-3">

<div class="flex justify-start items-center">

<button class="btn-back" @click="step--">



</button>

</div>



```
import { mapState, mapActions } from 'vuex'

import { required, minLength, email } from 'vuelidate/lib/validators'

export default {
  name: 'Register',

  data(){
    return{
      isPwd: true,

      step: 1,

      profile: {
        children: "",
        interest: [],
        complete_profile: false,
        role: 'user',
        check_status: true,
        background_image: 'Verde.jpg',
        photo: 'https://firebasestorage.googleapis.com/v0/b/kumari-
d465a.appspot.com/o/clases%2F1c18d9a6-dbc1-4960-975c-
fa01871bb26b.jpg?alt=media&token=4c8b718f-acfa-4581-a466-be54dc3d0ece'
      },
      intelligence1: {},
      intelligence2: {},
      intelligence3: {},
      intelligence4: {},
      intelligence5: {},
      intelligence6: {},
      intelligence7: {},
```

intelligence8: {},

interests: [

'Ciencia',

'Matemáticas',

'Robótica',

'Lectura',

'Escritura',

'Fotografía',

'Artes'

],

interests2: [

'Baile',

'Geografía',

'Programación',

'Naturaleza',

'Animales',

'Animación',

'Videojuegos'

],

myChart: null,

}

},

mounted() {

},

computed: {



```
...mapState('auth', [ 'user' ]),
```

```
suma1(){
```

```
  let total = 0
```

```
  for (const property in this.intelligence1) {
```

```
    if (this.intelligence1[property]) {
```

```
      total = total + 33.33
```

```
    }
```

```
  }
```

```
  return Math.round(total)
```

```
},
```

```
suma2(){
```

```
  let total = 0
```

```
  for (const property in this.intelligence2) {
```

```
    if (this.intelligence2[property]) {
```

```
      total = total + 33.33
```

```
    }
```

```
  }
```

```
  return Math.round(total)
```

```
},
```

```
suma3(){
```

```
  let total = 0
```

```
for (const property in this.intelligence3) {

    if (this.intelligence3[property]) {
        total = total + 33.33
    }

}

return Math.round(total)
},
suma4(){
    let total = 0

    for (const property in this.intelligence4) {

        if (this.intelligence4[property]) {
            total = total + 33.33
        }

    }

    return Math.round(total)
},
suma5(){
    let total = 0

    for (const property in this.intelligence5) {

        if (this.intelligence5[property]) {
            total = total + 33.33
        }

    }

}
```

```
    }  
    return Math.round(total)  
  },  
  suma6(){  
    let total = 0  
  
    for (const property in this.intelligence6) {  
  
      if (this.intelligence6[property]) {  
        total = total + 33.33  
      }  
  
    }  
  
    return Math.round(total)  
  },  
  suma7(){  
    let total = 0  
  
    for (const property in this.intelligence7) {  
  
      if (this.intelligence7[property]) {  
        total = total + 33.33  
      }  
  
    }  
  
    return Math.round(total)  
  },  
  suma8(){
```

```
let total = 0
```

```
for (const property in this.intelligence8) {
```

```
  if (this.intelligence8[property]) {
```

```
    total = total + 33.33
```

```
  }
```

```
}
```

```
return Math.round(total)
```

```
},
```

```
},
```

```
validations: {
```

```
  profile: {
```

```
    children: {
```

```
      required,
```

```
    },
```

```
    birthdate: {
```

```
      required,
```

```
    },
```

```
    interest: {
```

```
      required,
```

```
    },
```

```
    academy: {
```

```
      required,
```

```
    },
```

```
    name: {
```

```
      required,
```

```
    },
    direction: {
      required,
    },
    cp: {
      required,
    },
    city: {
      required,
    },
    state: {
      required,
    },
    email: {
      required,
      email
    },
    password: {
      required,
      minLength: minLength(6)
    },
  },
},
```

```
methods: {
  ...mapActions('auth', [ 'registerUser', 'closeSesion' ]),
```

```
results(){
  this.step = 14
```

```
let data = {  
  labels: [  
    'Intrapersonal',  
    'Lingüística',  
    'Interpersonal',  
    'Lógico matemática',  
    'Espacial',  
    'Cinética',  
    'Musical',  
    'Naturalista'  
  ],  
  datasets: [{  
    label: 'Mi inteligencia',  
    data: [  
      this.suma1,  
      this.suma2,  
      this.suma3,  
      this.suma4,  
      this.suma5,  
      this.suma6,  
      this.suma7,  
      this.suma8  
    ],  
    fill: true,  
    backgroundColor: 'rgba(255, 99, 132, 0.2)',  
    borderColor: 'rgb(255, 99, 132)',  
    pointBackgroundColor: 'rgb(255, 99, 132)',  
    pointBorderColor: '#fff',  
  }],  
}
```

```
        pointHoverBackgroundColor: '#fff',
        pointHoverBorderColor: 'rgb(255, 99, 132)'
    }}
};

const ctx = document.getElementById(`myChart`);
if (this.myChart) {
    this.myChart.destroy();
}

this.myChart = new Chart(ctx, {
    type: 'radar',
    data: data,
    options: {
        elements: {
            line: {
                borderWidth: 3
            }
        },
        scale: {
            ticks: {
                beginAtZero: true,
                max: 100,
                min: 0,
                stepSize: 10
            }
        }
    },
});
```

```
},
```

```
goToNext(value){
```

```
  if (value == 2) {
```

```
    if (this.$v.profile.$invalid) {
```

```
      if (!this.$v.profile.email.email) {
```

```
        this.$q.notify({
```

```
          message: 'Ingresa un correo valido.',
```

```
          color: 'negative'
```

```
        })
```

```
      return
```

```
    }
```

```
    if (!this.$v.profile.password.minLength) {
```

```
      this.$q.notify({
```

```
        message: 'La contraseña debe ser minimo de 6 caracteres.',
```

```
        color: 'negative'
```

```
      })
```

```
    return
```

```
  }
```

```
  this.$q.notify({
```

```
    message: 'Todos los campos son requeridos.',
```

```
    color: 'negative'
```

```
  })
```



```
        return
      }else{
        this.step = value
      }
    }
  }
}
```

```
if (value == 3) {
  this.step = value
}
```

```
},
```

```
async makeRegister(){
```

```
  let data = {
    profile: this.profile,
  }
}
```

```
  this.registerUser(data)
```

```
},
```

```
},
```

```
}
```

```
</script>
```

## Boot

### Auth.txt

```
// import something here
```

```
import { auth } from 'boot/firebase'
```

```
// "async" is optional;
```

// more info on params: <https://quasar.dev/quasar-cli/boot-files>

```
export default async ({ store }) => {  
  auth.onAuthStateChanged(user => {  
    if(user){  
      console.log('Este es el usuario desde auth: ', user);  
  
      // if (!user.emailVerified) {  
      //   console.log('No verificado');  
      //   store.dispatch('auth/closeSesion')  
      // }  
      store.dispatch('auth/getUser', user.uid)  
      // localStorage.setItem('loggedIn', true)  
    }  
    // else{  
    //   store.dispatch('auth/closeSesion')  
    // }  
  })  
}
```

### Firestore.txt

```
import firebase from 'firebase/app';
```

```
require('firebase/auth')  
require('firebase/firestore')  
require('firebase/storage')  
require('firebase/functions')
```

```
var firebaseConfig = {  
  apiKey: "AlzaSyClo3o_NygtY1YsY_86No_PJD8x1zVJZvQ",
```

```
    authDomain: "kumari-d465a.firebaseio.com",
    projectId: "kumari-d465a",
    storageBucket: "kumari-d465a.appspot.com",
    messagingSenderId: "569332193641",
    appId: "1:569332193641:web:e35f82245845e5cb76c5f8",
    measurementId: "G-ZXMNN30GEC"
  };
```

```
// Initialize Firebase
```

```
firebase.initializeApp(firebaseConfig);
```

```
// firebase.analytics();
```

```
const auth = firebase.auth()
```

```
const db = firebase.firestore()
```

```
const storage = firebase.storage()
```

```
const functions = firebase.functions()
```

```
export {
```

```
    firebase,
```

```
    auth,
```

```
    db,
```

```
    storage,
```

```
    functions
```

```
}
```

[Instructors](#)

[Classes.txt](#)

```
<style scoped lang="scss">
```

```
.main-container{
```

```
    background-color: white;
```

```
border-radius: 10px;  
padding: 20px;  
}
```

```
.main-title{  
font-size: 22px;  
font-weight: bold;  
}
```

```
.btn-options{  
border: none;  
margin: 5px;  
font-size: 18px;  
color: white;  
border-radius: 5px;  
cursor: pointer;  
}
```

```
.btn-modal{  
border: none;  
background-color: #24B8CB;  
color: white;  
border-radius: 15px;  
padding: 10px 15px;  
font-weight: bold;  
font-size: 17px;  
}
```

```
.form-group{
```

```
margin-top: 20px;
```

```
label{  
    font-weight: bold;  
    font-size: 18px;  
}  
}
```

```
.input-text{  
    width: 100%;  
    border: 2px solid #666666;  
    border-radius: 10px;  
    padding: 5px;  
}
```

```
.input-text:focus{  
    outline: none;  
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
}
```

```
.btn-submit{
```

```
color: white;
padding: 5px 20px;
border: none;
background-color: #24B8CB;
border-radius: 10px;
margin-left: 10px;
cursor: pointer;
}
```

```
.layer{
width: 100%;
height: 100%;
position: absolute;
top: 0;
left: 0;
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
background-color: rgba(0, 0, 0, 0.5);
color: white;
}
```

```
</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<q-table
```

```
:data="clases"
```

```

:columns="columns"

:filter="filter"

row-key="name"

flat

>

<template v-slot:top>

  <div class="flex justify-between items-center" style="width: 100%;">

    <span style="font-size: 25px; font-weight: bold;">Mis clases</span>

    <button class="btn-modal cursor-pointer" @click="modalClase = true">Crear
clase</button>

    <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">

      <template v-slot:append>

        <q-icon name="search" />

      </template>

    </q-input>

  </div>

</template>

<template v-slot:body-cell-course="props">

  <q-td :props="props">

    {{ returnCourse(props.row.course) }}

  </q-td>

</template>

<template v-slot:body-cell-video="props">

  <q-td :props="props">

    <span class="cursor-pointer" @click="modalVideo = true, claseSelect =
props.row">Ver</span>

  </q-td>

</template>

```

```

<template v-slot:body-cell-material="props">
  <q-td :props="props">
    <a target="_blank" :href="props.row.material">Ver</a>
  </q-td>
</template>

<template v-slot:body-cell-details="props">
  <q-td :props="props">
    <a target="_blank" :href="props.row.details">Ver</a>
  </q-td>
</template>

<template v-slot:body-cell-active="props">
  <q-td :props="props">
    <span v-if="props.row.active">Activo</span>
    <span v-else>No activo</span>
  </q-td>
</template>

<template v-slot:body-cell-options="props">
  <q-td :props="props">
    <button class="btn-options" style="background-color: #2CB79F;"
@click="modalEditClase = true, editClase = props.row">
      <q-icon name="edit" />
    </button>
    <button class="btn-options" style="background-color: #DC3545;"
@click="deleteClase(props.row)">
      <q-icon name="delete" />
    </button>
  </q-td>
</template>
</q-table>
</div>

```



```

<q-dialog v-model="modalClase" persistent>
  <q-card style="min-width: 800px; position:relative">

    <q-card-section class="row">
      <div class="col-12">
        <q-form
          @submit="addClase"
          class="q-gutter-md"
        >
          <div class="row">
            <div class="col-6 q-pa-md">
              <div class="form-group">
                <label for="">Clase</label> <br>
                <input class="input-text" type="text" v-model="newClase.name">
              </div>
              <div class="form-group">
                <label for="">Curso</label> <br>
                <select class="input-text" v-model="newClase.course">
                  <option :value="course.id" v-for="(course, index) in courses"
:key="index">{{ course.name }}</option>
                </select>
              </div>
            <div class="row form-group">
              <div class="col-6">
                <label for="">PDF Materiales</label>
                <input type="file" @change="getMaterial($event)"
accept="application/pdf">
              <br>
              <label for="">PDF Detalles</label>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <input type="file" @change="getDetails($event)"
accept="application/pdf">

        <br>

        <label for="">Sticker</label>

        <input type="file" @change="getSticker($event)" accept="image/*">

    </div>

</div>

</div>

<div class="col-6 q-pa-md">

    <div class="row form-group">

        <div class="col-6">

            <label for="">Banner</label>

            <input type="file" @change="getImage($event)" accept="image/*">

            <br>

            <label for="">Video</label>

            <input ref="inputVideo" type="file" @change="getVideo($event)">

        </div>

    </div>

    <div class="form-group">

        <label for="">Estatus</label>

        <select class="input-text" v-model="newClase.status">

            <option value="Activo">Activo</option>

            <option value="Pausado">Pausado</option>

        </select>

    </div>

    <div class="form-group">

        <label for="">Orden</label> <br>

        <input class="input-text" type="text" v-model="newClase.order">

```

```

        </div>
    </div>
</div>
<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalClase =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>

```

```

<div v-if="uploading" class="layer">
    <q-spinner
        color="primary"
        size="3em"
    />
    <p>Creando registro...</p>
</div>
</q-card>
</q-dialog>

```

```

<q-dialog v-model="modalEditClase" persistent v-if="editClase != null">
    <q-card style="min-width: 800px; position:relative">

        <q-card-section class="row">
            <div class="col-12">
                <q-form
                    @submit="addClaseEdit"

```

```

class="q-gutter-md"
>
<div class="row">
  <div class="col-6 q-pa-md">
    <div class="form-group">
      <label for="">Clase</label> <br>
      <input class="input-text" type="text" v-model="editClase.name">
    </div>
    <div class="form-group">
      <label for="">Curso</label> <br>
      <select class="input-text" v-model="editClase.course">
        <option :value="course.id" v-for="(course, index) in courses"
:key="index">{{ course.name }}</option>
      </select>
    </div>
    <div class="form-group">
      <label for="">Clase URL</label> <br>
      <input class="input-text" type="text" v-model="editClase.url">
    </div>
  <div class="row form-group">
    <div class="col-6">
      <label for="">PDF Materiales</label>
      <input type="file" @change="getMaterial($event)"
accept="application/pdf">
      <br>
      <label for="">PDF Detalles</label>
      <input type="file" @change="getDetails($event)"
accept="application/pdf">
      <br>
      <label for="">Sticker</label>

```

```

        <input type="file" @change="getSticker($event)" accept="image/*">
    </div>
</div>
</div>
<div class="col-6 q-pa-md">
    <div class="row form-group">
        <div class="col-6">
            <label for="">Banner</label>
            <input type="file" @change="getImage($event)" accept="image/*">
            <br>
            <label for="">Video</label>
            <input ref="inputVideo" type="file" @change="getVideo($event)">
        </div>
    </div>

    <div class="form-group">
        <label for="">Estatus</label>
        <select class="input-text" v-model="editClase.status">
            <option value="Activo">Activo</option>
            <option value="Pausado">Pausado</option>
        </select>
    </div>

    <div class="form-group">
        <label for="">Orden</label> <br>
        <input class="input-text" type="text" v-model="editClase.order">
    </div>
</div>
</div>
<div class="q-pa-md flex justify-end">

```

```
        <button class="btn-cancel" type="button" @click="modalEditClase =
false">Cerrar</button>
```

```
        <button class="btn-submit" type="submit">Guardar</button>
```

```
    </div>
```

```
  </q-form>
```

```
</div>
```

```
</q-card-section>
```

```
<div v-if="uploading" class="layer">
```

```
  <q-spinner
```

```
    color="primary"
```

```
    size="3em"
```

```
  <p>Actualizando...</p>
```

```
</div>
```

```
</q-card>
```

```
</q-dialog>
```

```
<q-dialog v-model="modalVideo" persistent>
```

```
  <q-card>
```

```
    <q-card-section>
```

```
      <div class="text-h6">{{ claseSelect.name }}</div>
```

```
    </q-card-section>
```

```
    <q-card-section class="q-pt-none">
```

```
      <video controls controlsList="nodownload" width="100%"
:src="claseSelect.url"></video>
```

```
    </q-card-section>
```

```
  <q-card-actions align="right">
```

```

        <q-btn flat label="Cerrar" color="primary" v-close-popup />
      </q-card-actions>
    </q-card>
  </q-dialog>

  <ProgressComponent :upload="upload" :progress="progress" :progress-label="progressLabel"
:file="claseVideo" />
</q-page>
</template>

<script>
//Firebase
import { firebase, db, storage } from 'boot/firebase'
const ref = storage.ref()
import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import VueTagsInput from '@johmun/vue-tags-input';
import ProgressComponent from './ProgressComponent'

export default {
  name: 'feed',

  components: {
    VueTagsInput,
    ProgressComponent
  },

  data(){
    return{

```

uploading: false,

newClase: {

name: "",

photo: "",

status: null,

course: "",

video: "",

order: 1,

url: "",

},

courses: [],

modalVideo: false,

modalClase: false,

modalEditClase: false,

editClase: null,

claseImage: null,

claseVideo: null,

claseMaterial: null,

claseDetails: null,

claseSticker: null,

claseSelect: {},

clases: [],

filter: "",



```

columns: [
  {
    name: 'name',
    required: true,
    label: 'Nombre',
    align: 'left',
    field: row => row.name,
    format: val => `${val}`,
    sortable: true
  },
  { name: 'course', align: 'left', label: 'Curso', field: 'course', sortable: true },
  { name: 'video', align: 'left', label: 'Video', field: 'video', sortable: true },
  { name: 'material', align: 'left', label: 'Material', field: 'material', sortable: true },
  { name: 'details', align: 'left', label: 'Detalles', field: 'details', sortable: true },
  { name: 'status', align: 'left', label: 'Estatus', field: 'status' },
  { name: 'options', label: 'Opciones', field: 'options' }
],

upload: false,
progress: null,
progressLabel: "",
}
},

```

```

filters: {
  formatDate(args){
    try {
      let date = moment(args.toDate()).fromNow();
      return date
    }
  }
}

```

```
    } catch (error) {  
      let date = moment(args).fromNow();  
      return date  
    }  
  }  
},
```

```
created() {  
  this.getClases()  
  this.getCourses()  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ])  
},
```

```
methods: {  
  truncateString(str, num) {  
    if (str.length <= num) {  
      return str  
    }  
  
    return str.slice(0, num) + '...'  
  },
```

```
  async addClase(){  
    try {  
      this.$q.dialog({  
        title: 'Agregar nuevo clase',
```

```
message: '¿Quieres registrar este clase?',
cancel: true,
persistent: true
}).onOk(async () => {
  this.uploading = true

  let url = null
  let materialUrl = null
  let detailsUrl = null
  let stickerUrl = null

  if (this.claseImage == null) {
    url = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-jihan-cover-680x384.jpg'
  }else{
    url = await this.uploadImage(this.claseImage)
    this.claseImage = null
  }

  if (this.claseMaterial == null) {
    materialUrl = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-jihan-cover-680x384.jpg'
  }else{
    materialUrl = await this.uploadPDF(this.claseMaterial)
    this.claseMaterial = null
  }

  if (this.claseDetails == null) {
    detailsUrl = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-jihan-cover-680x384.jpg'
```

```

    }else{

        detailsUrl = await this.uploadPDF(this.claseDetails)

        this.claseDetails = null

    }

    if (this.claseSticker == null) {

        stickerUrl =
'http://pm1.narvii.com/7584/8ef97fd2da9ef17bc680cb3691cc767d079873f3r1-574-569v2_uhq.jpg'

    }else{

        stickerUrl = await this.uploadImage(this.claseSticker)

        this.claseSticker = null

    }

    this.newClase.instructor = this.user.uid

    this.newClase.photo = url

    this.newClase.material = materialUrl

    this.newClase.details = detailsUrl

    this.newClase.sticker = stickerUrl

    if (this.newClase.url == '') {

        this.newClase.url = 'https://firebasestorage.googleapis.com/v0/b/kumari-
d465a.appspot.com/o/videos%2FSigrid%20-
%20Burning%20Bridges.mp4?alt=media&token=20dabddb-3494-403b-8040-99d210bd5052'

    }

    this.newClase.created_at = moment().format()

    this.newClase.created_atUnix = moment(this.newClase.created_at).unix()

```

```
let reponse = await db.collection('classes')
    .add(this.newClase)
    .then(query => {
        this.newClase.id = query.id
        let course = JSON.parse(JSON.stringify(this.newClase))

        this.clases.push(course)

        this.claseImage = null

        this.modalClase = false

        this.newClase = {
            name: "",
            photo: "",
            status: null,
            course: "",
        }

        this.uploading = false
    })
this.$refs.inputVideo.value = ""
this.$q.notify({
    message: 'Clase registrado',
    color: 'positive'
})
})

} catch (error) {
```

```
        console.log(error);
        this.uploading = false
    }
},
```

```
returnCourse(id){
    let found = this.courses.find(doc => doc.id == id)
```

```
    if (found) {
        return found.name
    }
```

```
},
```

```
async addClaseEdit(){
```

```
    try {
        this.uploading = true
```

```
        if (this.claseImage == null) {
            this.editClase.photo = this.editClase.photo
```

```
        }else{
            let url = await this.uploadImage(this.claseImage)
            this.claseImage = null
            this.editClase.photo = url
        }
```

```
        if (this.claseMaterial == null) {
            this.editClase.material = this.editClase.material
        }else{
```

```
    let materialUrl = await this.uploadPDF(this.claseMaterial)

    this.claseMaterial == null

    this.editClase.material = materialUrl
  }
```

```
if (this.claseDetails == null) {

  this.editClase.details = this.editClase.details
}else{

  let detailsURL = await this.uploadPDF(this.claseDetails)

  this.claseDetails == null

  this.editClase.details = detailsURL
}
```

```
if (this.claseSticker == null) {

  this.editClase.sticker = this.editClase.sticker
}else{

  let stickerUrl = await this.uploadImage(this.claseSticker)

  this.claseSticker == null

  this.editClase.sticker = stickerUrl
}
```

```
let response = await db.collection('clases')

    .doc(this.editClase.id)

    .update(this.editClase)
```

```
this.$q.notify({

  message: 'Clase editado',

  color: 'positive'

})
```

```

    this.$refs.inputVideo.value = ""

    this.modalEditClase = false

    this.uploading = false
  } catch (error) {
    console.log(error);
    this.uploading = false
  }
},

async getCourses(){
  this.courses = []

  try {
    let response = await db.collection('courses')
      .where('instructor', '==', this.user.uid)
      .get()
      .then(query => {
        query.forEach(async doc => {
          let course = doc.data()
          course.id = doc.id

          this.courses.push(course)
        })
      })
  } catch (error) {
    console.log(error);
  }
},

```



```

async getClases(){
  this.clases = []

  try {
    let response = await db.collection('clases')
      .where('instructor', '==', this.user.uid)
      .get()
      .then(query => {
        query.forEach(async doc => {
          let clase = doc.data()
          clase.id = doc.id

          this.clases.push(clase)
        })
      })
  } catch (error) {
    console.log(error);
  }
},

async deleteClase(clase){
  try {
    this.$q.dialog({
      title: 'Eliminar clase',
      message: '¿Estas seguro de eliminar este clase?',
      cancel: true,
      persistent: true
    }).onOk(async () => {
      let index = this.clases.findIndex(doc => doc.id == clase.id)

```

```

this.$q.notify({
  message: 'Clase eliminado',
  color: 'positive'
})

this.clases.splice(index, 1)
})

} catch (error) {
  console.log(error);
}
},

//Generar UUID
generateUUID(){
  var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
  var k = ['x','x','x','x','x','x','x','x','x','-','x','x','x','x','x','-','4','x','x','x','x','-','y','x','x','x','x','-','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];
  var u = "",i=0,rb=Math.random()*0xffffffff|0;
  while(i++<36) {
    var c=k[i-1],r=rb&0xf,v=c=='x'?r:(r&0x3|0x8);
    u+=(c=='-'||c=='4')?c:h[v];rb=i%8==0?Math.random()*0xffffffff|0:rb>>4
  }
  return u
},

async uploadVideo(file){

```

```

this.upload = true

let ext = file.name.substring(file.name.lastIndexOf('.')+1, file.name.length) || file.name

try {
  let metadata = {
    contentType: file.type,
    contentLanguage: "es",
  }

  let fileId = this.generateUUID()

  const refFile = ref.child('vid/' + fileId + '.' + ext)

  var task = refFile.put(file, metadata);

  task.on('state_changed',
    (snapshot) => {
      // Observe state change events such as progress, pause, and resume

      // Get task progress, including the number of bytes uploaded and the total number
of bytes to be uploaded

      this.progress = ((snapshot.bytesTransferred / snapshot.totalBytes) * 100) / 100;

      this.progressLabel = Math.trunc((this.progress * 100)) + '%'

      console.log('Upload is ' + (this.progress * 100) + '% done');

      switch (snapshot.state) {

        case firebase.storage.TaskState.PAUSED: // or 'paused'
          console.log('Upload is paused');

          break;

        case firebase.storage.TaskState.RUNNING: // or 'running'
          console.log('Upload is running');

          break;

      }
    },
    (error) => {
      // Handle unsuccessful uploads

```

```

    },
    () => {

        // Handle successful uploads on complete
        // For instance, get the download URL: https://firebasestorage.googleapis.com/...
        task.snapshot.ref.getDownloadURL().then(async (downloadURL) => {

            console.log('File available at', downloadURL);

            this.newClase.url = downloadURL

            if (this.editClase != null) {
                this.editClase.url = downloadURL
            }
            this.upload = false
        });
    }
    );
} catch (error) {
    console.log(error);
}
},

```

```

async uploadImage(image){

```

```

    try {
        let metadata = {
            contentType: "image/jpeg",
            contentLanguage: "es",

```

```

    }

    let fotold = this.generateUUID()

    const reflmg = ref.child('clases/' + fotold + '.jpg')

    return reflmg.put(image, metadata)

    .then(e => {

        console.log(e)

        return ref.child('clases/' + fotold + '.jpg').getDownloadURL()

        .then(async (url) => {

            return url

        })

    })

    .catch(error => console.log(error))

} catch (error) {

    console.log(error);

}

},

```

```

async uploadPDF(data){

    try {

        let metadata = {

            contentType: "application/pdf",

            contentLanguage: "es",

        }

        let pdfId = this.generateUUID()

        const reflmg = ref.child('PDFs/' + pdfId + '.pdf')

        return reflmg.put(data, metadata)

    }

```

```

        .then(e => {
            console.log(e)
            return ref.child('PDFs/' + pdfId + '.pdf').getDownloadURL()
        })
        .then(async (url) => {
            return url
        })
    })
    .catch(error => console.log(error))
} catch (error) {
    console.log(error);
}
},

```

```

getMaterial(e){
    this.claseMaterial = e.target.files[0]
    console.log(this.claseMaterial);
},

```

```

getDetails(e){
    this.claseDetails = e.target.files[0]
    console.log(this.claseDetails);
},

```

```

getSticker(e){
    this.claseSticker = e.target.files[0]
    console.log(this.claseSticker);
},

```

```
getImage(e){
    this.claseImage = e.target.files[0]
    console.log(this.claseImage);
},

getVideo(e){
    this.claseVideo = e.target.files[0]
    console.log(this.claseVideo);

    this.uploadVideo(this.claseVideo)
},

},
}
```

</script>

### Courses.txt

<style scoped lang="scss">

```
.main-container{
    background-color: white;
    border-radius: 10px;
    padding: 20px;
}
```

```
.main-title{
    font-size: 22px;
    font-weight: bold;
}
```

```
.btn-options{  
    border: none;  
    margin: 5px;  
    font-size: 18px;  
    color: white;  
    border-radius: 5px;  
    cursor: pointer;  
}
```

```
.btn-modal{  
    border: none;  
    background-color: #24B8CB;  
    color: white;  
    border-radius: 15px;  
    padding: 10px 15px;  
    font-weight: bold;  
    font-size: 17px;  
}
```

```
.form-group{  
    margin-top: 20px;  
  
    label{  
        font-weight: bold;  
        font-size: 18px;  
    }  
}
```

```
.input-text{
```



```
width: 100%;  
border: 2px solid #666666;  
border-radius: 10px;  
padding: 5px;  
}
```

```
.input-text:focus{  
    outline: none;  
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;  
}
```

```

.badge-color{
  width: 20px;
  height: 20px;
  border-radius: 50%;
  margin-right: 10px;
  cursor: pointer;

}

.active{
  box-shadow: 1px 1px 10px rgba(0, 0, 0, 0.8);
}
</style>

<template>
  <q-page>
    <div class="main-container">
      <q-table
        title="Mis Cursos"
        :data="courses"
        :columns="columns"
        :filter="filter"
        row-key="name"
        flat
      >
        <template v-slot:top>
          <div class="flex justify-between items-center" style="width: 100%;">
            <span style="font-size: 25px; font-weight: bold;">Mis cursos</span>

```

```
        <button class="btn-modal cursor-pointer" @click="modalCourse = true">Crear
curso</button>
```

```
        <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">
```

```
            <template v-slot:append>
```

```
                <q-icon name="search" />
```

```
            </template>
```

```
        </q-input>
```

```
    </div>
```

```
</template>
```

```
<template v-slot:body-cell-clases="props">
```

```
    <q-td :props="props">
```

```
        {{ props.row.clases.length }}
```

```
    </q-td>
```

```
</template>
```

```
<template v-slot:body-cell-status="props">
```

```
    <q-td :props="props">
```

```
        <span v-if="props.row.status">Activo</span>
```

```
        <span v-else>No activo</span>
```

```
    </q-td>
```

```
</template>
```

```
<template v-slot:body-cell-options="props">
```

```
    <q-td :props="props">
```

```
        <button class="btn-options" style="background-color: #55B995;"
@click="modalEditCourse = true, editCourse = props.row">
```

```
            <q-icon name="edit" />
```

```
        </button>
```

```

        <button class="btn-options" style="background-color: #F3920B;"
@click="deleteCourse(props.row)">
            <q-icon name="delete" />
        </button>
    </q-td>
</template>
</q-table>
</div>

```

```

<q-dialog v-model="modalCourse" persistent>
    <q-card style="min-width: 800px;">
        <q-card-section>
            <p class="main-title">Agregar nuevo curso</p>
        </q-card-section>
        <q-card-section class="row">
            <div class="col-12">
                <q-form
                    @submit="addCourse"
                    class="q-gutter-md"
                >
                    <div class="row">
                        <div class="col-6 q-pa-md">
                            <div class="form-group">
                                <label for="">Curso</label> <br>
                                <input class="input-text" type="text" v-model="newCourse.name">
                            </div>
                            <div class="form-group">
                                <label for="">Descripción</label> <br>
                                <textarea class="input-text" v-model="newCourse.description"></textarea>

```

```

</div>

<div class="form-group">
  <vue-tags-input
    placeholder="Edades"
    v-model="tagAge"
    :tags="newCourse.age"
    @tags-changed="newTagsAge => newCourse.age = newTagsAge"
  />
</div>

```

```

<div>
  <vue-tags-input
    placeholder="Palabras clave"
    v-model="tag"
    :tags="newCourse.clave_words"
    @tags-changed="newTags => newCourse.clave_words = newTags"
  />
</div>

```

```

</div>

<div class="col-6 q-pa-md">
  <div class="form-group">
    <input type="file" @change="getImage($event)" accept="image/">
  </div>

  <div class="form-group">
    <label for="">Color de Acento</label>

    <div class="flex">
      <div class="badge-color" @click="newCourse.color = item" :class="[
newCourse.color == item ? 'active' : '' ]" :style="{ backgroundColor: item }" v-for="(item, index) in
palette" :key="index">

```

```

        </div>
    </div>
</div>
<div class="form-group">
    <label for="">Estatus</label>
    <div>
        <select class="input-text" v-model="newCourse.status">
            <option value="Activo">Activo</option>
            <option value="Pausado">Pausado</option>
        </select>
    </div>
</div>
<div class="form-group">
    <label for="">Precio</label>
    <div>
        <input type="number" v-model="newCourse.price" class="input-text">
    </div>
</div>
</div>
</div>
<div class="q-pa-md flex justify-end">
    <button class="btn-cancel" type="button" @click="modalCourse =
false">Cerrar</button>
    <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>

```

```
</q-card>
```

```
</q-dialog>
```

```
<q-dialog v-model="modalEditCourse" persistent v-if="editCourse != null">
```

```
<q-card style="min-width: 800px;">
```

```
<q-card-section>
```

```
<p class="main-title">Editar curso</p>
```

```
</q-card-section>
```

```
<q-card-section class="row">
```

```
<div class="col-12">
```

```
<q-form
```

```
@submit="addCourseEdit"
```

```
class="q-gutter-md"
```

```
>
```

```
<div class="row">
```

```
<div class="col-6 q-pa-md">
```

```
<div class="form-group">
```

```
<label for="">Curso</label> <br>
```

```
<input class="input-text" type="text" v-model="editCourse.name">
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Descripción</label> <br>
```

```
<textarea class="input-text" v-model="editCourse.description"></textarea>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="">Edades</label> <br>
```

```
<vue-tags-input
```

```
placeholder="Palabras clave"
```

```
v-model="tag"
```

```

      :tags="editCourse.age"

      @tags-changed="newTags => editCourse.age = newTags"

    />

  </div>

  <div>
    <vue-tags-input
      placeholder="Palabras clave"
      v-model="tag"
      :tags="editCourse.clave_words"
      @tags-changed="newTags => editCourse.clave_words = newTags"
    />
  </div>

</div>

<div class="col-6 q-pa-md">
  <div class="form-group">
    <input type="file" @change="getImage($event)" accept="image/">
  </div>
  <div class="form-group">
    <label for="">Color de Acento</label>
    <div class="flex">
      <div class="badge-color" @click="editCourse.color = item" :class="[
editCourse.color == item ? 'active' : '' ]" :style="{ backgroundColor: item }" v-for="(item, index) in
palette" :key="index">

    </div>
  </div>
</div>

```



```

<div class="form-group">
  <label for="">Estatus</label>
  <div>
    <select class="input-text" v-model="editCourse.status">
      <option value="Activo">Activo</option>
      <option value="Pausado">Pausado</option>
    </select>
  </div>
</div>
<div class="form-group">
  <label for="">Precio</label>
  <div>
    <input type="number" v-model="editCourse.price" class="input-text">
  </div>
</div>
</div>
<div class="q-pa-md flex justify-end">
  <button class="btn-cancel" type="button" @click="modalEditCourse =
false">Cerrar</button>
  <button class="btn-submit" type="submit">Guardar</button>
</div>
</q-form>
</div>
</q-card-section>
</q-card>
</q-dialog>
</q-page>
</template>

```

```
<script>

import { db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'

import moment from 'moment-timezone'

import VueTagsInput from '@johmun/vue-tags-input';

export default {
  name: 'feed',

  components: {
    VueTagsInput,
  },

  data(){
    return{
      newCourse: {
        name: "",
        age: [],
        clave_words: [],
        photo: "",
        color: "",
        status: null,
      },

      palette: [
        '#019a9d', '#d9b801', '#e8045a', '#b2028a',
        '#2a0449',
```

```
],

tag: "",
tagAge: "",
modalCourse: false,
modalEditCourse: false,
editCourse: null,

courseImage: null,

courses: [],
filter: "",
columns: [
  {
    name: 'name',
    required: true,
    label: 'Nombre',
    align: 'left',
    field: row => row.name,
    format: val => `${val}`,
    sortable: true
  },
  { name: 'actives', align: 'left', label: 'Activos', field: 'actives', sortable: true },
  { name: 'clases', align: 'left', label: 'Clases', field: 'clases', sortable: true },
  { name: 'status', align: 'left', label: 'Estatus', field: 'status' },
  { name: 'options', label: 'Opciones', field: 'options' }
],
}
},
```

```
filters: {  
  formatDate(args){  
    try {  
      let date = moment(args.toDate()).fromNow();  
      return date  
    } catch (error) {  
      let date = moment(args).fromNow();  
      return date  
    }  
  }  
},
```

```
created() {  
  this.getCourses()  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ])  
},
```

```
methods: {  
  truncateString(str, num) {  
    if (str.length <= num) {  
      return str  
    }  
  
    return str.slice(0, num) + '...'  
  },
```

```

async addCourse(){
  try {
    this.$q.dialog({
      title: 'Agregar nuevo curso',
      message: '¿Quieres registrar este curso?',
      cancel: true,
      persistent: true
    }).onOk(async () => {
      let url = null
      if (this.courseImage == null) {
        url = 'https://thumbnails.kpopmap.com/2020/07/sm-wooah-nana-weeekly-jihan-cover-680x384.jpg'
      }else{
        url = await this.uploadImage()
        this.courseImage = null
      }
      this.newCourse.photo = url
      this.newCourse.instructor = this.user.uid
      this.newCourse.price = 500

      this.newCourse.created_at = moment().format()
      this.newCourse.price = 100

      this.newCourse.created_atUnix = moment(this.newCourse.created_at).unix()

      let reponse = await db.collection('courses')
        .add(this.newCourse)
        .then(query => {

```

```
        this.newCourse.id = query.id

        let course = JSON.parse(JSON.stringify(this.newCourse))

        course.clases = []

        this.courses.push(course)

        this.courseImage = null

        this.modalCourse = false

        this.newCourse = {
            name: "",
            age: "",
            clave_words: [],
            photo: "",
            color: "",
            status: null,
        }
    })

    this.$q.notify({
        message: 'Curso registrado',
        color: 'positive'
    })
})

} catch (error) {
    console.log(error);
}
```

```
},
```

```
async addCourseEdit(){
```

```
  try {
```

```
    if (this.courseImage == null) {
```

```
      this.editCourse.photo = this.editCourse.photo
```

```
    }else{
```

```
      let url = await this.uploadImage()
```

```
      this.courseImage = null
```

```
      this.editCourse.photo = url
```

```
    }
```

```
    let response = await db.collection('courses')
```

```
      .doc(this.editCourse.id)
```

```
      .update(this.editCourse)
```

```
    this.modalEditCourse = false
```

```
    this.$q.notify({
```

```
      message: 'Curso editado',
```

```
      color: 'positive'
```

```
    })
```

```
  } catch (error) {
```

```
    console.log(error);
```

```
  }
```

```
},
```

```
async getCourses(){
```

```
  this.courses = []
```

```

try {
  let response = await db.collection('courses')
    .where('instructor', '==', this.user.uid)
    .get()
    .then(query => {
      query.forEach(async doc => {
        let course = doc.data()
        course.id = doc.id
        course.clases = []

        let clases = await db.collection('clases')
          .where('course', '==', doc.id)
          .get()
          .then(query => {
            query.forEach(doc => {
              let clase = doc.data()
              clase.id = doc.id

              course.clases.push(clase)
            })
          })

        this.courses.push(course)
      })
    })
} catch (error) {
  console.log(error);
}

```



```

    }
  },

  async deleteCourse(course){
    try {
      this.$q.dialog({
        title: 'Eliminar curso',
        message: '¿Estas seguro de eliminar este curso?',
        cancel: true,
        persistent: true
      }).onOk(async () => {
        let index = this.courses.findIndex(doc => doc.id == course.id)
        let response = await db.collection('courses').doc(course.id).delete()

        this.$q.notify({
          message: 'Curso eliminado',
          color: 'positive'
        })

        this.courses.splice(index, 1)
      })

    } catch (error) {
      console.log(error);
    }
  },

  //Generar UUID
  generateUUID(){

```



```

    })
    .catch(error => console.log(error))
  } catch (error) {
    console.log(error);
  }
},

getImage(e){
  this.courserImage = e.target.files[0]
  console.log(this.courserImage);
},

},
}
</script>

```

## Progresscomponent.txt

```

<template>

  <!-- Upload progress -->

  <q-dialog v-model="upload" persistent>

    <q-card style="width: 400px;">

      <q-card-section class="q-pt-none text-center">

        <p style="font-size: 18px; padding: 10px;" v-if="file">{{ truncateString(file.name, 12)
      }}</p>

      </q-card-section>

      <q-card-section>

        <q-linear-progress size="lg" :value="progress" color="#00CA72">

          <div class="absolute-full flex flex-center">

            <q-badge color="white" text-color="accent" :label="progressLabel" />

          </div>

        </q-linear-progress>

```

```

        </q-card-section>
    </q-card>
</q-dialog>
</template>

<script>
export default {
  props: [
    'file',
    'progress',
    'progressLabel',
    'upload'
  ],
  data(){
    return{
      show: false,
    }
  },
  created(){
    this.show = this.upload
  },
  methods: {
    truncateString(str, num) {
      if (str.length <= num) {
        return str
      }
      return str.slice(0, num) + '...'
    },
  },
}
```

```
}
```

```
</script>
```

```
<style>
```

```
</style>
```

## Layouts

### Mainlayout.txt

```
<style scoped lang="scss">
```

```
.profile{
```

```
  width: 100%;
```

```
  height: 100px;
```

```
  background-color: #1976D2;
```

```
  display: flex;
```

```
  justify-content: flex-start;
```

```
  align-items: center;
```

```
}
```

```
.profile{
```

```
  display: flex;
```

```
  justify-content: flex-start;
```

```
  align-items: center;
```

```
  padding-left: 10px;
```

```
  color: white;
```

```
}
```

```
.profile div span:nth-child(1){
```

```
  font-weight: bold;
```

```
  font-size: 20px;
```

```
}
```

```
.img-container{  
  width: 50px;  
  height: 50px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
.img-container img{  
  width: 50px;  
  height: 50px;  
  border-radius: 50%;  
  object-fit: cover;  
  object-position: center;  
}
```

```
.content{  
  background-color: #E6E6E6;  
}
```

```
.menu-links{  
  display: flex;  
  justify-content: space-around;  
  align-items: center;  
  flex-direction: column;  
  color: white;  
  min-height: 100vh;  
  background-position: center;
```

```
background-size: cover;  
background-repeat: no-repeat;  
background-color: rgba(0, 0, 0, 0.8);
```

```
.links{  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;  
  
    a{  
        text-align: center;  
        padding: 10px 20px;  
        min-width: 150px;  
        text-decoration: none;  
        color: white;  
        border-radius: 15px;  
        font-size: 22px;  
  
        &:hover{  
            background-color: yellow;  
            color: black;  
        }  
    }  
}
```

```
.close-session{  
    padding: 5px 15px;  
    min-width: 150px;
```

```
        text-align: center;

        border-radius: 15px;

        font-size: 20px;

        background-color: gray;

        cursor: pointer;
    }
}
```

```
header{

    background-position: center;

    background-repeat: repeat;

    background-color: rgba(0, 0, 0, 0.8);

    display: none;

}
```

```
@media (max-width: 1023px) {

    header{

        display: block;

    }

}

</style>
```

```
<template>

    <q-layout view="IHh Lpr IFf">

        <q-header elevated :style="{ 'background-image': 'url(/bg/sidebar/' + user.background_image
+ ')' }">

            <q-toolbar>

                <q-btn

                    flat
```



```

        dense
        round
        icon="menu"
        aria-label="Menu"
        @click="leftDrawerOpen = !leftDrawerOpen"
    />

</q-toolbar>
</q-header>

<q-drawer
    v-model="leftDrawerOpen"
    show-if-above
    class="side"
    >
    <div class="menu-links" :style="{ 'background-image': 'url(/bg/sidebar/' +
user.background_image + ')' }">
        

        <div class="links">
            <router-link class="link" :to="link.link" v-for="(link, index) in essentialLinks" v-
if="link.roles.includes(user.role)" :key="index">
                {{ link.title }}
            </router-link>
            <!-- <router-link class="link" to="/market">
                Market
            </router-link> -->
            <a href="https://app.kumari.mx" target="_blank" v-if="user.role == 'user'">
                Ir a la APP
            </a>

```

```

        </div>

        <span class="close-session" @click="logout">

            Salir

        </span>

    </div>

</q-drawer>

<q-page-container class="content">

    <div style="padding: 20px;">

        <router-view />

    </div>

</q-page-container>

</q-layout>

</template>

<script>

import EssentialLink from 'components/EssentialLink.vue'

import { db } from 'boot/firebase'

import { mapState, mapActions } from 'vuex'

const linksData = [

    {

        title: 'Dashboard',

        icon: 'dashboard',

        link: '/inicio',

        roles: ['admin']

    },

    {

```

```
    title: 'Inicio',
    icon: 'dashboard',
    link: '/dashboard',
    roles: ['admin']
  },
  {
    title: 'Usuarios',
    icon: 'dashboard',
    link: '/users',
    roles: ['admin']
  },
  {
    title: 'Pedidos',
    icon: 'dashboard',
    link: '/orders',
    roles: ['admin']
  },
  {
    title: 'Instructores',
    icon: 'dashboard',
    link: '/instructors',
    roles: ['admin']
  },
  {
    title: 'Cursos',
    icon: 'dashboard',
    link: '/all-courses',
    roles: ['admin']
  },
}
```

```
{
  title: 'Clases',
  icon: 'dashboard',
  link: '/all-lessons',
  roles: ['admin']
},
{
  title: 'Ayuda para usuarios',
  icon: 'dashboard',
  link: '/questions',
  roles: ['admin']
},
// {
//   title: 'Market',
//   icon: 'person',
//   link: '/market',
//   roles: ['admin']
// },
{
  title: 'Mis Cursos',
  icon: 'person',
  link: '/my-courses',
  roles: ['instructor']
},
{
  title: 'Mis Clases',
  icon: 'person',
  link: '/my-clases',
  roles: ['instructor']
}
```

```
},  
{  
  title: 'Mi Cuenta',  
  icon: 'person',  
  link: '/my-account',  
  roles: ['user']  
},  
{  
  title: 'Actividad',  
  icon: 'person',  
  link: '/my-activity',  
  roles: ['user']  
},  
{  
  title: 'Test',  
  icon: 'text_snippet',  
  link: '/test',  
  roles: ['user']  
},  
{  
  title: 'Recibos',  
  icon: 'person',  
  link: '/receipts',  
  roles: ['user']  
},  
{  
  title: 'Preguntas frecuentes',  
  icon: 'person',  
  link: '/frequent-questions',
```

```
      roles: ['user']
    },
  ];

export default {
  name: 'MainLayout',

  components: { EssentialLink },

  computed: {
    ...mapState('auth', [ 'user' ])
  },

  data () {
    return {
      leftDrawerOpen: false,
      essentialLinks: linksData
    }
  },

  methods: {
    ...mapActions('auth', [ 'closeSesion' ]),

    logOut(){
      this.closeSesion()
    }
  }
}
```

</script>

## Market

### Firstcollection.txt

```
<template>

  <!-- Upload progress -->

  <q-dialog v-model="upload" persistent>

    <q-card style="width: 400px;">

      <q-card-section class="q-pt-none text-center">

        <p style="font-size: 18px; padding: 10px;" v-if="file">{{ truncateString(file.name, 12)
}}</p>

      </q-card-section>

      <q-card-section>

        <q-linear-progress size="lg" :value="progress" color="#00CA72">

          <div class="absolute-full flex flex-center">

            <q-badge color="white" text-color="accent" :label="progressLabel" />

          </div>

        </q-linear-progress>

      </q-card-section>

    </q-card>

  </q-dialog>

</template>
```

```
<script>

export default {

  props: [

    'file',

    'progress',

    'progressLabel',

    'upload'

  ],

  data(){
```

```
        return{
            show: false,
        }
    },
    created(){
        this.show = this.upload
    },
    methods: {
        truncateString(str, num) {
            if (str.length <= num) {
                return str
            }
            return str.slice(0, num) + '...'
        },
    },
}
</script>
```

<style>

</style>

[Market.txt](#)

<style scoped lang="scss">

```
.main-container{
    background-color: white;
    border-radius: 10px;
    padding: 20px;

    button{
        background-color: #24B8CB;
```



```
    color: white;

    border-radius: 10px;

    padding: 8px 20px;

    border: none;

    margin-top: 20px;

    cursor: pointer;

}

}
```

```
.course{

    padding: 5px;


    img{

        width: 100%;

    }

}
```

```
.btn-cancel{

    background-color: red;

    color: white;

    border-radius: 10px;

    padding: 8px 20px;

    border: none;

    margin-top: 20px;

    cursor: pointer;

}
```

```
.btn-submit{

    background-color: #24B8CB;
```

```
color: white;
border-radius: 10px;
padding: 8px 20px;
border: none;
margin-top: 20px;
cursor: pointer;
margin-left: 10px;
}
```

```
.course-container-body{
border-radius: 10px;
aspect-ratio: 16/9;
display: flex;
justify-content: center;
align-items: center;
position: relative;
```

```
img{
width: 80%;
}
```

```
.layer-course{
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
```

```
}  
}
```

```
.course-container{
```

```
padding: 10px;
```

```
cursor: pointer;
```

```
.course-body{
```

```
background-color: white;
```

```
border-radius: 10px;
```

```
position: relative;
```

```
aspect-ratio: 16/9;
```

```
object-fit: cover;
```

```
object-position: center;
```

```
img{
```

```
width: 100%;
```

```
height: 100%;
```

```
object-fit: cover;
```

```
object-position: center;
```

```
border-radius: 10px;
```

```
}
```

```
.layer{
```

```
position: absolute;
```

```
top: 0;
```

```
left: 0;
```

```
width: 100%;
```

```
height: 100%;  
border-radius: 10px;  
background-color: rgba(0, 0, 0, 0.1);
```

```
&.active{  
    background-color: rgba(0, 0, 0, 0.7);  
}  
}  
}
```

```
.name-title{  
    text-align: center;  
    font-size: 22px;  
    font-weight: bold;  
}  
}
```

```
.buyed{  
    background-color: red;  
    color: white;  
    padding: 5px 10px;  
    border-radius: 5px;  
    position: absolute;  
    top: 0;  
    left: 0;  
    margin: 10px;  
}
```

```
.layer-modal{
```

```
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.5);
z-index: 10000;
display: flex;
justify-content: center;
align-items: center;
}
```

```
.all-courses{
  img{
    width: 100%;
    height: 100%;
    object-fit: cover;
    object-position: center;
    border-radius: 10px;
  }
}
```

</style>

<template>

<q-page>

<div class="main-container">

<div class="row">

<div class="col-12 col-sm-6 col-md-6 col-lg-4 course" v-if="allCourses != null">

<div class="course-container">

```
<div class="course-container-body all-courses" :style="{backgroundColor: 'purple'}">
```

```
    
```

```
    <div class="layer-course"></div>
```

```
</div>
```

```
</div>
```

```
<p style="font-weight: bold; font-size: 22px;">{{ allCourses.name }}</p>
```

```
<div class="flex justify-end items-center">
```

```
    <span style="font-weight: bold;">Incluye {{ allCoursesFilter.length }} cursos</span>
```

```
</div>
```

```
<p>{{ allCourses.description }}</p>
```

```
<div>
```

```
    <span style="font-size: 20px;">${{ allCourses.price }}</span>
```

```
</div>
```

```
<button @click="buyAllCourses">Comprar</button>
```

```
</div>
```

```
<div class="col-12 col-sm-6 col-md-6 col-lg-4 course" v-for="(course, index) in  
courses_market" :key="course.id">
```

```
    <div class="course-container">
```

```
        <div class="course-container-body" :style="{backgroundColor: course.color}">
```

```
            
```

```
            <div class="layer-course"></div>
```

```
                <span class="bought" v-if="getIt(course.id)">Comprado</span>
```

```
            </div>
```

```
        </div>
```

```

<p style="font-weight: bold; font-size: 22px;">{{ course.name }}</p>

<div class="flex justify-end items-center">
  <span style="font-weight: bold;">{{ course.clases.length }} clases</span>
</div>

<p>{{ course.description }}</p>

<div>
  <span style="font-size: 20px;">${{ course.price }}</span>
</div>

  <button :disabled="getIt(course.id)" @click="buyCourse(course)">Comprar</button>
</div>
</div>
</div>

<q-dialog v-model="modalBuy" persistent v-if="course != null">
  <q-card style="min-width: 600px;">
    <q-card-section class="row">
      <div class="col-12">

        <div class="row">
          <div class="col-12">
            <div class="course-container">
              <div class="course-container-body" :style="{backgroundColor: course.color}">

                

                <div class="layer-course"></div>
              </div>
            </div>
          </div>
          <p style="font-weight: bold; font-size: 22px;">{{ course.name }}</p>

```

```

        <div class="flex justify-end items-center">
            <span style="font-weight: bold;">{{ course.clases.length }} clases</span>
        </div>
        <p>{{ course.description }}</p>
        <div>
            <span>${{ course.price }}</span>
        </div>
    </div>
    <div class="col-12 q-mt-md">
        <stripe-element-card
            ref="elementRef"
            :pk="key"
            @token="tokenCreated"
        />
    </div>
</div>
<div class="q-pa-md">
    <button class="btn-cancel" type="button" @click="modalBuy =
false">Cancelar</button>
    <button class="btn-submit" @click="submit">Comprar</button>
</div>

</div>
</q-card-section>
</q-card>
</q-dialog>

<q-dialog v-model="modalBuyAllCourses" v-if="allCourses != null">
    <q-card style="min-width: 600px; position: relative;">

```



```

<q-card-section class="row">

  <div class="col-12">

    <div class="row">

      <div class="col-12">

        <div class="course-container">

          <div class="course-container-body all-courses" :style="{backgroundColor:
'purple'}">

            

            <div class="layer-course"></div>

          </div>

        </div>

        <p style="font-weight: bold; font-size: 22px;">{{ allCourses.name }}</p>

        <div class="flex justify-end items-center">

          <span style="font-weight: bold;">Incluye {{ allCoursesFilter.length }}
cursos</span>

        </div>

        <p>{{ allCourses.description }}</p>

        <ul>

          <li v-for="(course) in allCoursesFilter" :key="course.id">

            {{ course.name }}

          </li>

        </ul>

        <div>

          <span style="font-size: 20px;">${{ allCourses.price }}</span>

        </div>

      </div>

    </div>

  </div>

```

```

        <div class="col-12 q-mt-md">
            <stripe-element-card
                ref="elementRef"
                :pk="key"
                @token="tokenCreated"
            />
        </div>
    </div>

    <div class="q-pa-md">
        <button class="btn-cancel" type="button" @click="modalBuyAllCourses =
false">Cancelar</button>

        <button class="btn-submit" @click="submit">Comprar</button>
    </div>

</div>
</q-card-section>
</q-card>
</q-dialog>
</q-page>
</template>

<script>
import { StripeElementCard } from '@vue-stripe/vue-stripe';
import { db, storage } from 'boot/firebase'
const ref = storage.ref()
import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'

```

```
export default {
  components: {
    StripeElementCard,

  },

  data(){
    return{
      key:
'pk_live_51KLi0qC0m96Gq1Q1NAXfrePQMbWciRwB04e8NQWWDRdiT6PTy4sFdKTyMSnkUMKsKN
uRnUT5Y9Z3YfBhzz83dFGL00myjmvqYe',

      token: null,
      courses_market: [],
      allCourses: null,
      course: null,

      modalBuy: false,
      modalBuyAllCourses: false,

      isLoading: false,
      fullPage: true,

      daily_background: "",
    }
  },

  computed: {

    ...mapState('auth', [ 'user', 'courses' ]),

    allCoursesFilter(){
```

```
        return this.courses_market.filter(doc => this.allCourses.courses.includes(doc.id))
    },
},
```

```
created() {
    this.getAllCourses()
    this.getCoursesMarket()
},
```

```
methods: {
    ...mapActions('auth', [ 'getCourses' ]),
```

```
getIt(id){
    let found = this.courses.find(doc => doc.id == id)
```

```
    if (found) {
        return true
    }
},
```

```
submit () {
    // this.makeOrder()
    this.$refs.elementRef.submit();
},
```

```
async makeOrder(){
    console.log('SE hace la orden');
    try {
        let order = {
```

```

        user: this.user.uid,
        course: this.modalBuyAllCourses ? 'Coleccion' : this.course.name,
        date: moment().format('YYYY-MM-DD'),
        status: false,
        generate_shipment: false,

    }

    let response = await db.collection('orders').add(order)
  } catch (error) {
    console.log(error);
  }
},

buyAllCourses(){
  this.modalBuyAllCourses = true
},

buyCourse(course){
  this.course = course

  this.modalBuy = true
},

async getAllCourses(){

  try {
    let response = await db.collection('all-courses')
      .doc('course')

```

```

        .get()

    if (response.exists) {
        this.allCourses = response.data()
    }
} catch (error) {
    console.log(error);
}
},

async getCoursesMarket(){
    this.courses_market = []

    try {
        let response = await db.collection('courses')
            .get()
            .then(query => {
                query.forEach(async doc => {
                    let course = doc.data()
                    course.id = doc.id
                    course.clases = []

                    let clases = await db.collection('clases')
                        .where('course', '==', doc.id)
                        .get()
                        .then(query => {
                            query.forEach(doc => {
                                let clase = doc.data()
                                clase.id = doc.id

```

```

        course.clases.push(clase)

    })

})

        this.courses_market.push(course)
    })
}

} catch (error) {
    console.log(error);
}

},

tokenCreated (token) {
    console.log(token);
    this.isLoading = true;
    this.$q.loading.show()
    try {
        let request = {
            user: this.user.uid,
            amount: this.modalBuyAllCourses ? this.allCourses.price : this.course.price,
            name: this.user.name,
            email: this.user.email,
            token_from_stripe: token.id,
            created_at: new Date()
        }
    }

```

```

    axios.post(`https://kumari-payments.herokuapp.com/charge`, request).then((response)
=> {

    if (!response.data.error) {

        request.charge = response.data.charge
        this.registerReceipt(request)
    }else{
        this.btnShow = true
        this.displayError(response.data)
    }

    }).catch(error => {
        this.btnShow = true
        console.log('Este es el error: ', error);
    });
} catch (error) {
    console.log(error);
}
},

```

```

async registerReceipt(request){
    try {
        let response = await db.collection('receipts')
            .add(request)

        if (this.modalBuyAllCourses) {
            this.registerAllCourses()
        } else {
            this.registerOneCourse()

```



```

    }
  } catch (error) {
    console.log(error);
  }
},

async registerOneCourse(){
  try {
    let date1 = moment(new Date()).unix();
    let date2 = moment(new Date()).add(6, 'months').unix();

    let data = {
      course: this.course.id,
      user: this.user.uid,
      date_active: date1,
      date_deactivate: date2
    }

    let responseCourse = await db.collection('shopping')
      .add(data)

    let firstLesson = await db.collection('classes')
      .where('course', '==', this.course.id)
      .where('order', '==', 1)
      .get()
      .then(query => {
        query.forEach(async doc => {
          let lesson = doc.data()
          lesson.id = doc.id

```

```

        let responseFinished = await db.collection('finished_courses')
            .doc(this.user.uid)
            .collection('courses')
            .doc(this.course.id)
            .add({ lessons: [lesson.id], name: lesson.name })
        })
    })

    this.sendEmail()
    this.startCourseRegister()
    this.makeOrder()

    } catch (error) {
        console.log(error);
    }
},

async startCourseRegister(){
    try {
        let dataCourse = {
            name: this.course.name,
            lessons: [],
        }

        let response = await db.collection('finished_courses')

```

```
.doc(this.user.uid)
.collection('courses')
.doc(this.course.id)
.set(dataCourse)
```

```
this.isLoading = false;
this.$q.loading.hide()
this.getCourses(this.user.uid)
alert('Gracias por su compra')
```

```
    this.$router.push('/my-account')
  } catch (error) {

  }
},
```

```
displayError(error = undefined) {
  this.$q.loading.hide()

  if (error == undefined) {
    console.log('Error');
    this.$q.notify({
      message: 'Ha ocurrido un error',
      color: 'negative'
    })
    return
  }
}
```

```
console.log(error);
```

```
if (error.error.raw.code == 'card_declined') {  
  switch (error.error.raw.decline_code) {  
    case 'insufficient_funds':  
      console.log('Fondos insuficientes');  
      this.$q.notify({  
        message: 'Fondos insuficientes',  
        color: 'negative'  
      })  
      break;  
    case 'lost_card':  
      console.log('Tarjeta extraviada');  
      this.$q.notify({  
        message: 'Tarjeta extraviada',  
        color: 'negative'  
      })  
      break;  
    case 'stolen_card':  
      console.log('Tarjeta robada');  
      this.$q.notify({  
        message: 'Tarjeta robada',  
        color: 'negative'  
      })  
      break;  
    case 'fraudulent':  
      console.log('Cargo rechazado por motivo desconocido');  
      this.$q.notify({
```

```
        message: 'Cargo rechazado por motivo desconocido',
        color: 'negative'
    })
    break;
case 'generic_decline':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'Cargo rechazado por motivo desconocido, contacta tu banco emisor',
        color: 'negative'
    })
    break;
case 'authentication_required':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'Cargo rechazado por motivo desconocido, contacta tu banco emisor',
        color: 'negative'
    })
    break;
case 'approve_with_id':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'Cargo rechazado por motivo desconocido, contacta tu banco emisor',
        color: 'negative'
    })
    break;
case 'call_issuer':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'Cargo rechazado por motivo desconocido, contacta tu banco emisor',
```

```

        color: 'negative'
    })
    break;
case 'card_not_supported':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'La tarjeta no admite este tipo de compras, contacta tu banco emisor',
        color: 'negative'
    })
    break;
case 'card_velocity_exceeded':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'Excedido el límite del saldo o del crédito disponible, contacta tu banco
emisor',
        color: 'negative'
    })
    break;
case 'currency_not_supported':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'La tarjeta no acepta la moneda especificada.',
        color: 'negative'
    })
    break;
case 'duplicate_transaction':
    console.log('Cargo rechazado por motivo desconocido');
    this.$q.notify({
        message: 'Hace muy poco se realizó otra transacción por el mismo importe con los
mismos datos de tarjeta de crédito.',

```

```
        color: 'negative'
    })
    break;

    default:
        console.log('Ocurrio un error');
        break;
    }
} else if(error.error.raw.code == 'expired_card'){
    console.log('Tarjeta expirada');
    this.$q.notify({
        message: 'Tarjeta expirada',
        color: 'negative'
    })
    return
} else if(error.error.raw.code == 'incorrect_cvc'){
    console.log('CVC incorrecto');
    this.$q.notify({
        message: 'CVC incorrecto',
        color: 'negative'
    })
    return
} else{
    console.log('Ocurrio un error');
    this.$q.notify({
        message: 'Ocurrio un error',
        color: 'negative'
    })
    return
}
```

```
}
```

```
this.$q.notify({  
  message: 'Ocurrio un error',  
  color: 'negative'  
})  
console.log('Ocurrio un error');  
return  
},  
  
async registerAllCourses(){  
  try {  
    let bar = new Promise((resolve, reject) => {  
      this.allCoursesFilter.forEach(async (course, index, array) => {  
  
        let found = this.courses.find(doc => doc.id == course.id)  
  
        if (!found) {  
  
          let date1 = moment(new Date()).unix();  
          let date2 = moment(new Date()).add(6, 'months').unix();  
  
          let data = {  
            course: course.id,  
            user: this.user.uid,  
            date_active: date1,  
            date_deactivate: date2  
          }  
        }  
      })  
    })  
  }  
}
```



```

let responseCourse = await db.collection('shopping')
    .add(data)

let firstLesson = await db.collection('classes')
    .where('course', '==', course.id)
    .where('order', '==', 1)
    .get()
    .then(query => {
        query.forEach(async doc => {
            let lesson = doc.data()
            lesson.id = doc.id

            let responseFinished = await db.collection('finished_courses')
                .doc(this.user.uid)
                .collection('courses')
                .doc(course.id)
                .add({ lessons: [lesson.id], name:
lesson.name })

            })
        })

let dataCourse = {
    name: course.name,
    lessons: [],
}

await db.collection('finished_courses')
    .doc(this.user.uid)

```

```

        .collection('courses')
        .doc(course.id)
        .set(dataCourse)
    }

    if (index === array.length -1) resolve();
  })
});

bar.then(() => {
  this.sendEmail()
  this.makeOrder()
  this.getCourses(this.user.uid)
  this.$q.loading.hide()

  alert('Gracias por su compra')
  // this.$router.push('/my-account')
});

} catch (error) {
  console.log(error);
}
},

async sendEmail(){
  try {
    let mail = {
      to: this.user.email,

```

```

subject: 'Compra KUMARI',
message: {
  text: '¡Gracias por tu compra!',
  html: `
    <p>¡Gracias por tu compra!</p>
    <br />
    <p>¿Estas list@ para vivir la experiencia Kumari?</p>
    <p>Descarga la aplicación oficial en tu dispositivo o ingresa en el siguiente link de
    acceso en tu computadora: <a href="https://app.kumari.mx">KUMARI</a></p>
    <p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>
  `
}

await db.collection('mails').add(mail)
} catch (error) {
  console.log(error);
}
},

// async registerAllCourses(){
//   try {
//     this.courses_market.forEach(async course => {
//       let date1 = moment(new Date()).unix();
//       let date2 = moment(new Date()).add(6, 'months').unix();

//       let data = {
//         course: course.id,

```

```

//      user: this.user.uid,
//      date_active: date1,
//      date_deactivate: date2
//    }

//    let responseCourse = await db.collection('shopping')
//      .add(data)

//    let firstLesson = await db.collection('clases')
//      .where('course', '==', course.id)
//      .where('order', '==', 1)
//      .get()
//      .then(query => {
//        query.forEach(async doc => {
//          let lesson = doc.data()
//          lesson.id = doc.id

//          let responseFinished = await db.collection('finished_courses')
//            .doc(this.user.uid)
//            .collection('courses')
//            .doc(course.id)
//            .add({ lessons: [lesson.id], name: lesson.name
//        })
//      })
//    })

//    let dataCourse = {
//      name: course.name,

```

```

//      lessons: [],
//      }

//      await db.collection('finished_courses')
//          .doc(this.user.uid)
//          .collection('courses')
//          .doc(course.id)
//          .set(dataCourse)

//      console.log('Gracias por su compra');

//      this.getCourses(this.user.uid)
//      })
//  } catch (error) {
//      console.log(error);
//  }
//}

},
}
</script>

```

## Users

### Account.txt

```
<style scoped lang="scss">
```

```

.main-container{

    background-color: white;

    border-radius: 10px;

    padding: 20px;

}

```

```
.header{
  width: 100%;
  display: flex;
  justify-content: space-around;
  align-items: center;

  .profile-picture-container{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;

    .profile-picture{
      width: 100px;
      height: 100px;

      img{
        width: 100px;
        height: 100px;
        object-fit: cover;
        object-position: center;
        border-radius: 15px;
      }
    }
  }

  .btn-change-photo{
    background-color: #24B8CB;
    padding: 5px 10px;
```

```
border-radius: 5px;  
cursor: pointer;  
color: white;  
}
```

```
}
```

```
.profile-info{  
margin-left: 50px;  
line-height: 15px;  
p{  
font-size: 25px;  
font-weight: bold;  
text-transform: capitalize;  
}
```

```
span{  
font-size: 23px;  
}  
}  
}
```

```
.main-course-container{  
width: 100%;  
overflow-x: hidden;
```

```
.course-container-body{  
border-radius: 10px;
```

```
aspect-ratio: 16/9;
display: flex;
justify-content: center;
align-items: center;
position: relative;
```

```
img{
  width: 80%;
}
```

```
.layer-course{
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
}
}
```

```
.course-container{

  padding: 10px;
  cursor: pointer;

.course-body{
  background-color: white;
```



```
border-radius: 10px;  
position: relative;  
aspect-ratio: 16/9;  
object-fit: cover;  
object-position: center;
```

```
img{  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
  object-position: center;  
  border-radius: 10px;  
}
```

```
.layer{  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 97%;  
  border-radius: 10px;  
  background-color: rgba(0, 0, 0, 0.1);  
  display: flex;  
  justify-content: center;  
  align-items: center;
```

```
i{  
  color: white;  
  font-size: 40px;
```

```
    }

    &.active{
        background-color: rgba(0, 0, 0, 0.4);
    }
}
}
```

```
.name-title{
    text-align: center;
    font-size: 22px;
    font-weight: bold;
}
}
}
```

```
.courses-web{
    display: block;
}
```

```
.courses-movil{
    display: none;
}
```

```
@media (max-width: 1006px) {
    .profile-info{
        margin-top: 20px;
        p{
            font-size: 16px !important;
        }
    }
}
```

```
}
```

```
span{
```

```
    font-size: 15px !important;
```

```
}
```

```
}
```

```
.courses-web{
```

```
    display: none;
```

```
}
```

```
.courses-movil{
```

```
    display: block;
```

```
}
```

```
.name-title{
```

```
    text-align: center;
```

```
    font-size: 15px;
```

```
    font-weight: bold;
```

```
}
```

```
}
```

```
.btn-modal{
```

```
border: none;
```

```
background-color: #24B8CB;
```

```
color: white;
```

```
border-radius: 15px;
```

```
padding: 10px 15px;  
font-weight: bold;  
font-size: 17px;  
}
```

```
.form-group{  
  margin-top: 20px;  
  
  label{  
    font-weight: bold;  
    font-size: 18px;  
  }  
}
```

```
.input-text{  
  width: 100%;  
  border: 2px solid #666666;  
  border-radius: 10px;  
  padding: 5px;  
}
```

```
.input-text:focus{  
  outline: none;  
}
```

```
.btn-cancel{  
  color: white;  
  padding: 5px 20px;  
  border: none;
```

```
background-color: #E72B81;  
border-radius: 10px;  
cursor: pointer;  
}
```

```
.btn-submit{  
color: white;  
padding: 5px 20px;  
border: none;  
background-color: #24B8CB;  
border-radius: 10px;  
margin-left: 10px;  
cursor: pointer;  
}
```

```
.badge-color{  
width: 20px;  
height: 20px;  
border-radius: 50%;  
margin-right: 10px;  
cursor: pointer;  
  
}
```

```
.active{  
box-shadow: 1px 1px 10px rgba(0, 0, 0, 0.8);  
}
```

```
.form-group{
```

```
padding: 10px 5px;  
}
```

```
label{  
    font-size: 18px;
```

```
    input{  
        margin-right: 10px;  
    }  
}
```

```
input[type=text]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
input[type=email]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
input[type=date]{  
    width: 100%;
```

```
border-radius: 10px;  
border: none;  
border: 1px solid black;  
padding: 5px;  
}
```

```
input[type=password]{  
width: 100%;  
border-radius: 10px;  
border: none;  
border: 1px solid black;  
padding: 5px;  
}
```

```
select{  
width: 100%;  
border-radius: 10px;  
border: none;  
border: 1px solid black;  
padding: 5px;  
}
```

```
.btn-edit{  
background-color: transparent;  
border: none;  
cursor: pointer;  
width: 25px;  
height: 25px;  
display: flex;
```

```
justify-content: center;
align-items: center;
margin-top: -15px;
margin-left: 10px;
border-radius: 10px;
background-color: #24B8CB;
color: white;
font-size: 15px;
```

```
&:hover{
    background-color: #E72B81;
    color: white;
}
}
```

</style>

<template>

<q-page>

<div class="main-container">

<div class="row">

<div class="col-12 header">

<div class="flex justify-center items-center">

<div class="profile-picture-container">

<div class="profile-picture">



</div>

<div>

<input @change="getUserImage" style="display: none;" id="user\_image" type="file">



```

    </div>

    <label class="btn-change-photo" for="user_image">Cambiar</label>
  </div>

  <div class="profile-info">
    <div class="flex">
      <p class="">{{ user.name }}</p>

      <button class="btn-edit" @click="modalEditUser = true, editUser = user">
        <q-icon name="edit" />
      </button>
    </div>

    <span>No. Cuenta: {{ user.partner_number }}</span>
  </div>
</div>
</div>
</div>
</div>

<div class="row">
  <div class="col-12">
    <h5>Licencia</h5>
  </div>

  <div class="col-12 flex justify-around items-center">
    <div>
      <span style="font-size: 22px; color: green;">Activa</span>

      <span style="font-size: 22px;"> Renovación <span v-if="user.date_deactivate">"{{
user.date_deactivate | formatDate }}"</span></span>
    </div>
  </div>
</div>
</div>

```

```

<div class="row">
  <div class="col-12">
    <h5>Mis cursos activos</h5>
  </div>
  <div class="main-course-container courses-web" v-if="courses.length != 0">
    <swiper ref="mySwiper" :options="swiperOptions">
      <swiper-slide v-for="(item, index) in courses" :key="item.id">
        <div class="course-container">
          <div class="course-container-body" :style="{backgroundColor: item.color}">

            

            <div class="layer-course"></div>

          </div>
          <p class="name-title">{{ item.name }}</p>
        </div>
      </swiper-slide>
    </swiper>
  </div>

```

```

<div class="main-course-container courses-movil" v-if="courses.length != 0">
  <div v-for="(item, index) in courses" :key="item.id">
    <div class="course-container">
      <div class="course-container-body" :style="{backgroundColor: item.color}">

        

        <div class="layer-course"></div>

      </div>
      <p class="name-title">{{ item.name }}</p>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-12">
        <h5>Fondos de pantalla</h5>
    </div>
    <div class="main-course-container courses-web">
        <swiper ref="mySwiper2" :options="swiperOptions">
            <swiper-slide v-for="(item, index) in list_backgrounds" :key="index" class="course-
container">
                <div @click="selectPhoto(item, index)">
                    <div class="course-body">
                        
                        <div class="layer" :class="[ user.background_image == item.photo ? 'active' : "
]">
                            <q-icon v-if="user.background_image == item.photo" name="check" />
                        </div>
                    </div>
                    <p class="name-title">{{ item.name }}</p>
                </div>
            </swiper-slide>
        </swiper>
    </div>

<div class="main-course-container courses-movil">
    <div v-for="(item, index) in list_backgrounds" :key="index" class="course-container">
        <div @click="selectPhoto(item, index)">
            <div class="course-body">

```

```

        

        <div class="layer" :class="[ user.background_image == item.photo ? 'active' : "
]">

            <q-icon v-if="user.background_image == item.photo" name="check" />

        </div>

    </div>

    <p class="name-title">{{ item.name }}</p>

</div>

</div>

</div>

</div>

```

```

<q-dialog v-model="modalEditUser" persistent v-if="editUser != null">

    <q-card style="min-width: 800px;">

        <q-card-section class="row">

            <div class="col-12">

                <q-form

                    @submit="addUserEdit"

                    class="q-gutter-md">

                        <div class="row">

                            <div class="col-6" style="padding: 20px;">

                                <div class="form-group">

                                    <label for="">Nombre del niño</label>

                                    <br>

                                    <input v-model="editUser.children" type="text">

                                    <!-- <div class="error" v-if="!$v.editUser.children.required">Field is
required</div -->

                                </div>

                                <div class="form-group">

                                    <label for="">Fecha de nacimiento</label>

```

```

        <br>

        <input v-model="editUser.birthdate" type="date">
    </div>
    <div class="form-group">
        <label for="">Intereses</label>

        <br>

        <div>
            <div v-for="(interest, index) in interests" :key="index">

                <label>

                    <input v-model="editUser.interest" :value="interest"
type="checkbox" name="interes">{{ interest }}

                </label>

                <br>

            </div>

        </div>
    </div>
    <div class="form-group">
        <label for="">Grado Academico</label>

        <br>

        <select v-model="editUser.academy" name="grado" id="">

            <option value="1">1ro de Primaria</option>
            <option value="2">2o de Primaria</option>
            <option value="3">3ro de Primaria</option>
            <option value="4">4ro de Primaria</option>
            <option value="5">5ro de Primaria</option>
            <option value="6">6ro de Primaria</option>

        </select>

    </div>
    <div class="form-group">

```

```
<label for="">Escuela</label>

<br>

<input v-model="editUser.school" type="text">

</div>

</div>

<div class="col-6" style="padding: 20px;">

  <div class="form-group">

    <label for="">Nombre del Padre o Tutor</label>

    <br>

    <input v-model="editUser.name" type="text">

  </div>

  <div class="row">

    <div class="col-6">

      <div class="form-group">

        <label for="">Direccion</label>

        <br>

        <input v-model="editUser.direction" type="text">

      </div>

    </div>

    <div class="col-6">

      <div class="form-group">

        <label for="">Codigo Postal</label>

        <br>

        <input v-model="editUser.cp" type="text">

      </div>

    </div>

  </div>

</div>
```

```
<div class="row">

  <div class="col-6">

    <div class="form-group">

      <label for="">Ciudad</label>

      <br>

      <input v-model="editUser.city" type="text">

    </div>

  </div>

  <div class="col-6">

    <div class="form-group">

      <label for="">Estado</label>

      <br>

      <input v-model="editUser.state" type="text">

    </div>

  </div>

  <div class="form-group">

    <label for="">Telefono</label>

    <br>

    <input v-model="editUser.telephone" type="text">

  </div>

</div>

<div class="q-pa-md flex justify-end">

  <button class="btn-cancel" type="button" @click="modalEditUser =
false">Cerrar</button>

  <button class="btn-submit" type="submit">Guardar</button>

</div>
```

```
        </div>
      </q-form>
    </div>
  </q-card-section>
</q-card>
</q-dialog>
</div>
</q-page>
</template>
```

```
<script>
import { StripeElementCard } from '@vue-stripe/vue-stripe';
import { db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions, mapMutations } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'

import VueSlickCarousel from 'vue-slick-carousel'
import 'vue-slick-carousel/dist/vue-slick-carousel.css'
// optional style for arrows & dots
import 'vue-slick-carousel/dist/vue-slick-carousel-theme.css'

import { Swiper, SwiperSlide, directive } from 'vue-awesome-swiper'
// import style (>= Swiper 6.x)
import 'swiper/swiper-bundle.css'

export default {
  components: {
```



```
StripeElementCard,  
VueSlickCarousel,  
Swiper,  
SwiperSlide  
},
```

```
directives: {  
  swiper: directive  
},
```

```
data(){  
  return{  
    key: 'pk_test_RhrZoLpQYWs3NO9Ah4J5Fnbh00k04KEtnp',  
    token: null,  
  
    modalEditUser: false,  
    editUser: null,  
  
    swiperOptions: {  
      slidesPerView: 4,  
      spaceBetween: 10  
    },  
  
    list_courses: [],  
    list_backgrounds: [  
      {  
        photo: 'Verde.jpg',  
        name: 'Verde',  
      },  
    ],
```

```
{  
  photo: 'Azul.jpg',  
  name: 'Azul',  
},  
{  
  photo: 'Naranja.jpg',  
  name: 'Naranja',  
}  
],
```

```
settings2: {  
  "arrows": false,  
  "dots": false,  
  "focusOnSelect": false,  
  "infinite": false,  
  "speed": 500,  
  "slidesToShow": 3,  
  "touchThreshold": 5,  
  "centerMode": false,  
  "centerPadding": "20px",  
},
```

```
settings: {  
  "arrows": false,  
  "dots": false,  
  "focusOnSelect": false,  
  "infinite": true,  
  "speed": 500,  
  "slidesToShow": 3,
```

```
"touchThreshold": 5,  
"centerMode": true,  
"centerPadding": "20px",  
},
```

```
generalIndex: this.generateUUID(),
```

```
handClick: true,  
bgImage: "",
```

```
interests: [  
  'Ciencia',  
  'Matemáticas',  
  'Robótica',  
  'Lectura',  
  'Escritura',  
  'Fotografía',  
  'Artes',  
  'Baile',  
  'Geografía',  
  'Programación',  
  'Naturaleza',  
  'Animales',  
  'Animación',  
  'Videojuegos'  
]  
}  
},
```

```

filters: {
  formatDate(date){
    // multiplied by 1000 so that the argument is in milliseconds, not seconds.
    var new_date = new Date(date * 1000);

    return moment(new_date).format('L')
  }
},

computed: {
  ...mapState('auth', [ 'user', 'courses']),
},

created() {
  this.getCourses(this.user.uid)
  // this.getUsers()
},

methods: {
  ...mapMutations('auth', [ 'updateBgUser', 'updatePhotoUser' ]),

  ...mapActions('auth', [ 'getCourses' ]),

  // async getUsers(){

  //   try {
  //     let response = await db.collection('users')
  //       .doc(this.user.uid)
  //       .get(query => {

```

```

//          this.users = []

//          query.forEach(async doc => {
//              let user = doc.data()
//              user.id = doc.id

//              this.users.push(user)
//          })
//      })
//  } catch (error) {
//      console.log(error);
//  }
// },

```

```

handleUp(){
    this.handleClick = false
    console.log(this.handleClick);
},

```

```

handleDown(){
    this.handleClick = true
    console.log(this.handleClick);
},

```

```

async selectPhoto(item){
    setTimeout(async () => {
        try {
            if (this.handleClick) {

```

```
        await db.collection('users').doc(this.user.uid).update({ background_image:
item.photo })
```

```
        this.updateBgUser(item.photo)
```

```
        alert('Foto actualizada')
```

```
    }
```

```
    } catch (error) {
```

```
        console.log(error);
```

```
    }
```

```
    }, 100)
```

```
  },
```

```
  async getUserImage(e){
```

```
    let user_image = e.target.files[0]
```

```
    try {
```

```
      let metadata = {
```

```
        contentType: "image/jpeg",
```

```
        contentType: "image/jpeg",
```

```
      }
```

```
      let fotold = this.generateUUID()
```

```
      const reflmg = ref.child('user_image/' + fotold + '.jpg')
```

```
      return reflmg.put(user_image, metadata)
```

```
    }.then(e => {
```

```
      console.log(e)
```

```
      return ref.child('user_image/' + fotold + '.jpg').getDownloadURL()
```

```
    }).then(async (url) => {
```

```

        await db.collection('users').doc(this.user.uid).update({ photo: url })

        this.updatePhotoUser(url)

    })

    })

    .catch(error => console.log(error))
  } catch (error) {
    console.log(error);
  }
},

getImage(e){
  this.bgImage = e.target.files[0]
  console.log(this.bgImage);
},

async uploadImage(){
  try {
    let metadata = {
      contentType: "image/jpeg",
      contentLanguage: "es",
    }

    let fotold = this.generateUUID()
    const refImg = ref.child('bg/' + fotold + '.jpg')
    return refImg.put(this.bgImage, metadata)
    .then(e => {
      console.log(e)
    })
  }
}

```

```

return ref.child('bg/' + fotold + '.jpg').getDownloadURL()

.then(async (url) => {
    await db.collection('users').doc(this.user.uid).update({ background_image: url })

    this.updateBgUser(url)

})

})

.catch(error => console.log(error))
} catch (error) {
    console.log(error);
}
},

```

```

async getCourses(){
    try {
        let response = await db.collection('shopping')
            .where('user', '==', this.user.uid)
            .get()
            .then(query => {
                query.forEach(async course => {
                    let co = await db.collection('courses')
                        .doc(course.data().course)
                        .get()

                    let cour = co.data()

                    cour.id = course.data().course

                    this.list_courses.push(cour)
                })
            })
    }
}

```



```

        })
    } catch (error) {
        console.log(error);
    }
},

submit () {
    // You will be redirected to Stripe's secure checkout page
    this.$refs.elementRef.submit();
},

tokenCreated (token) {
    console.log(token);

    let request = {
        user: this.user.uid,
        amount: 100,
        name: this.user.name,
        email: this.user.email,
        token_from_stripe: token.id,
        created_at: new Date()
    }

    axios.post(`https://kumari-payments.herokuapp.com/charge`, request).then((response) =>
    {
        if (!response.data.error) {

            request.charge = response.data.charge
            this.submitPayment(request)

```

```

    }else{

        this.btnShow = true

        this.displayError(response.data)

    }

}).catch(error => {

    this.btnShow = true

    loader.hide()

    console.log('Este es el error: ', error);

});

},

async submitPayment(request){

    try {

        console.log(request);

    } catch (error) {

        loader.hide()

        console.log(error);

    }

},

displayError(error = undefined) {

    if (error == undefined) {

        console.log('Error');

        return

    }

    if (error.error.raw.code == 'card_declined') {

        switch (error.error.raw.decline_code) {

```

```
    case 'insufficient_funds':  
        console.log('Fondos insuficientes');  
        break;  
    case 'lost_card':  
        console.log('Tarjeta extraviada');  
        break;  
    case 'stolen_card':  
        console.log('Tarjeta robada');  
        break;  
  
    default:  
        console.log('Ocurrio un error');  
        break;  
    }  
} else if(error.error.raw.code == 'expired_card'){  
    console.log('tarjeta expirada');  
    return  
} else if(error.error.raw.code == 'incorrect_cvc'){  
    console.log('CVC incorrecto');  
    return  
} else if(error.error.raw.code == 'incorrect_cvc'){  
    console.log('CVC incorrecto');  
    return  
} else{  
    console.log('Ocurrio un error');  
    return  
}  
}
```

```
console.log('Ocurrio un error');
```



```
        console.log(error);
    }
},

},
}
</script>
```

### Activity.txt

```
<style scoped lang="scss">

    .main-container{

        background-color: white;

        border-radius: 10px;

        padding: 20px;

    }


    .main-title{

        font-size: 22px;

        font-weight: bold;

    }


    .btn-options{

        border: none;

        margin: 5px;

        font-size: 18px;

        color: white;

        border-radius: 5px;

        cursor: pointer;

    }
```

```
.btn-modal{  
    border: none;  
    background-color: #24B8CB;  
    color: white;  
    border-radius: 15px;  
    padding: 10px 15px;  
    font-weight: bold;  
    font-size: 17px;  
}
```

```
.form-group{  
    margin-top: 20px;  
  
    label{  
        font-weight: bold;  
        font-size: 18px;  
    }  
}
```

```
.input-text{  
    width: 100%;  
    border: 2px solid #666666;  
    border-radius: 10px;  
    padding: 5px;  
}
```

```
.input-text:focus{  
    outline: none;  
}
```

```
.btn-cancel{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #E72B81;  
    border-radius: 10px;  
    cursor: pointer;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;  
}
```

```
.input-date{  
    border: 1px solid gray;  
    padding: 2px;  
    border-radius: 5px;  
}
```

```
.btn-date-search{  
    margin-left: 10px;  
}
```

</style>

<template>

<q-page>

<div class="main-container">

<q-table

title="Mis Cursos"

:data="homeworks"

:columns="columns"

:filter="filter"

row-key="name"

flat

>

<template v-slot:top>

<div class="flex justify-between items-center" style="width: 100%;">

<span style="font-size: 25px; font-weight: bold;">Mi actividad</span>

<div class="flex justify-between items-center">

<input type="date" v-model="date1" class="input-date">

<p style="margin: 10px;">a</p>

<input type="date" v-model="date2" class="input-date">

<button @click="searchHomework" class="btn-date-search">

<q-icon name="search" />

</button>

</div>

<q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"  
color="primary" v-model="filter">

<template v-slot:append>



```
        <q-icon name="search" />
      </template>
    </q-input>
  </div>
```

```
</template>
```

```
<template v-slot:body-cell-course="props">
```

```
  <q-td :props="props">
    {{ props.row.course_data.name }}
  </q-td>
```

```
</template>
```

```
<template v-slot:body-cell-lesson="props">
```

```
  <q-td :props="props">
    {{ props.row.lesson_data.name }}
  </q-td>
```

```
</template>
```

```
<template v-slot:body-cell-date="props">
```

```
  <q-td :props="props">
    {{ props.row.date | formatDate }}
  </q-td>
```

```
</template>
```

```
<template v-slot:body-cell-evidence="props">
```

```
  <q-td :props="props">
    <a :href="props.row.homework" target="_blank" download>Ver</a>
  </q-td>
```

```
</template>
```

```
<template v-slot:body-cell-state1="props">
```

```
  <q-td :props="props">
    
```

```

        </q-td>
    </template>
    <template v-slot:body-cell-state2="props">
        <q-td :props="props">
            
        </q-td>
    </template>
</q-table>
</div>
</q-page>
</template>

```

```

<script>
import { db, storage } from 'boot/firebase'
const ref = storage.ref()
import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'
import { required, minLength, email } from 'vuelidate/lib/validators'

```

```

export default {
  data(){
    return{
      date1: "",
      date2: "",

      homeworks: [],
      filter: "",
      columns: [

```

```

    {
      name: 'course',
      required: true,
      label: 'Curso',
      align: 'left',
      field: row => row.created_at,
      format: val => `${val}`,
      sortable: true
    },
    { name: 'lesson', align: 'left', label: 'Clase', field: 'lesson', sortable: true },
    { name: 'date', align: 'left', label: 'fecha', field: 'date', sortable: true },
    { name: 'evidence', label: 'Evidencia', field: 'evidence' },
    { name: 'state1', label: 'Estado Inicial', field: 'state1' },
    { name: 'state2', label: 'Estado Final', field: 'state2' }

  ],
}
},

```

```

validations: {
  date1: {
    required
  },
  date2: {
    required
  },
},

```

```

computed: {

```

```
...mapState('auth', [ 'user' ])  
},
```

```
filters: {  
  formatDate(args){  
    let date = moment(args.toDate()).format('LL');  
    return date  
  }  
},
```

```
created() {  
  this.getHomeworks()  
},
```

```
methods: {  
  returnImageStatus(status){  
    switch (status) {  
      case 0:  
        return '/Enojado.png'  
        break;  
      case 1:  
        return '/Triste.png'  
        break;  
      case 2:  
        return '/Cansado.png'  
        break;  
      case 3:  
        return '/Feliz.png'  
        break;  
    }  
  }  
}
```

```

    case 4:
        return '/Emocionado.png'
        break;

    default:
        return '/Feliz.png'
        break;
    }
},

async searchHomework(){
    try {

        if (!this.$v.date0.required || !this.$v.date2.required) {
            this.$q.notify({
                message: 'Ingresa una fecha correcta',
                color: 'negative'
            })

            return
        }

        let date1 = moment(this.date1).unix()
        let date2 = moment(this.date2).unix()

        let response = await db.collection('homework')
            .where('user', '==', this.user.uid)
            .where('dateUnix', '>=', date1)
            .where('dateUnix', '<=', date2)
    }
}

```

```

        .get()
        .then(query => {
            this.homeworks = []

            query.forEach(async doc => {
                let homework = doc.data()
                homework.id = doc.id

                let course = await db.collection('courses').doc(homework.course).get()
                homework.course_data = course.data()

                let lesson = await db.collection('clases').doc(homework.lesson).get()
                homework.lesson_data = lesson.data()

                this.homeworks.push(homework)
            })
        })
    } catch (error) {
        console.log(error);
    }
},

```

```

async getHomeworks(){
    try {
        let response = await db.collection('homework')
            .where('user', '==', this.user.uid)
            .get()
            .then(query => {
                query.forEach(async doc => {

```

```
let homework = doc.data()
```

```
homework.id = doc.id
```

```
let course = await db.collection('courses').doc(homework.course).get()
```

```
homework.course_data = course.data()
```

```
let lesson = await db.collection('clases').doc(homework.lesson).get()
```

```
homework.lesson_data = lesson.data()
```

```
this.homeworks.push(homework)
```

```
}}
```

```
}}
```

```
} catch (error) {
```

```
  console.log(error);
```

```
}
```

```
},
```

```
showReceipt(receipt){
```

```
  return
```

```
}
```

```
},
```

```
}
```

```
</script>
```

Questions.txt

```
<style scoped lang="scss">
```

```
.main-container{
```

```
  background-color: white;
```

```
  border-radius: 10px;
```

```
padding: 20px;  
}
```

```
.main-title{  
  font-size: 22px;  
  font-weight: bold;  
}
```

```
.form-question{  
  border-radius: 10px;  
}
```

```
.question{  
  border-radius: 10px;  
}
```

```
.question-header{  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
.question-header p:nth-child(1){  
  font-size: 18px;  
  font-weight: bold;  
}
```

```
.formulario{
```



```

button{
    color: white;
    border: none;
    background-color: #E72B81;
    border-radius: 10px;
    cursor: pointer;
}
}

```

```

</style>

```

```

<template>

```

```

<q-page>

```

```

<div class="main-container">

```

```

<p class="main-title">Preguntas Frecuentes</p>

```

```

<div class="row">

```

```

<div class="col-8 q-pl-md q-pr-md">

```

```

<div class="row">

```

```

<div class="col-12 q-pa-md question q-mt-sm" v-for="(question, index) in questions"
:key="index">

```

```

<div class="question-header">

```

```

<p>{{ question.question }}</p>

```

```

</div>

```

```

<p>{{ question.answer }}</p>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```
    </q-page>
  </template>
```

```
<script>
//Firebase
import { firebase, db, firestore, storage } from 'boot/firebase'
const ref = storage.ref()
```

```
import { mapState, mapActions } from 'vuex'
var moment = require('moment');
```

```
export default {
  name: 'questions',
```

```
  data(){
    return{
      questions: [],
      questions: [],

      newQuestion: {
        question: "",
        answer: "",
      },

      modalSpace: false,
      modalEditQuestion: false,
      editQuestion: null,
    }
  },
```

```
created() {  
  this.getQuestions()  
},
```

```
filters: {  
  formatDay(time){  
    let date = time.toDate()  
  
    moment.locale('es');  
    return moment(date).format('LLL');  
  }  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ])  
},
```

```
methods: {  
  async addQuestion(){  
    try {  
      try {  
        this.$q.dialog({  
          title: 'Agregar nueva pregunta',  
          message: '¿Quieres registrar esta pregunta?',  
          cancel: true,  
          persistent: true  
        }).onOk(async () => {  
          let reponse = await db.collection('questions').add(this.newQuestion)
```

```
this.newQuestion.question = "
```

```
this.newQuestion.answer = "
```

```
this.$q.notify({  
  message: 'Pregunta registrada',  
  color: 'positive'  
})
```

```
}}
```

```
} catch (error) {  
  console.log(error);  
}
```

```
} catch (error) {  
  console.log(error);  
}
```

```
},
```

```
async getQuestions(){  
  try {  
    let response = await db.collection('questions')
```

```
      .onSnapshot(query => {
```

```
        this.questions = []
```

```
        let question
```

```
        query.forEach(doc => {
```

```
          question = doc.data()
```

```
        Object.defineProperty(question, 'id', {
```

```
          enumerable: true,
```

```

        configurable: true,
        writable: true,
        value: doc.id
    });

    this.questions.push(question)
    })
    }
} catch (error) {
    console.log(error);
}
},

```

```

async addQuestionEdit(){
    try {
        try {
            let response = await db.collection('questions')
                .doc(this.editQuestion.id)
                .update(this.editQuestion)

            this.$q.notify({
                message: 'Pregunta editada',
                color: 'positive'
            })
        } catch (error) {
            console.log(error);
        }

    } catch (error) {

```

```

        console.log(error);
    }
},

async deleteQuestion(question){
    try {
        this.$q.dialog({
            title: 'Eliminar pregunta',
            message: '¿Estas seguro de eliminar esta pregunta?',
            cancel: true,
            persistent: true
        }).onOk(async () => {
            let response = await db.collection('questions').doc(question.id).delete()
            this.editQuestion = null
            this.$q.notify({
                message: 'Pregunta eliminado',
                color: 'positive'
            })
        })

    } catch (error) {
        console.log(error);
    }
},

},

}
</script>

```

Receipts.txt

```
<style scoped lang="scss">

  .main-container{

    background-color: white;

    border-radius: 10px;

    padding: 20px;

  }


  .btn-options{

    color: black;

  }


  // .section-receipt{
  //   background-image: url('/full.png');
  //   background-position: top;
  //   background-size: cover;
  //   height: 100vh;
  // }

</style>
```

```
<template>

  <q-page>

    <div class="main-container">

      <div class="row">

        <div class="col-12 flex justify-center">

          <q-table

            title="Mis Cursos"

            :data="receipts"

            :columns="columns"

            :filter="filter"

          >
```

```

        row-key="date"

        flat
    >
    <template v-slot:top>
        <div class="flex justify-between items-center" style="width: 100%;">
            <span style="font-size: 25px; font-weight: bold;">Recibos</span>
        </div>

    </template>

    <template v-slot:body-cell-created_at="props">
        <q-td :props="props">
            {{ props.row.created_at | formatDate }}
        </q-td>
    </template>

    <template v-slot:body-cell-course="props">
        <q-td :props="props">
            {{ props.row.id }}
        </q-td>
    </template>

    <template v-slot:body-cell-options="props">
        <q-td :props="props">
            <span class="btn-options cursor-pointer" @click="showReceipt(props.row)">
                <i class="fas fa-download"></i>
            </span>
        </q-td>
    </template>
</q-table>
</div>
</div>

```



```
<!-- <div class="row">

  <div class="col-12">

    <div class="row">

      <div class="col-12">

        

      </div>

    </div>

    <div class="row">

      <div class="col-12">

        <p style="color: #E72B81; font-weight: bold;">APRENDE CON KUMARI S.A. de
C.V.</p>

        <p style="color: #E72B81; font-weight: bold; font-size: 35px; line-height:
20px;">Recibo por servicios</p>

        <div style="margin-top: 50px;">

          <p style="color: #293270; font-weight: bold; letter-spacing: 3px;">FACTURA N.° #
| 31 DE JULIO DE 2022</p>

        </div>

      </div>

    </div>

  </div>

  <hr style="border: 1px solid yellow;">

  <div class="row" style="margin-top: 20px;">

    <div class="col-12">

      <p style="color: #293270; font-weight: bold;">Factura a nombre de:</p>
```

14710

Gerardo Lucero

Juan Leautaud

Descripcion

Suscripcion 6 meses de todos los cursos de Kumari + K-Box

Precio unitario

1 x \$3000.00

Total

Solo \$3000.00

</div>

<hr style="border: 1px solid yellow;">

<div class="row" style="margin-top: 20px;">

<div class="col-6">

</div>

<div class="col-3">

<p style="color: #E72B81; font-weight: bold;">Subtotal</p>

<p style="color: #E72B81; font-weight: bold;">IVA</p>

<p style="color: #E72B81; font-weight: bold;">Total</p>

</div>

<div class="col-3">

<p style="color: #293270">\$3000.00</p>

<p style="color: #293270">\$480.00</p>

<p style="color: #293270">\$3480.00</p>

</div>

</div>

</div>

</div>

<div class="row" style="margin-top: 20px;">

<div class="col-12">

<p style="color: #E72B81; font-weight: bold;">Informacion del banco</p>

```
        <p style="color: #293270">Bancamex</p>
    </div>
</div>
<div class="row">
    <div class="col-12">
        <p style="color: #E72B81; font-weight: bold;">N.º de cuenta</p>

        <p style="color: #293270">ABC987654321</p>
    </div>
</div>
</div>
</div> -->
```

```
<div class="row">
    <div class="col-12">
        <VueHtml2pdf
            :show-layout="false"
            :float-layout="true"
            :enable-download="false"
            :preview-modal="true"
            :paginate-elements-by-height="1400"
            filename="myPDF"
            :pdf-quality="2"
            :image="{ type: 'jpg', quality: 1 }"
            :manual-pagination="false"
            pdf-format="a4"
            pdf-orientation="portrait"
            pdf-content-width="800px"
```

```

ref="html2Pdf"
>
<section slot="pdf-content">
  
  <div class="row section-receipt" style="padding: 78px;" v-if="receipt != null">
    <div class="col-12">

      <div class="row">
        <div class="col-12">
          <p style="color: #E72B81; font-weight: bold;">APRENDE CON KUMARI S.A.
de C.V.</p>

          <p style="color: #E72B81; font-weight: bold; font-size: 35px; line-height:
20px;">Recibo por servicios</p>

          <div style="margin-top: 50px;">
            <p style="color: #293270; font-weight: bold; letter-spacing:
3px;">FACTURA ID.° #{{ receipt.id }} | {{ receipt.created_at | formatDate }}</p>
          </div>
        </div>
      </div>
    </div>

    <hr style="border: 1px solid yellow;">

    <div class="row" style="margin-top: 20px;">
      <div class="col-12">
        <p style="color: #293270; font-weight: bold;">Factura a nombre de:</p>

        <div>
          <p style="font-weight: bold; font-size: 20px;">{{ user.name }}</p>

```

<p style="font-weight: bold; font-size: 20px;">{{ user.direction }}</p>  
</div>

<div class="row" style="margin-top: 50px;">

<div class="col-6">

<p style="color: #E72B81; font-weight: bold;">Descripcion</p>

<div>

<p>Suscripcion 6 meses de todos los cursos de Kumari + K-Box</p>

</div>

</div>

<div class="col-3 text-right">

<p style="color: #E72B81; font-weight: bold;">Precio unitario</p>

<div>

<p style="color: #293270">1 x \${{ receipt.amount }}</p>

</div>

</div>

<div class="col-3 text-right">

<p style="color: #E72B81; font-weight: bold;">Total</p>

<div>

<p style="color: #293270">Solo \${{ receipt.amount }}</p>

</div>

</div>

</div>

<hr style="border: 1px solid yellow;">

0.16) }}</p>

```
<div class="row" style="margin-top: 20px;">
  <div class="col-6">

  </div>
  <div class="col-3 text-right">
    <p style="color: #E72B81; font-weight: bold;">Subtotal</p>

    <p style="color: #E72B81; font-weight: bold;">IVA</p>

    <p style="color: #E72B81; font-weight: bold;">Total</p>
  </div>
  <div class="col-3 text-right">
    <p style="color: #293270">${{ receipt.amount - (receipt.amount *
0.16) }}</p>

    <p style="color: #293270">${{ receipt.amount * 0.16 }}</p>

    <p style="color: #293270">${{ receipt.amount }}</p>
  </div>
</div>

</div>
</div>

<!-- <div class="row" style="margin-top: 20px;">
  <div class="col-12">
    <p style="color: #E72B81; font-weight: bold;">Informacion del banco</p>

    <p style="color: #293270">Bancamex</p>
  </div>
</div>
```

```

    </div>

    <div class="row">

      <div class="col-12">

        <p style="color: #E72B81; font-weight: bold;">N.º de cuenta</p>

        <p style="color: #293270">ABC987654321</p>

      </div>

    </div> -->

  </div>

</div>



</section>

</VueHtml2pdf>

</div>

</div>

</div>

</q-page>

</template>

```

```

<script>

import { db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'

import moment from 'moment-timezone'

import axios from 'axios'

import VueHtml2pdf from 'vue-html2pdf'

export default {
  components: {

```



VueHtml2pdf

},

data(){

  return{

    receipts: [],

    filter: "",

    columns: [

      {

        name: 'created\_at',

        required: true,

        label: 'Fecha',

        align: 'left',

        field: row => row.created\_at,

        format: val => `\${val}`,

        sortable: true

      },

      { name: 'course', align: 'left', label: 'ID referencia', field: 'course', sortable: true },

      { name: 'amount', align: 'left', label: 'Precio', field: 'amount', sortable: true },

      { name: 'options', label: 'Opciones', field: 'options' }

    ],

    receipt: null,

  }

},

computed: {

  ...mapState('auth', [ 'user' ])

},

```

filters: {
  formatDate(args){
    moment.locale('es')

    let date = moment(args.toDate()).format('LL');

    return date
  }
},

created() {
  this.getReceipts()
},

methods: {
  generateReport () {
    this.$refs.html2Pdf.generatePdf()
  },

  async getReceipts(){
    try {
      let response = await db.collection('receipts')
        .where('user', '==', this.user.uid)
        .get()
        .then(query => {
          query.forEach(async doc => {
            let receipt = doc.data()
            receipt.id = doc.id

            this.receipts.push(receipt)
          })
        })
    } catch (error) {
      console.log(error)
    }
  }
}

```

```

        })
    })

    } catch (error) {
        console.log(error);
    }
},

showReceipt(receipt){
    this.receipt = receipt
    this.generateReport()
}
},
}
</script>

```

### Test.txt

```

<style scoped lang="scss">

    .main-container{

        background-color: white;

        border-radius: 10px;

        padding: 20px;

    }


    .btn-options{

        color: black;

    }


    .card{

        width: 900px;
    }

```

```
border-radius: 20px;
```

```
p{  
  font-size: 18px;  
}
```

```
.sec-1{  
  margin-top: 20px;  
}
```

```
.main-container{  
  position: relative;  
  background-color: white;  
  border-radius: 40px;  
  padding: 40px;  
  min-height: 55vh;  
  overflow: scroll;  
  display: flex;  
  justify-content: space-around;  
  align-items: center;  
  flex-direction: column;
```

```
-ms-overflow-style: none; /* IE and Edge */
```

```
scrollbar-width: none; /* Firefox */
```

```
&::-webkit-scrollbar {  
  display: none;  
}
```

```
.monito_1{  
    position: absolute;  
    top: 30%;  
    left: -5%;  
    width: 20%;  
}
```

```
.monito_2{  
    position: absolute;  
    top: 0;  
    right: -5%;  
    width: 20%;  
}
```

```
.monito_3{  
    position: absolute;  
    top: 20%;  
    left: -1%;  
    width: 12%;  
}
```

```
.monito_4{  
    position: absolute;  
    top: 25%;  
    right: -7%;  
    width: 22%;  
}
```

```
}
```

```
.form-group{  
  padding: 10px 5px;  
}
```

```
label{  
  font-size: 18px;
```

```
  input{  
    margin-right: 10px;  
  }  
}
```

```
input[type=text]{  
  width: 100%;  
  border-radius: 10px;  
  border: none;  
  border: 1px solid black;  
  padding: 5px;  
}
```

```
input[type=date]{  
  width: 100%;  
  border-radius: 10px;  
  border: none;  
  border: 1px solid black;  
  padding: 5px;  
}
```

```
input[type=password]{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
select{  
    width: 100%;  
    border-radius: 10px;  
    border: none;  
    border: 1px solid black;  
    padding: 5px;  
}
```

```
.btn-next{  
    background-color: #A0C62E;  
    min-width: 150px;  
    color: white;  
    border-radius: 10px;  
    padding: 8px 20px;  
    border: none;  
    margin-top: 20px;  
    cursor: pointer;  
    text-align: center;  
}
```

```
.btn-back{  
    width: 45px;  
    height: 45px;  
    border-radius: 50%;  
    padding: 5px;  
    border: none;  
    margin-top: 20px;  
    cursor: pointer;  
    display: flex;  
    justify-content: center;  
    align-items: center;
```

```
    img{  
        width: 45px;  
    }  
}
```

```
.title{  
    font-size: 2em;  
    font-weight: bold;  
    line-height: 20px;  
}  
}
```

```
.main{  
    background-image: url('/recursos_2/Verde.jpg');  
}
```

```
.pointer{
```



```
width: 10px;

height: 10px;

border-radius: 50%;

border: 1px solid #E72B81;

margin: 10px;
```

```
&.active{

    background-color: #E72B81;

}

}
```

```
// @media (max-width: 1023px) {

//   .card{

//     width: 100%;

//   }

// }

// }

</style>
```

```
<template>

<q-page>

  <div class="main-container">

    <div class="row">

      <div class="col-12 flex justify-center">

        <div>

          <div class="col-12 flex justify-center">

            <div class="card" v-if="step == 1">

              <div class="main-container">
```

```


<div class="row" style="width: 100%;">

  <div class="col-12 text-center">


    </div>

  </div>

  <div class="row" style="width: 100%;">

    <div class="col-12 text-center">

      <p class="title" style="font-size: 4em;">¡Hola!</p>

      <p>Vamos a determinar los tipos de inteligencia que posee tu niño(a)</p>

    </div>

  </div>

  <div class="row" style="width: 100%;">

    <div class="col-3">


    </div>

    <div class="col-6">

      <div class="flex justify-around items-center">


        <button class="btn-next" @click="step = 2">Continuar</button>

      </div>

    </div>

    <div class="col-3"></div>

    <div class="col-12 flex justify-center items-center">


      <span class="pointer active"></span>

      <span class="pointer"></span>

      <span class="pointer"></span>

    </div>

  </div>

</div>
```

</div>

</div>

</div>

</div>

<div class="card" v-if="step == 2">

<div class="main-container">



<div class="row" style="width: 100%;">

<div class="col-12 text-center">

<p class="title">Tipos de inteligencia</p>

<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-1"></div>

<div class="col-10">

<div class="sec-1">

<label>

<input v-model="intelligence1.option1" type="checkbox">Es muy bueno entendiéndose a sí mismo, reconoce sus puntos fuertes y sus debilidades, es muy bueno estableciendo objetivos.

</label>

<br>

<label>

<input v-model="intelligence1.option2" type="checkbox">Le gusta trabajar solo, reflexionar y seguir sus intereses.

</label>

<br>

<label>

☐Aprende mejor cuando trabaja solo, hace los proyectos a su ritmo, teniendo espacio o reflexionando.

</label>

<br>

</div>

</div>

<div class="col-1"></div>

</div>

<div class="row" style="width: 100%;">

<div class="col-3">

<div class="flex justify-start items-center">

<button class="btn-back" @click="step--">



</button>

</div>

</div>

<div class="col-6">

<div class="flex justify-around items-center">

<button class="btn-next" @click="step = 3">Continuar</button>

</div>

</div>

<div class="col-3"></div>

<div class="col-12 flex justify-center items-center">

<span class="pointer"></span>

<span class="pointer active"></span>

<span class="pointer"></span>

</div>

</div>

</div>

</div>

<div class="card" v-if="step == 3">

<div class="main-container">



<div class="row" style="width: 100%;">

<div class="col-12 text-center">

<p class="title">Tipos de inteligencia</p>

<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-1"></div>

<div class="col-10">

<div class="sec-1">

<label>

<input v-model="intelligence2.option1" type="checkbox" name="option2">Es muy bueno leyendo, escribiendo y narrando historias. Es muy bueno memorizando fechas y frases.

</label>

<br>

<label>

<input v-model="intelligence2.option2" type="checkbox" name="option2">Le gusta leer, escribir, contar historias/cuentos/chistes, jugar juegos con palabras y armar rompecabezas.

</label>

<br>

```
        <label>

            <input v-model="intelligence2.option3" type="checkbox"
name="option2">Aprende mejor leyendo, escuchando, viendo palabras, hablando, escribiendo,
discutiendo y debatiendo.
```

```
        </label>
```

```
        <br>
```

```
    </div>
```

```
</div>
```

```
<div class="col-1"></div>
```

```
</div>
```

```
<div class="row" style="width: 100%;">
```

```
    <div class="col-3">
```

```
        <div class="flex justify-start items-center">
```

```
            <button class="btn-back" @click="step--">
```

```
                
```

```
            </button>
```

```
        </div>
```

```
    </div>
```

```
<div class="col-6">
```

```
    <div class="flex justify-around items-center">
```

```
        <button class="btn-next" @click="step = 4">Continuar</button>
```

```
    </div>
```

```
</div>
```

```
<div class="col-3"></div>
```

```
<div class="col-12 flex justify-center items-center">
```

```
    <span class="pointer"></span>
```

```
    <span class="pointer active"></span>
```

```
    <span class="pointer"></span>
```

</div>

</div>

</div>

</div>

<div class="card" v-if="step == 4">

<div class="main-container">



<div class="row" style="width: 100%;">

<div class="col-12 text-center">

<p class="title">Tipos de inteligencia</p>

<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-1"></div>

<div class="col-10">

<div class="sec-1">

<label>

<input v-model="intelligence3.option1" type="checkbox" name="option3">Es muy bueno hablando con la gente, tiene muchos amigos y disfruta juntarse con la gente.

</label>

<br>

<label>

<input v-model="intelligence3.option2" type="checkbox" name="option3">Le gusta entender a las demás personas, liderando, organizando, comunicando y resolviendo conflictos.

</label>

```

        <br>

        <label>

            <input v-model="intelligence3.option3" type="checkbox"
name="option3">Aprende mejor al compartir, al relacionarse, al entrevistar y cooperar.

        </label>

        <br>

    </div>

</div>

<div class="col-1"></div>

</div>

<div class="row" style="width: 100%;">

    <div class="col-3">

        <div class="flex justify-start items-center">

            <button class="btn-back" @click="step--">

                

            </button>

        </div>

    </div>

    <div class="col-6">

        <div class="flex justify-around items-center">

            <button class="btn-next" @click="step = 5">Continuar</button>

        </div>

    </div>

</div>

<div class="col-3"></div>

<div class="col-12 flex justify-center items-center">

    <span class="pointer"></span>

    <span class="pointer active"></span>

```



<span class="pointer"></span>

</div>

</div>

</div>

</div>

<div class="card" v-if="step == 5">

<div class="main-container">



<div class="row" style="width: 100%;">

<div class="col-12 text-center">

<p class="title">Tipos de inteligencia</p>

<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-1"></div>

<div class="col-10">

<div class="sec-1">

<label>

<input v-model="intelligence4.option1" type="checkbox" name="option4">Es muy bueno en matemáticas, razonamiento lógico, robótica, resolución de problemas y programas virtuales.

</label>

<br>

<label>

<input v-model="intelligence4.option2" type="checkbox" name="option4">Les gusta experimentar, cuestionar, descubrir, trabajar con números, calcular y resolver problemas.

</label>

<br>

<label>

<input v-model="intelligence4.option3" type="checkbox"  
name="option4">Aprende mejor cuando hay cosas por explorar y pensar, cuando se clasifica y  
trabajan cosas abstractas.

</label>

<br>

</div>

</div>

<div class="col-1"></div>

</div>

<div class="row" style="width: 100%;">

<div class="col-3">

<div class="flex justify-start items-center">

<button class="btn-back" @click="step--">



</button>

</div>

</div>

<div class="col-6">

<div class="flex justify-around items-center">

<button class="btn-next" @click="step = 6">Continuar</button>

</div>

</div>

<div class="col-3"></div>

<div class="col-12 flex justify-center items-center">

<span class="pointer"></span>

```
<span class="pointer active"></span>
```

```
<span class="pointer"></span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="card" v-if="step == 6">
```

```
<div class="main-container">
```

```

```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-12 text-center">
```

```
<p class="title">Tipos de inteligencia</p>
```

```
<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>
```

```
</div>
```

```
</div>
```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-1"></div>
```

```
<div class="col-10">
```

```
<div class="sec-1">
```

```
<label>
```

```
<input v-model="intelligence5.option1" type="checkbox"
name="option5">Es muy bueno con las ubicaciones, lectura de mapas y dibujos, rompecabezas y
laberintos. Disfruta imaginando cosas y visualizando.
```

```
</label>
```

```
<br>
```

```
<label>
```

```
<input v-model="intelligence5.option2" type="checkbox"
name="option5">Le gusta diseñar, dibujar, crear, soñar despierto y mirar dibujos.
```

```
</label>
```

```
<br>
```

```
<label>
```

```
      <input v-model="intelligence5.option3" type="checkbox"
name="option5">Aprende mejor cuando trabaja con dibujos y colores, visualizando y usando su
ojo mental.
```

```
</label>
```

```
<br>
```

```
</div>
```

```
</div>
```

```
<div class="col-1"></div>
```

```
</div>
```

```
<div class="row" style="width: 100%;">
```

```
  <div class="col-3">
```

```
    <div class="flex justify-start items-center">
```

```
      <button class="btn-back" @click="step--">
```

```
        
```

```
      </button>
```

```
    </div>
```

```
</div>
```

```
<div class="col-6">
```

```
  <div class="flex justify-around items-center">
```

```
    <button class="btn-next" @click="step = 7">Continuar</button>
```

```
  </div>
```

```
</div>
```

```
<div class="col-3"></div>
```

```
<div class="col-12 flex justify-center items-center">
```

```
  <span class="pointer"></span>
```

```
<span class="pointer active"></span>
```

```
<span class="pointer"></span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="card" v-if="step == 7">
```

```
<div class="main-container">
```

```

```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-12 text-center">
```

```
<p class="title">Tipos de inteligencia</p>
```

```
<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>
```

```
</div>
```

```
</div>
```

```
<div class="row" style="width: 100%;">
```

```
<div class="col-1"></div>
```

```
<div class="col-10">
```

```
<div class="sec-1">
```

```
<label>
```

```
<input v-model="intelligence6.option1" type="checkbox"
name="option6">Es muy bueno en atletismo, baile, teatro, trabajos manuales y utilizando
herramientas físicas.
```

```
</label>
```

```
<br>
```

```
<label>
```

☐Le gusta moverse, tocar y hablar alto. Utiliza mucho su lenguaje corporal y es muy expresivo.

</label>

<br>

<label>

☐Aprende mejor tocando, moviéndose, experimentando y procesando información a través de sensaciones corporales.

</label>

<br>

</div>

</div>

<div class="col-1"></div>

</div>

<div class="row" style="width: 100%;">

<div class="col-3">

<div class="flex justify-start items-center">

<button class="btn-back" @click="step--">



</button>

</div>

</div>

<div class="col-6">

<div class="flex justify-around items-center">

<button class="btn-next" @click="step = 8">Continuar</button>

</div>

```
</div>
```

```
<div class="col-3"></div>
```

```
<div class="col-12 flex justify-center items-center">
```

```
    <span class="pointer"></span>
```

```
    <span class="pointer active"></span>
```

```
    <span class="pointer"></span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="card" v-if="step == 8">
```

```
  <div class="main-container">
```

```
    
```

```
    <div class="row" style="width: 100%;">
```

```
      <div class="col-12 text-center">
```

```
        <p class="title">Tipos de inteligencia</p>
```

```
        <p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>
```

```
      </div>
```

```
    </div>
```

```
    <div class="row" style="width: 100%;">
```

```
      <div class="col-1"></div>
```

```
      <div class="col-10">
```

```
        <div class="sec-1">
```

```
          <label>
```

```
            <input v-model="intelligence7.option1" type="checkbox"
name="option7">Es muy bueno al cantar, tiene ritmo y reconoce sonidos o canciones.
```

```

        </label>

        <br>

        <label>

            <input v-model="intelligence7.option2" type="checkbox"
name="option7">Le gusta escuchar música, cantar, tararear, silbar o tocar un instrumento.

        </label>

        <br>

        <label>

            <input v-model="intelligence7.option3" type="checkbox"
name="option7">Aprende mejor al escuchar música, recordando y repitiendo bajo un ritmo o
canción.

        </label>

        <br>

    </div>

</div>

<div class="col-1"></div>

</div>

<div class="row" style="width: 100%;">

    <div class="col-3">

        <div class="flex justify-start items-center">

            <button class="btn-back" @click="step--">

                

            </button>

        </div>

    </div>

    <div class="col-6">

        <div class="flex justify-around items-center">

            <button class="btn-next" @click="step = 9">Continuar</button>

        </div>

```



</div>

<div class="col-3"></div>

<div class="col-12 flex justify-center items-center">

<span class="pointer"></span>

<span class="pointer active"></span>

<span class="pointer"></span>

</div>

</div>

</div>

</div>

<div class="card" v-if="step == 9">

<div class="main-container">



<div class="row" style="width: 100%;">

<div class="col-12 text-center">

<p class="title">Tipos de inteligencia</p>

<p>Seleccione las afirmaciones que más se identifiquen con su hijo(a)</p>

</div>

</div>

<div class="row" style="width: 100%;">

<div class="col-1"></div>

<div class="col-10">

<div class="sec-1">

<label>

<input v-model="intelligence8.option1" type="checkbox"

name="option8">Es muy bueno entendiendo la naturaleza, haciendo preguntas, distinciones o identificando la flora y fauna del planeta.

</label>

<br>

<label>

<input v-model="intelligence8.option2" type="checkbox" name="option8">Le gusta realizar actividades al aire libre, las películas con personajes basados en animales, conservar la naturaleza y los proyectos de investigación que les permita explorar su medio ambiente.

</label>

<br>

<label>

<input v-model="intelligence8.option3" type="checkbox" name="option8">Aprende mejor trabajando en el medio natural, explorando los diferentes seres y cuando identifica temas relacionados con la naturaleza.

</label>

<br>

</div>

</div>

<div class="col-1"></div>

</div>

<div class="row" style="width: 100%;">

<div class="col-3">

<div class="flex justify-start items-center">

<button class="btn-back" @click="step--">



</button>

</div>

</div>

<div class="col-6">

<div class="flex justify-around items-center">

```
        <button class="btn-next" @click="results" style="background-color:
#24B8CB;">Ver Resultado</button>
```

```
    </div>
```

```
</div>
```

```
<div class="col-3"></div>
```

```
<div class="col-12 flex justify-center items-center">
```

```
    <span class="pointer"></span>
```

```
    <span class="pointer active"></span>
```

```
    <span class="pointer"></span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="card" v-show="step == 10">
```

```
    <div class="main-container">
```

```
        
```

```
        <div class="row" style="width: 100%;">
```

```
            <div class="col-12 text-center">
```

```
                </div>
```

```
        </div>
```

```
        <div class="row" style="width: 100%;">
```

```
            <div class="col-2"></div>
```

```
            <div class="col-8 text-center">
```

```
                <div class="row">
```

```
                    <canvas id="myChart" width="400" height="400"></canvas>
```

```

        </div>
    </div>
    <div class="col-2"></div>
</div>
<div class="row" style="width: 100%;">
    <div class="col-3">
        <div class="flex justify-start items-center">
            <button class="btn-back" @click="step--">
                
            </button>
        </div>
    </div>
    <div class="col-6">
        <div class="flex justify-around items-center">
            <button class="btn-next" @click="step = 1">Intentar de
nuevo</button>
        </div>
    </div>
    <div class="col-3"></div>
    <div class="col-12 flex justify-center items-center">
        <span class="pointer"></span>
        <span class="pointer"></span>
        <span class="pointer active"></span>
    </div>
</div>
</div>
</div>

```

```
        </div>
      </div>
    </div>
  </div>
</q-page>
</template>
```

```
<script>
import { db, storage, firebase } from 'boot/firebase'
const ref = storage.ref()

import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'
import axios from 'axios'
import VueHtml2pdf from 'vue-html2pdf'

export default {
  components: {
    VueHtml2pdf
  },

  data(){
    return{
      step: 1,
      myChart: null,
      intelligence1: {},
      intelligence2: {},
      intelligence3: {},
      intelligence4: {},
```

```
    intelligence5: {},  
    intelligence6: {},  
    intelligence7: {},  
    intelligence8: {},  
  }  
},
```

```
computed: {  
  ...mapState('auth', [ 'user' ]),  
  
  suma1(){  
    let total = 0  
  
    for (const property in this.intelligence1) {  
  
      if (this.intelligence1[property]) {  
        total = total + 33.33  
      }  
  
    }  
  
    return Math.round(total)  
  },  
  suma2(){  
    let total = 0  
  
    for (const property in this.intelligence2) {  
  
      if (this.intelligence2[property]) {  
        total = total + 33.33  
      }  
    }  
  }  
}
```

```

    }

    }
    return Math.round(total)
  },
  suma3(){
    let total = 0

    for (const property in this.intelligence3) {

      if (this.intelligence3[property]) {
        total = total + 33.33
      }

    }

    return Math.round(total)
  },
  suma4(){
    let total = 0

    for (const property in this.intelligence4) {

      if (this.intelligence4[property]) {
        total = total + 33.33
      }

    }

    return Math.round(total)
  },

```

```
suma5(){  
  let total = 0  
  
  for (const property in this.intelligence5) {  
  
    if (this.intelligence5[property]) {  
      total = total + 33.33  
    }  
  
  }  
  
  return Math.round(total)  
},  
suma6(){  
  let total = 0  
  
  for (const property in this.intelligence6) {  
  
    if (this.intelligence6[property]) {  
      total = total + 33.33  
    }  
  
  }  
  
  return Math.round(total)  
},  
suma7(){  
  let total = 0  
  
  for (const property in this.intelligence7) {
```



```

        if (this.intelligence7[property]) {
            total = total + 33.33
        }

    }

    return Math.round(total)
},

suma8(){
    let total = 0

    for (const property in this.intelligence8) {

        if (this.intelligence8[property]) {
            total = total + 33.33
        }

    }

    return Math.round(total)
},
},

```

```

methods: {
    async results(){
        this.step = 10

        let data = {
            labels: [
                'Intrapersonal',
                'Lingüística',

```

```

        'Interpersonal',
        'Lógico matemática',
        'Espacial',
        'Cinética',
        'Musical',
        'Naturalista'
    ],
    datasets: [{
        label: 'Mi inteligencia',
        data: [
            this.suma1,
            this.suma2,
            this.suma3,
            this.suma4,
            this.suma5,
            this.suma6,
            this.suma7,
            this.suma8
        ],
        fill: true,
        backgroundColor: 'rgba(255, 99, 132, 0.2)',
        borderColor: 'rgb(255, 99, 132)',
        pointBackgroundColor: 'rgb(255, 99, 132)',
        pointBorderColor: '#fff',
        pointHoverBackgroundColor: '#fff',
        pointHoverBorderColor: 'rgb(255, 99, 132)'
    }]
};

```

```
const ctx = document.getElementById(`myChart`);  
  
if (this.myChart) {  
  this.myChart.destroy();  
}
```

```
this.myChart = new Chart(ctx, {  
  type: 'radar',  
  data: data,  
  options: {  
    elements: {  
      line: {  
        borderWidth: 3  
      }  
    },  
    scale: {  
      ticks: {  
        beginAtZero: true,  
        max: 100,  
        min: 0,  
        stepSize: 10  
      }  
    }  
  },  
});
```

```
let sumas = [  
  this.suma1,  
  this.suma2,  
  this.suma3,
```

```
    this.suma4,  
    this.suma5,  
    this.suma6,  
    this.suma7,  
    this.suma8  
  ]  
}
```

```
    await db.collection('users').doc(this.user.uid).update({  
      test: sumas  
    })  
  },  
},  
}  
</script>
```

### Actions.txt

```
import { auth, db } from 'boot/firebase'
```

```
import { Notify } from 'quasar'
```

```
import moment from 'moment'
```

```
export async function loginUser ({commit}, payload) {  
  console.log('Desde vuex: ', payload);  
  try {  
    return new Promise((resolve, reject) =>{  
      auth.signInWithEmailAndPassword(payload.email, payload.password)  
        .then(response => {  
          resolve(response)  
        }).catch(error => {  
          reject(error)  
        })  
    })  
  }  
}
```

```

    })
  } catch (error) {
    console.log(error);
  }
}

```

```

export async function passwordReset ({commit}, payload) {
  console.log('Desde vuex: ', payload);
  try {
    return new Promise((resolve, reject) =>{
      auth.sendPasswordResetEmail (payload)
        .then(response => {
          resolve(response)
        }).catch(error => {
          reject(error)
        })
    })
  } catch (error) {
    console.log(error);
  }
}

```

```

export async function registerUser ({commit}, payload) {
  console.log('Desde vuex: ', payload);
  try {
    return new Promise((resolve, reject) =>{
      auth.createUserWithEmailAndPassword(payload.profile.email, payload.profile.password)

```

```

.then(async (userCredential) => {
  console.log(userCredential);

  let user = payload.profile
  user.uid = userCredential.user.uid
  user.partner_number = moment().unix()

  let mail = {
    to: payload.profile.email,
    subject: 'Registro KUMARI',
    message: {
      text: '¡Bienvenido a la comunidad Kumari!',
      html: `
        

        <p>¡Bienvenido a la comunidad Kumari!</p>

        <br />

        <p>En este espacio seguro podrás encontrar sorpresas, nuevas formas de
aprender y sobre todo mucha diversión. Te invitamos a descargar la aplicación oficial en el
siguiente link: <a
href="https://play.google.com/store/apps/details?id=com.kumari.kumari">Link</a></p>

        <p><strong>¡Suscríbete y obtén tu K-Box hasta la puerta de tu
casa!</strong></p>

        <p>¿Tienes dudas? Escríbenos por wpp al 614-463-27-64</p>

        

```

```

    }
  }

  await db.collection('users').doc(userCredential.user.uid).set(user)
  await db.collection('mails').add(mail)

  // this.$router.replace('/')

  await auth.signInWithEmailAndPassword(payload.profile.email,
    payload.profile.password)
    .then(async response => {
      console.log('Fierestore dice: ', response);

    }).catch(error => {
      console.log(error);
    })

    resolve(userCredential)

  })
  .catch((error) => {
    console.log(error);

    reject(error)
  });
})
} catch (error) {
  console.log(error);
}

```

```
}
```

```
export async function getUser({commit, dispatch}, payload){  
  try {  
    let document = await db.collection('users').doc(payload).get()  
  
    if (document.exists) {  
      let user = document.data()  
      console.log('Este es el user: ', user);  
  
      commit('updateUser', user)  
  
      dispatch('getCourses', user.uid)  
  
      if (user.role == 'user') {  
  
        if (!user.complete_profile) {  
          this.$router.replace('/my-first-collection')  
          return  
        }  
        this.$router.replace('/my-account')  
        return  
      }else if(user.role == 'instructor'){  
        this.$router.replace('/my-courses')  
        return  
      }  
  
      this.$router.replace('/dashboard')
```



```
    }  
  } catch (error) {  
    console.log(error)  
  }  
}
```

```
export async function getCourses({ commit }, payload) {  
  try {  
    let list_courses = []  
  
    let response = await db.collection('shopping')  
      .where('user', '==', payload)  
      .get()  
      .then(query => {  
        query.forEach(async course => {  
          let co = await db.collection('courses')  
            .doc(course.data().course)  
            .get()  
  
          if (co.exists) {  
            let co_data = co.data()  
            co_data.id = co.id  
  
            list_courses.push(co_data)  
          }  
        })  
      })  
  }  
}
```

```

        })
    })

    commit('updateListCourses', list_courses)
  } catch (error) {
    console.log(error);
  }
}

```

```

export function closeSesion({ commit }) {
  auth.signOut()
  commit('updateUser', null)
  this.$router.replace('/')
}

```

## Dashboard.txt

```
<style lang="scss">
```

```

.card{
  padding: 10px;
  width: 100%;

  .card-content{
    padding: 10px;
    width: 100%;
    min-height: 300px;
    background-color: white;
    border-radius: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
  }
}

```

```
flex-direction: column;
```

```
.card-counter{  
    font-size: 32px;  
    font-weight: bold;  
}  
}  
}  
</style>
```

```
<template>  
<q-page class="">  
    <div class="row">  
        <div class="col-4 card">  
            <div class="card-content shadow-2">  
                <p class="card-counter">{{ users.length }}</p>  
                <p class="card-name">Usuarios Totales</p>  
            </div>  
        </div>  
        <div class="col-4 card">  
            <div class="card-content shadow-2">  
                <p class="card-counter">{{ users.length }}</p>  
                <p class="card-name">Usuarios Activos</p>  
            </div>  
        </div>  
        <div class="col-4 card">  
            <div class="card-content shadow-2">  
                <p class="card-counter">0</p>  
                <p class="card-name">Usuarios por Renovar</p>
```

```

        </div>
    </div>
    <div class="col-4 card">
        <div class="card-content shadow-2">
            <p class="card-counter">{{ courses.length }}</p>
            <p class="card-name">Cursos Activos</p>
        </div>
    </div>
</div>
</q-page>
</template>

```

```

<script>
import { firebase, db, firestore, storage } from 'boot/firebase'
const ref = storage.ref()
import { mapState, mapActions } from 'vuex'
import moment from 'moment-timezone'

export default {
  name: 'Dashboard',

  data(){
    return{
      users: [],
      courses: [],
    }
  },

  created(){

```

```
    this.getUsers()
    this.getCourses()
  },
```

```
  methods: {
```

```
    async getUsers(){
```

```
      try {
```

```
        let response = await db.collection('users')
```

```
          .where('role', '==', 'user')
```

```
          .onSnapshot(query => {
```

```
            this.users = []
```

```
            query.forEach(async doc => {
```

```
              let user = doc.data()
```

```
              user.id = doc.id
```

```
              this.users.push(user)
```

```
            })
```

```
          })
```

```
        } catch (error) {
```

```
          console.log(error);
```

```
        }
```

```
      },
```

```
    async getCourses(){
```

```
      this.courses = []
```

```
      try {
```

```

    let response = await db.collection('courses')
      .get()
      .then(query => {
        query.forEach(async doc => {
          let course = doc.data()
          course.id = doc.id
          this.courses.push(course)
        })
      })
  } catch (error) {
    console.log(error);
  }
},
}
}
</script>

```

## Index.txt

```
<style lang="scss">
```

```

.main-container{
  background-color: white;
  border-radius: 10px;
  padding: 20px;
}

```

```

.btn-options{
  border: none;
  margin: 5px;
  font-size: 18px;
  color: white;
}

```

```
border-radius: 5px;  
cursor: pointer;  
}
```

```
.btn-modal{  
border: none;  
background-color: #24B8CB;  
color: white;  
border-radius: 15px;  
padding: 10px 15px;  
font-weight: bold;  
font-size: 17px;  
cursor: pointer;  
}
```

```
.btn-cancel{  
color: white;  
padding: 5px 20px;  
border: none;  
background-color: #E72B81;  
border-radius: 10px;  
cursor: pointer;  
}
```

```
.titles{  
font-size: 18px;  
font-weight: bold;  
}
```

```
.btn-submit{  
    color: white;  
    padding: 5px 20px;  
    border: none;  
    background-color: #24B8CB;  
    border-radius: 10px;  
    margin-left: 10px;  
    cursor: pointer;  
}
```

```
.daily_background{
```

```
.course-body{  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    background-color: white;  
    border-radius: 10px;  
    aspect-ratio: 16/9;  
    position: relative;  
    object-fit: cover;  
    object-position: center;
```

```
img{  
    width: 80%;  
    object-fit: cover;  
    object-position: center;  
    aspect-ratio: 16/9;
```



```
}
```

```
.layer{
```

```
  position: absolute;
```

```
  top: 0;
```

```
  left: 0;
```

```
  width: 100%;
```

```
  height: 100%;
```

```
  border-radius: 10px;
```

```
  background-color: rgba(0, 0, 0, 0.5);
```

```
  display: flex;
```

```
  justify-content: center;
```

```
  align-items: center;
```

```
}
```

```
.layer-disabled{
```

```
  background-color: rgba(0, 0, 0, 0.7);
```

```
}
```

```
}
```

```
}
```

```
</style>
```

```
<template>
```

```
<q-page>
```

```
<div class="main-container">
```

```
<div class="row">
```

```
<div class="col-12 q-pa-sm">
```

```
<q-table
```

```
  title="Frases"
```

```

      :data="tickets"

      :columns="columns"

      :filter="filter"

      row-key="name"

      flat
    >
    <template v-slot:top>
      <div class="flex justify-between items-center" style="width: 100%;">
        <span style="font-size: 25px; font-weight: bold;">Tickets</span>

        <button class="btn-modal cursor-pointer" @click="modalQuote = true">Crear
ticket</button>

        <q-input style="width: 500px;" outlined dense debounce="300" label="Buscar"
color="primary" v-model="filter">

          <template v-slot:append>
            <q-icon name="search" />
          </template>
        </q-input>
      </div>
    </template>
    <template v-slot:body-cell-options="props">
      <q-td :props="props">
        <button class="btn-options" style="background-color: #F3920B;"
@click="deleteQuote(props.row)">

          <q-icon name="delete" />
        </button>
      </q-td>
    </template>
  </q-table>
</div>
<!-- <div class="col-4 q-pa-sm">

```

```

<div class="flex justify-between items-center">
  <p class="titles">Frases</p>

  <button class="btn-submit" @click="modalQuote = true">Agregar</button>
</div>

<div class="q-mt-md">
  <q-list bordered separator>
    <q-item clickable v-ripple v-for="(ticket, index) in tickets" :key="index">
      <q-item-section>{{ ticket.ticket }}</q-item-section>
      <q-item-section avatar>
        <button class="btn-cancel" @click="deleteQuote(ticket,
index)">Eliminar</button>
      </q-item-section>
    </q-item>
  </q-list>
</div>
</div> -->

<div class="col-12 col-md-6 q-pa-sm">
  <div class="flex justify-between items-center">
    <p class="titles">Activacion diaria</p>

    <button class="btn-modal" @click="modalDailyAdd = true">Agregar</button>
  </div>

  <div class="q-mt-md">
    <q-list bordered separator>
      <draggable v-model="daily" group="daily" @change="updateOrder">
        <q-item clickable v-ripple v-for="(video, index) in daily" :key="index">
          <q-item-section @click="modalDaily = true, dailySelect = video">Activacion {{
video.order }}</q-item-section>
          <q-item-section avatar>

```

```

        <button class="btn-cancel" @click="modalEditDaily = true, editDaily =
video">Actualizar</button>

    </q-item-section>

</q-item>

</draggable>

</q-list>

</div>

</div>

<div class="col-12 col-md-6 q-pa-sm">

    <div>

        <div class="daily_background">

            <div class="course-body">

                

                <div class="layer" v-if="uploading">

                    

                </div>

            </div>

        </div>

    </div>

    </div>

    <div style="margin-top: 30px;" class="text-center">

        <label for="daily" class="btn-modal">Cambiar imagen</label>

        <input id="daily" type="file" style="display: none;" @change="getImage($event)"
accept="image/">

    </div>

</div>

</div>

```

</div>

<q-dialog v-model="modalQuote" persistent>

<q-card style="min-width: 500px; border-radius: 20px;">

<q-card-section class="row">

<div class="col-12">

<q-form

@submit="addQuote"

class="q-gutter-md"

>

<div class="row">

<div class="col-12">

<div class="form-group">

<q-input v-model="newQuote.ticket" type="text" label="Frase" />

</div>

</div>

</div>

<div class="q-pa-md flex justify-end">

<button class="btn-cancel" type="button" @click="modalQuote = false">Cerrar</button>

<button class="btn-submit" type="submit">Guardar</button>

</div>

</q-form>

</div>

</q-card-section>

</q-card>

</q-dialog>

<q-dialog v-model="modalDailyAdd" persistent>

```

<q-card style="min-width: 500px; border-radius: 20px;">
  <q-card-section class="row">
    <div class="col-12">
      <q-form
        @submit="addDaily"
        class="q-gutter-md"
      >
        <div class="row">
          <div class="col-12">
            <div class="form-group">
              <p class="titles">Agregar nueva activacion</p>
              <input ref="inputVideo" type="file" @change="getVideo($event)">
            </div>
          </div>
        </div>
        <div class="q-pa-md flex justify-end">
          <button class="btn-cancel" type="button" @click="modalDailyAdd =
false">Cerrar</button>
          <button class="btn-submit" type="submit">Guardar</button>
        </div>
      </q-form>
    </div>
  </q-card-section>
</q-card>
</q-dialog>

```

```

<q-dialog v-model="modalEditDaily" persistent v-if="editDaily != null">
  <q-card style="min-width: 500px; border-radius: 20px;">
    <q-card-section class="row">

```

```

<div class="col-12">
  <q-form
    @submit="addDailyEdit"
    class="q-gutter-md"
  >
    <div class="row">
      <div class="col-12">
        <div class="form-group">
          <p class="titles">Editar activacion {{ editDaily.order }}</p>
          <input ref="inputVideo" type="file" @change="getVideo($event)">
        </div>
      </div>
    </div>
    <div class="q-pa-md flex justify-end">
      <button class="btn-cancel" type="button" @click="modalEditDaily =
false">Cerrar</button>
      <button class="btn-submit" type="submit">Guardar</button>
    </div>
  </q-form>
</div>
</q-card-section>
</q-card>
</q-dialog>

<q-dialog v-model="modalDaily" persistent>
  <q-card>
    <q-card-section>
      <div class="text-h6">Activacion diaria {{ dailySelect.order }}</div>
    </q-card-section>

```

```

        <q-card-section class="q-pt-none">

            <video controls controlsList="nodownload" width="100%"
:src="dailySelect.url"></video>

        </q-card-section>

        <q-card-actions align="right">

            <q-btn flat label="Cerrar" color="primary" v-close-popup />

        </q-card-actions>

    </q-card>

</q-dialog>

    <ProgressComponent :upload="upload" :progress="progress" :progress-label="progressLabel"
:file="dailyVideo" />

</q-page>
</template>

```

```

<script>

import { firebase, db, storage } from 'boot/firebase'

const ref = storage.ref()

import { mapState, mapActions } from 'vuex'

import moment from 'moment-timezone'

import ProgressComponent from './instructor/ProgressComponent'

import draggable from 'vuedraggable'

export default {

  name: 'Dashboard',

  components: {

    ProgressComponent,


```



```
    draggable
  },

  data(){
    return{
      uploading: false,

      modalQuote: false,
      modalDailyAdd: false,
      modalDaily: false,

      modalEditDaily: false,
      editDaily: null,

      newQuote: {
        ticket: ""
      },

      newDaily: {
        url: "",
        order: 0
      },

      tickets: [],
      daily: [],

      dailySelect: {},

      dailyVideo: null,
```

```

        upload: false,
        progress: null,
        progressLabel: "",

        filter: "",
        columns: [
            {
                name: 'ticket',
                required: true,
                label: 'Ticket',
                align: 'left',
                field: row => row.ticket,
                format: val => `${val}`,
                sortable: true
            },
            { name: 'options', label: 'Opciones', field: 'options' }
        ],

        daily_background: "",
    },

    created(){
        this.getQuotes()
        this.getDaily()

        this.getDailyBG()
    },

```

```
filters: {

},

methods: {

  async getDailyBG(){

    try {

      let response = await db.collection('daily_background').doc('current').get()

      this.daily_background = response.data()

    } catch (error) {

      console.log(error);

    }

  },

  updateOrder(){

    this.daily.map((daily, index) => {

      daily.order = index + 1

      this.updateDaily(daily)

    })

  },

  async updateDaily(daily){

    try {

      let response = await db.collection('daily_activation').doc(daily.id).update({ order:
daily.order })

    } catch (error) {

      console.log(error);

    }

  }

}
```

```
},
```

```
async addDaily(){
  try {
    let response = await db.collection('daily_activation').add(this.newDaily)

    this.modalDailyAdd = false
    this.newDaily = {
      url: "",
      order: 0,
    }

    this.$refs.inputVideo.value = ""

    this.$q.notify({
      message: 'Activacion agregada',
      color: 'positive'
    })

    this.getDaily()
  } catch (error) {
    console.log(error);
  }
},
```

```
async addDailyEdit(){
  try {
    let response = await db.collection('daily_activation')
      .doc(this.editDaily.id)
```

```
.update(this.editDaily)
```

```
this.$q.notify({  
  message: 'Activacion editada',  
  color: 'positive'  
})
```

```
this.$refs.inputVideo.value = "
```

```
this.modalEditDaily = false
```

```
this.editDaily = null
```

```
  } catch (error) {  
    console.log(error);  
  }  
},
```

```
async addQuote(){  
  try {  
    let response = await db.collection('tickets').add(this.newQuote)  
  
    this.newQuote.ticket = "  
  
    this.modalQuote = false  
  
    this.$q.notify({  
      message: 'Frase agregada',  
      color: 'positive'  
    })
```

```

        this.getQuotes()
    } catch (error) {
        console.log(error);
    }
},

async deleteQuote(ticket){
    try {
        let response = await db.collection('tickets').doc(ticket.id).delete()

        let index = this.tickets.findIndex(doc => doc == ticket)
        this.tickets.splice(index, 1)

        this.$q.notify({
            message: 'Frase eliminada',
            color: 'positive'
        })

    } catch (error) {
        console.log(error);
    }
},

async getQuotes(){
    this.tickets = []

    try {
        let response = await db.collection('tickets')
            .get()
    }

```

```

        .then(query => {
            query.forEach(doc => {
                let ticket = doc.data()
                ticket.id = doc.id

                this.tickets.push(ticket)
            });
        })
    } catch (error) {
        console.log(error);
    }
},

changeDaily(){

},

async getDaily(){
    this.daily = []

    try {
        let response = await db.collection('daily_activation')
            .get()
            .then(query => {
                query.forEach(doc => {
                    if (doc.id !== 'current' && doc.id !== 'counter') {
                        let daily = doc.data()
                        daily.id = doc.id

```

```

        this.daily.push(daily)
    }
});

    this.daily.sort((a,b) => (a.order > b.order) ? 1 : ((b.order > a.order) ? -1 :
0))

    })
} catch (error) {
    console.log(error);
}
},

getVideo(e){
    this.dailyVideo = e.target.files[0]
    console.log(this.dailyVideo);

    this.uploadVideo(this.dailyVideo)
},

async uploadVideo(file){
    this.upload = true
    let ext = file.name.substring(file.name.lastIndexOf('.')+1, file.name.length) || file.name
    try {
        let metadata = {
            contentType: file.type,
            contentLanguage: "es",
        }
        let fileId = this.generateUUID()
        const refFile = ref.child('vid/' + fileId + '.' + ext)

```



```

var task = refFile.put(file, metadata);

task.on('state_changed',
  (snapshot) => {
    // Observe state change events such as progress, pause, and resume

    // Get task progress, including the number of bytes uploaded and the total number
of bytes to be uploaded

    this.progress = ((snapshot.bytesTransferred / snapshot.totalBytes) * 100) / 100;

    this.progressLabel = Math.trunc((this.progress * 100)) + '%'

    console.log('Upload is ' + (this.progress * 100) + '% done');

    switch (snapshot.state) {
      case firebase.storage.TaskState.PAUSED: // or 'paused'
        console.log('Upload is paused');

        break;

      case firebase.storage.TaskState.RUNNING: // or 'running'
        console.log('Upload is running');

        break;
    }
  },
  (error) => {
    // Handle unsuccessful uploads
  },
  () => {

    // Handle successful uploads on complete
    // For instance, get the download URL: https://firebasestorage.googleapis.com/...

    task.snapshot.ref.getDownloadURL().then(async (downloadURL) => {
      console.log('File available at', downloadURL);

      this.newDaily.url = downloadURL
    })
  }
)

```

```

        this.newDaily.order = this.daily.length + 1

        if (this.editDaily != null) {
            this.editDaily.url = downloadURL
        }

        this.upload = false
    });
}
);
} catch (error) {
    console.log(error);
}
},

getImage(e){
    this.uploading = true
    this.uploadImage(e.target.files[0])
},

async uploadImage(image){
    // this.isLoading = true;
    try {
        let metadata = {
            contentType: "image/jpeg",
            contentLanguage: "es",
        }

        let fotold = this.generateUUID()
        const refImg = ref.child('daily_bg/' + fotold + '.jpg')

```

```

return refImg.put(image, metadata)

.then(e => {
  console.log(e)
  return ref.child('daily_bg/' + fotold + '.jpg').getDownloadURL()
  .then(async (url) => {

    await db.collection('daily_background').doc('current').update({ url: url })

    this.daily_background.url = url

    this.$q.notify({
      message: 'Imagen actualizada',
      color: 'positive'
    })

    this.uploading = false
  })
  // this.getResizePath(fotold)
})
.catch(error => console.log(error))
} catch (error) {
  console.log(error);
}
},

//Generar UUID
generateUUID(){
  var h = ['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];

```

```

        var k = ['x','x','x','x','x','x','x','x','-','x','x','x','x','-','4','x','x','x','x','-','y','x','x','x','x','-','x','x','x','x','x','x','x','x','x','x','x','x','x','x'];

        var u = "",i=0,rb=Math.random()*0xffffffff|0;

        while(i++<36) {

            var c=k[i-1],r=rb&0xf,v=c=='x'?r:(r&0x3|0x8);

            u+=(c=='-' || c=='4')?c:h[v];rb=i%8==0?Math.random()*0xffffffff|0:rb>>4

        }

        return u

    },

}

}

</script>

```

### Mutations.txt

```

export function updateUser (state, payload) {

    state.user = payload

}

export function updateBgUser (state, payload) {

    state.user.background_image = payload

}

export function updatePhotoUser (state, payload) {

    state.user.photo = payload

}

export function updateListCourses (state, payload) {

    state.courses = payload

}

```

### Routes.txt

```

const routes = [

```

```
{
  path: '/',
  component: () => import('layouts/AuthLayout.vue'),
  children: [
    { path: '', component: () => import('pages/auth/Login.vue') }
  ]
},
{
  path: '/register',
  component: () => import('layouts/AuthLayout.vue'),
  children: [
    { path: '', component: () => import('pages/auth/Register.vue') }
  ]
},
{
  path: '/forgot-password',
  component: () => import('layouts/AuthLayout.vue'),
  children: [
    { path: '', component: () => import('pages/auth/ForgotPassword.vue') }
  ]
},
{
  path: '/complete-profile',
  component: () => import('layouts/AuthLayout.vue'),
  children: [
    { path: '', component: () => import('pages/auth/Profile.vue') }
  ]
},
{
```

```

    path: '/my-first-collection',
    component: () => import('layouts/AuthLayout.vue'),
    children: [
      { path: '', component: () => import('pages/market/FirstCollection.vue') }
    ]
  },
  {
    path: '/dashboard',
    component: () => import('layouts/MainLayout.vue'),
    children: [
      { path: '', component: () => import('pages/Index.vue') },
      { path: '/my-courses', component: () => import('pages/instructor/Courses.vue') },
      { path: '/my-clases', component: () => import('pages/instructor/Clases.vue') },
    ]
  },
  {
    path: '/admin',
    component: () => import('layouts/MainLayout.vue'),
    children: [
      { path: '/users', component: () => import('pages/admin/Users.vue') },
      { path: '/orders', component: () => import('pages/admin/Orders.vue') },
      { path: '/instructors', component: () => import('pages/admin/Instructors.vue') },
      { path: '/all-courses', component: () => import('pages/admin/Courses.vue') },
      { path: '/all-lessons', component: () => import('pages/admin/Lessons.vue') },
      { path: '/questions', component: () => import('pages/admin/Questions.vue') },
      { path: '/inicio', component: () => import('pages/Dashboard.vue') },
    ]
  },
  {

```

```

    path: '/user',
    component: () => import('layouts/MainLayout.vue'),
    children: [
      { path: '/my-account', component: () => import('pages/user/Account.vue') },
      { path: '/my-activity', component: () => import('pages/user/Activity.vue') },
      { path: '/receipts', component: () => import('pages/user/Receipts.vue') },
      { path: '/test', component: () => import('pages/user/Test2.vue') },
      { path: '/market', component: () => import('pages/market/Market.vue') },
      { path: '/frequent-questions', component: () => import('pages/user/Questions.vue') },
    ]
  },

  // Always leave this as last one,
  // but you can also remove it
  {
    path: '*',
    component: () => import('pages/Error404.vue')
  }
]

```

export default routes

[State.txt](#)

```

export default function () {
  return {
    user: null,
    courses: null,
  }
}

```