

Examen-nuevas-categorias-resuelt...



Acrub_ac



Introducción a la Ingeniería del Software y los Sistemas de Información II



2º Grado en Ingeniería Informática - Ingeniería del Software



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla**



La mejor escuela de negocios en energía, sostenibilidad y medio ambiente de España.

Más información
www.eoi.es

Formamos
talento para un futuro
Sostenible



100% Empleabilidad



Modalidad: Presencial u online



**Programa de Becas,
Bonificaciones y Descuentos**

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

IISII2: Evaluación individual laboratorio 2022-2023

Preparación del entorno

Windows:

- Abra un terminal y ejecute el comando `npm run install:all:win`.

Linux/macOS:

- Abra un terminal y ejecute el comando `npm run install:all:linux`.

Ejecución y depuración

- Para **ejecutar el backend**, abra un terminal y ejecute el comando `npm run start:backend`.
- Para **ejecutar el frontend**, abra un nuevo terminal y ejecute el comando `npm run start:frontend`.
- Para **depurar el backend**, asegúrese de que **NO** existe una instancia en ejecución, pulse en el botón **Run and Debug** de la barra lateral, seleccione **Debug Backend** en la lista desplegable, y pulse el botón de **Play**.
- Para **depurar el frontend**, asegúrese de que **EXISTE** una instancia en ejecución, pulse en el botón **Run and Debug** de la barra lateral, seleccione **Debug Frontend** en la lista desplegable, y pulse el botón de **Play**.
- Para **rehacer las migraciones y seeders del backend**, abra un terminal y ejecute el comando `npm run migrate:backend`.

Enunciado - Creación de categorías de restaurantes personalizadas

Realice las modificaciones que considere necesarias, tanto en backend como en frontend, para satisfacer los nuevos requisitos que a continuación se describen.

Se desea permitir a los dueños de restaurantes crear sus propias categorías de restaurantes. Para ello, en la pantalla de creación de restaurantes se incluirá un botón para acceder a una nueva pantalla que permitirá introducir un nuevo nombre de categoría de restaurante (ver capturas). Puede usar este icono para el botón:

```
<MaterialCommunityIcons name='folder-plus-outline' color='{white}' size={20} />
```

Al volver a la pantalla de creación de restaurantes tras introducir la nueva categoría, dicha categoría debe estar disponible en la lista desplegable de categorías de restaurantes.

No se debe permitir la creación de una categoría que ya existiera. En dicho caso, el Backend debe responder con un error que será visualizado en la pantalla de creación de categorías de restaurantes al pulsar el botón de submit. Además, el tamaño máximo para los nombres de las categorías de restaurante será de 50 caracteres. Esta restricción debe comprobarse tanto a nivel de formulario en el Frontend como a nivel de Backend.

← Create Restaurant

Name:

Rollito's

Description:

Comida mejicana

Address:

Calle Pajaritos, 8

Postal code:

41008

Url:

http://www.rollitosrestaurant.es

Shipping costs:

5

Email:

rollitos@rollitosrestaurant.es

Phone:

675562456

Select the restaurant category

▼

New category

Logo:

My Restaurants

Control Panel

Profile

← Create Restaurant Category

Name:

Mexican food

Save

My Restaurants

Control Panel

Profile

← Create Restaurant

Name:

Rollito's

Description:

Comida mejicana

Address:

Calle Pajaritos, 8

Postal code:

41008

Url:

http://www.rollitosrestaurant.es

Shipping costs:

5

Email:

rollitos@rollitosrestaurant.es

Phone:

675562456

Select the restaurant category

^

Spanish

Fast food

Mexican food

My Restaurants

Control Panel

Profile

← Create Restaurant Category

Name:

Fast food

name-The category Fast food already exists.

Save

My Restaurants

Control Panel

Profile

Procedimiento para la entrega

1. Elimine las carpetas node_modules de los proyectos de backend y frontend y la carpeta .expo del proyecto frontend.
2. Cree un zip que incluya ambos proyectos. **OJO: Compruebe que el zip resultante no es el mismo que descargó y que por lo tanto incluye sus soluciones**
3. Avise a su profesor antes de realizar el envío.
4. Cuando tenga el visto bueno de su profesor, envíe el fichero resultante a través de enseñanza virtual.

Imagínate aprobando el examen

Necesitas tiempo y concentración

Planes	 PLAN TURBO	 PLAN PRO	 PLAN PRO+
 Descargas sin publi al mes	10 	40 	80 
 Elimina el video entre descargas			
 Descarga carpetas			
 Descarga archivos grandes			
 Visualiza apuntes online sin publi			
 Elimina toda la publi web			
 Precios Anual <input type="checkbox"/>	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo,
¿Qué nota vas a sacar?

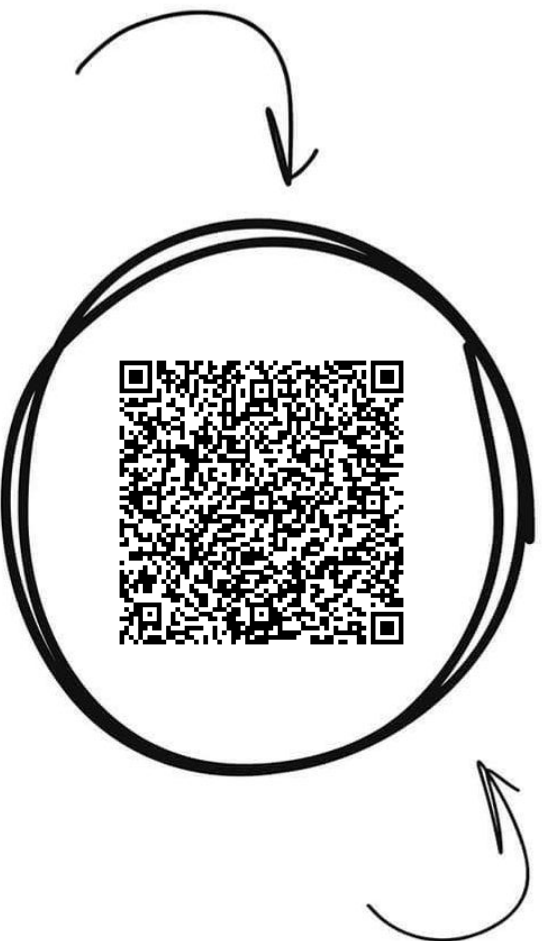


WUOLAH

Introducción a la Ingeniería...



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Banco de apuntes de la

WUOLAH

1 Imprime esta hoja

2 Recorta por la mitad

3 Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

4 Llévate dinero por cada descarga de los documentos descargados a través de tu QR



Ficheros a modificar:

RestaurantCategoryValidation.....	4
CreateRestaurantScreen.....	5
RestaurantCategoryController.....	13
RestaurantCategoryRoutes.....	14
RestaurantEndpoints.....	15
CreateRestaurantScreen.....	16
CreateRestaurantCategoryScreen.....	24
RestaurantStack.....	27

Las partes de cada fichero correspondientes a la solución están delimitadas por las etiquetas:

```
// SOLUCIÓN-----  
Código  
// -----SOLUCIÓN
```


Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

RestaurantCategoryValidation

```
// SOLUCIÓN-----  
import { RestaurantCategory } from '../models/models.js'  
import { check } from 'express-validator'  
  
const checkNonExistingCategory = async (value, req) => {  
  try {  
    const category = await RestaurantCategory.findOne({ where: { name: value } })  
  }  
  if (category) {  
    return Promise.reject(new Error('This category already exists'))  
  }  
  return Promise.resolve()  
} catch (err) {  
  return Promise.reject(new Error(err))  
}  
  
const create = [  
  
  check('name').isString().isLength({ min: 1, max: 50 }).trim(),  
  check('name').custom((value, { req }) => {  
    return checkNonExistingCategory(value, req)  
  })  
  
]  
  
export { create }  
// -----SOLUCIÓN
```

CreateRestaurantScreen

```
import React, { useEffect, useState } from 'react'
import { Image, Platform, Pressable, ScrollView, StyleSheet, View } from
'react-native'
import * as ExpoImagePicker from 'expo-image-picker'
import { MaterialCommunityIcons } from '@expo/vector-icons'
import * as yup from 'yup'
import DropDownPicker from 'react-native-dropdown-picker'
import { create, getRestaurantCategories } from
'../../api/RestaurantEndpoints'
import InputItem from '../../components/InputItem'
import TextRegular from '../../components/TextRegular'
import * as GlobalStyles from '../../styles/GlobalStyles'
import restaurantLogo from '../../assets/restaurantLogo.jpeg'
import restaurantBackground from '../../assets/restaurantBackground.jpeg'
import { showMessage } from 'react-native-flash-message'
import { ErrorMessage, Formik } from 'formik'
import TextError from '../../components/TextError'

export default function CreateRestaurantScreen ({ navigation, route }) {
  const [open, setOpen] = useState(false)
  const [restaurantCategories, setRestaurantCategories] = useState([])
  const [backendErrors, setBackendErrors] = useState()

  const initialRestaurantValues = { name: null, description: null, address:
null, postalCode: null, url: null, shippingCosts: null, email: null, phone:
null, restaurantCategoryId: null }
  const validationSchema = yup.object().shape({
    name: yup
      .string()
      .max(255, 'Name too long')
      .required('Name is required'),
    address: yup
      .string()
      .max(255, 'Address too long')
      .required('Address is required'),
    postalCode: yup
      .string()
      .max(255, 'Postal code too long')
```



```

        .required('Postal code is required'),
url: yup
    .string()
    .nullable()
    .url('Please enter a valid url'),
shippingCosts: yup
    .number()
    .positive('Please provide a valid shipping cost value')
    .required('Shipping costs value is required'),
email: yup
    .string()
    .nullable()
    .email('Please enter a valid email'),
phone: yup
    .string()
    .nullable()
    .max(255, 'Phone too long'),
restaurantCategoryId: yup
    .number()
    .positive()
    .integer()
    .required('Restaurant category is required')
}))

useEffect(() => {
    async function fetchRestaurantCategories () {
        try {
            const fetchedRestaurantCategories = await getRestaurantCategories()
            const fetchedRestaurantCategoriesReshaped =
fetchedRestaurantCategories.map((e) => {
                return {
                    label: e.name,
                    value: e.id
                }
            })
            setRestaurantCategories(fetchedRestaurantCategoriesReshaped)
        } catch (error) {
            showMessage({
                message: `There was an error while retrieving restaurant categories.
${error} `,
                type: 'error',
                style: GlobalStyles.flashStyle,

```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

```
        titleStyle: GlobalStyles.flashTextStyle
      })
    }
  }
  fetchRestaurantCategories()
}, [route])

useEffect(() => {
  (async () => {
    if (Platform.OS !== 'web') {
      const { status } = await
ExpoImagePicker.requestMediaLibraryPermissionsAsync()
      if (status !== 'granted') {
        alert('Sorry, we need camera roll permissions to make this work!')
      }
    }
  })()
}, [])

const pickImage = async (onSuccess) => {
  const result = await ExpoImagePicker.launchImageLibraryAsync({
    mediaTypes: ExpoImagePicker.MediaTypeOptions.Images,
    allowsEditing: true,
    aspect: [1, 1],
    quality: 1
  })
  if (!result.canceled) {
    if (onSuccess) {
      onSuccess(result)
    }
  }
}

const createRestaurant = async (values) => {
  setBackendErrors([])
  try {
    const createdRestaurant = await create(values)
    showMessage({
      message: `Restaurant ${createdRestaurant.name} succesfully created`,
      type: 'success',
      style: GlobalStyles.flashStyle,
      titleStyle: GlobalStyles.flashTextStyle
    })
  }
}
```

```

    })
    navigation.navigate('RestaurantsScreen', { dirty: true })
  } catch (error) {
    console.log(error)
    setBackendErrors(error.errors)
  }
}
return (
  <Formik
    validationSchema={validationSchema}
    initialValues={initialRestaurantValues}
    onSubmit={createRestaurant}>
    {({ handleSubmit, setFieldValue, values }) => (
      <ScrollView>
        <View style={{ alignItems: 'center' }}>
          <View style={{ width: '60%' }}>
            <InputItem
              name='name'
              label='Name:'
            />
            <InputItem
              name='description'
              label='Description:'
            />
            <InputItem
              name='address'
              label='Address:'
            />
            <InputItem
              name='postalCode'
              label='Postal code:'
            />
            <InputItem
              name='url'
              label='Url:'
            />
            <InputItem
              name='shippingCosts'
              label='Shipping costs:'
            />
            <InputItem
              name='email'

```

```

        label='Email:'
      />
      <InputItem
        name='phone'
        label='Phone:'
      />

      <DropDownPicker
        open={open}
        value={values.restaurantCategoryId}
        items={restaurantCategories}
        setOpen={setOpen}
        onSelectItem={ item => {
          setFieldValue('restaurantCategoryId', item.value)
        }}
        setItems={setRestaurantCategories}
        placeholder="Select the restaurant category"
        containerStyle={{ height: 40, marginTop: 20 }}
        style={{ backgroundColor: GlobalStyles.brandBackground }}
        dropDownStyle={{ backgroundColor: '#fafafa' }}
      />

      <ErrorMessage name={'restaurantCategoryId'} render={msg =>
<TextError>{msg}</TextError> }/>

// SOLUCIÓN-----
      <Pressable
        onPress={() => navigation.navigate('CreateRestaurantCategoryScreen')}
      />
      style=(({ pressed }) => [
        {
          backgroundColor: pressed
            ? GlobalStyles.brandGreenTap
            : GlobalStyles.brandGreen
        },
        styles.button
      ])>
      <View style={[{ flex: 1, flexDirection: 'row', justifyContent:
'center' }]]>
        <MaterialCommunityIcons name='folder-plus-outline' color={'white'}
size={20} />
        <TextRegular textStyle={styles.text}>

```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

```
Create restaurant category
</TextRegular>
</View>
</Pressable>

// -----SOLUCIÓN

<Pressable onPress={() =>
  pickImage(
    async result => {
      await setFieldValue('logo', result)
    }
  )
}
style={styles.imagePicker}
>
  <TextRegular>Logo: </TextRegular>
  <Image style={styles.image} source={values.logo ? { uri:
values.logo.assets[0].uri } : restaurantLogo} />
</Pressable>

<Pressable onPress={() =>
  pickImage(
    async result => {
      await setFieldValue('heroImage', result)
    }
  )
}
style={styles.imagePicker}
>
  <TextRegular>Hero image: </TextRegular>
  <Image style={styles.image} source={values.heroImage ? { uri:
values.heroImage.assets[0].uri } : restaurantBackground} />
</Pressable>

{backendErrors &&
  backendErrors.map((error, index) => <TextError
key={index}>{error.param}-{error.msg}</TextError>)
}

<Pressable
  onPress={handleSubmit}
  style={({ pressed }) => [
```

```

        {
          backgroundColor: pressed
            ? GlobalStyles.brandSuccessTap
            : GlobalStyles.brandSuccess
        },
        styles.button
      ]}>
    <View style=[{ flex: 1, flexDirection: 'row', justifyContent:
'center' }]]>
      <MaterialCommunityIcons name='content-save' color={'white'}
size={20}/>
      <TextRegular textStyle={styles.text}>
        Save
      </TextRegular>
    </View>
  </Pressable>
</View>
</View>
</ScrollView>
  )}
</Formik>
)
}

const styles = StyleSheet.create({
  button: {
    borderRadius: 8,
    height: 40,
    padding: 10,
    width: '100%',
    marginTop: 20,
    marginBottom: 20
  },
  text: {
    fontSize: 16,
    color: 'white',
    textAlign: 'center',
    marginLeft: 5
  },
  imagePicker: {
    height: 40,
    paddingLeft: 10,

```



```
        marginTop: 20,  
        marginBottom: 80  
    },  
    image: {  
        width: 100,  
        height: 100,  
        borderWidth: 1,  
        alignSelf: 'center',  
        marginTop: 5  
    }  
  })  
})
```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

RestaurantCategoryController

```
import { RestaurantCategory } from '../models/models.js'
const index = async function (req, res) {
  try {
    const restaurantCategories = await RestaurantCategory.findAll()
    res.json(restaurantCategories)
  } catch (err) {
    res.status(500).send(err)
  }
}

// SOLUCIÓN-----
const create = async function (req, res) {
  const newRestaurantCategory = RestaurantCategory.build(req.body)
  try {
    const restaurant = await newRestaurantCategory.save()
    res.json(restaurant)
  } catch (err) {
    console.log('error')
    res.status(500).send(err)
  }
}

// -----SOLUCIÓN
const RestaurantCategoryController = {
  index,
  create
}

export default RestaurantCategoryController
```

RestaurantCategoryRoutes

```
import RestaurantCategoryController from
'../controllers/RestaurantCategoryController.js'
import { isLoggedIn, hasRole } from '../middlewares/AuthMiddleware.js'
import * as RestaurantCategoryValidation from
'../controllers/validation/RestaurantCategoryValidation.js'
import { handleValidation } from
'../middlewares/ValidationHandlingMiddleware.js'

const loadFileRoutes = function (app) {
  app.route('/restaurantCategories')
    .get(RestaurantCategoryController.index)
  // SOLUCIÓN-----
  app.route('/restaurantCategories')
    .post(
      isLoggedIn,
      hasRole('owner'),
      RestaurantCategoryValidation.create,
      handleValidation,
      RestaurantCategoryController.create)
  // -----SOLUCIÓN

}

export default loadFileRoutes
```

RestaurantEndpoints

```
import { get, post, put, destroy } from './helpers/ApiRequestsHelper'

function getAll () {
  return get('users/myrestaurants')
}

function getDetail (id) {
  return get(`restaurants/${id}`)
}

function getRestaurantCategories () {
  return get('restaurantCategories')
}

function create (data) {
  return post('restaurants', data)
}

function update (id, data) {
  return put(`restaurants/${id}`, data)
}

function remove (id) {
  return destroy(`restaurants/${id}`)
}

// SOLUCIÓN-----
function createRestaurantCategory (data) {
  return post('restaurantCategories', data)
}
// -----SOLUCIÓN

export { getAll, getDetail, getRestaurantCategories, create, update, remove,
createRestaurantCategory }
```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

CreateRestaurantScreen

```
import React, { useEffect, useState } from 'react'
import { Image, Platform, Pressable, ScrollView, StyleSheet, View } from
'react-native'
import * as ExpoImagePicker from 'expo-image-picker'
import { MaterialCommunityIcons } from '@expo/vector-icons'
import * as yup from 'yup'
import DropDownPicker from 'react-native-dropdown-picker'
import { create, getRestaurantCategories } from
'../../api/RestaurantEndpoints'
import InputItem from '../../components/InputItem'
import TextRegular from '../../components/TextRegular'
import * as GlobalStyles from '../../styles/GlobalStyles'
import restaurantLogo from '../../assets/restaurantLogo.jpeg'
import restaurantBackground from '../../assets/restaurantBackground.jpeg'
import { showMessage } from 'react-native-flash-message'
import { ErrorMessage, Formik } from 'formik'
import TextError from '../../components/TextError'

export default function CreateRestaurantScreen ({ navigation, route }) {
  const [open, setOpen] = useState(false)
  const [restaurantCategories, setRestaurantCategories] = useState([])
  const [backendErrors, setBackendErrors] = useState()

  const initialRestaurantValues = { name: null, description: null, address:
null, postalCode: null, url: null, shippingCosts: null, email: null, phone:
null, restaurantCategoryId: null }
  const validationSchema = yup.object().shape({
    name: yup
      .string()
      .max(255, 'Name too long')
      .required('Name is required'),
    address: yup
      .string()
      .max(255, 'Address too long')
      .required('Address is required'),
    postalCode: yup
      .string()
      .max(255, 'Postal code too long')
```

```

        .required('Postal code is required'),
url: yup
    .string()
    .nullable()
    .url('Please enter a valid url'),
shippingCosts: yup
    .number()
    .positive('Please provide a valid shipping cost value')
    .required('Shipping costs value is required'),
email: yup
    .string()
    .nullable()
    .email('Please enter a valid email'),
phone: yup
    .string()
    .nullable()
    .max(255, 'Phone too long'),
restaurantCategoryId: yup
    .number()
    .positive()
    .integer()
    .required('Restaurant category is required')
}))

useEffect(() => {
    async function fetchRestaurantCategories () {
        try {
            const fetchedRestaurantCategories = await getRestaurantCategories()
            const fetchedRestaurantCategoriesReshaped =
fetchedRestaurantCategories.map((e) => {
                return {
                    label: e.name,
                    value: e.id
                }
            })
            setRestaurantCategories(fetchedRestaurantCategoriesReshaped)
        } catch (error) {
            showMessage({
                message: `There was an error while retrieving restaurant categories.
${error} `,
                type: 'error',
                style: GlobalStyles.flashStyle,

```



```

        titleStyle: GlobalStyles.flashTextStyle
      })
    }
  }
  fetchRestaurantCategories()
}, [route])

useEffect(() => {
  (async () => {
    if (Platform.OS !== 'web') {
      const { status } = await
ExpoImagePicker.requestMediaLibraryPermissionsAsync()
      if (status !== 'granted') {
        alert('Sorry, we need camera roll permissions to make this work!')
      }
    }
  })()
}, [])

const pickImage = async (onSuccess) => {
  const result = await ExpoImagePicker.launchImageLibraryAsync({
    mediaTypes: ExpoImagePicker.MediaTypeOptions.Images,
    allowsEditing: true,
    aspect: [1, 1],
    quality: 1
  })
  if (!result.canceled) {
    if (onSuccess) {
      onSuccess(result)
    }
  }
}

const createRestaurant = async (values) => {
  setBackendErrors([])
  try {
    const createdRestaurant = await create(values)
    showMessage({
      message: `Restaurant ${createdRestaurant.name} succesfully created`,
      type: 'success',
      style: GlobalStyles.flashStyle,
      titleStyle: GlobalStyles.flashTextStyle
    })
  } catch (error) {
    setBackendErrors([error.message])
  }
}

```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH



```
    })
    navigation.navigate('RestaurantsScreen', { dirty: true })
  } catch (error) {
    console.log(error)
    setBackendErrors(error.errors)
  }
}
return (
  <Formik
    validationSchema={validationSchema}
    initialValues={initialRestaurantValues}
    onSubmit={createRestaurant}>
    {({ handleSubmit, setFieldValue, values }) => (
      <ScrollView>
        <View style={{ alignItems: 'center' }}>
          <View style={{ width: '60%' }}>
            <InputItem
              name='name'
              label='Name:'
            />
            <InputItem
              name='description'
              label='Description:'
            />
            <InputItem
              name='address'
              label='Address:'
            />
            <InputItem
              name='postalCode'
              label='Postal code:'
            />
            <InputItem
              name='url'
              label='Url:'
            />
            <InputItem
              name='shippingCosts'
              label='Shipping costs:'
            />
            <InputItem
              name='email'
```

```

        label='Email:'
      />
      <InputItem
        name='phone'
        label='Phone:'
      />

      <DropDownPicker
        open={open}
        value={values.restaurantCategoryId}
        items={restaurantCategories}
        setOpen={setOpen}
        onSelectItem={ item => {
          setFieldValue('restaurantCategoryId', item.value)
        }}
        setItems={setRestaurantCategories}
        placeholder="Select the restaurant category"
        containerStyle={{ height: 40, marginTop: 20 }}
        style={{ backgroundColor: GlobalStyles.brandBackground }}
        dropDownStyle={{ backgroundColor: '#fafafa' }}
      />

      <ErrorMessage name={'restaurantCategoryId'} render={msg =>
<TextError>{msg}</TextError> }/>

// SOLUCIÓN-----
      <Pressable
        onPress={() => navigation.navigate('CreateRestaurantCategoryScreen')}
      />
      style=(({ pressed }) => [
        {
          backgroundColor: pressed
            ? GlobalStyles.brandGreenTap
            : GlobalStyles.brandGreen
        },
        styles.button
      ])
      <View style={[{ flex: 1, flexDirection: 'row', justifyContent:
'center' }]}>
        <MaterialCommunityIcons name='folder-plus-outline' color={'white'}
size={20} />
        <TextRegular textStyle={styles.text}>

```

```

        Create restaurant category
      </TextRegular>
    </View>
  </Pressable>
// -----SOLUCIÓN

    <Pressable onPress={() =>
      pickImage(
        async result => {
          await setFieldValue('logo', result)
        }
      )
    }
    style={styles.imagePicker}
  >
    <TextRegular>Logo: </TextRegular>
    <Image style={styles.image} source={values.logo ? { uri:
values.logo.assets[0].uri } : restaurantLogo} />
  </Pressable>

  <Pressable onPress={() =>
    pickImage(
      async result => {
        await setFieldValue('heroImage', result)
      }
    )
  }
  style={styles.imagePicker}
  >
    <TextRegular>Hero image: </TextRegular>
    <Image style={styles.image} source={values.heroImage ? { uri:
values.heroImage.assets[0].uri } : restaurantBackground} />
  </Pressable>

  {backendErrors &&
    backendErrors.map((error, index) => <TextError
key={index}>{error.param}-{error.msg}</TextError>)
  }

  <Pressable
    onPress={handleSubmit}
    style={({ pressed }) => [

```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

```
{
  backgroundColor: pressed
    ? GlobalStyles.brandSuccessTap
    : GlobalStyles.brandSuccess
},
styles.button
]]>
<View style={[{ flex: 1, flexDirection: 'row', justifyContent:
'center' }]]>
  <MaterialCommunityIcons name='content-save' color={'white'}
size={20}/>
  <TextRegular textStyle={styles.text}>
    Save
  </TextRegular>
</View>
</Pressable>
</View>
</View>
</ScrollView>
)}
</Formik>
)
}
```

```
const styles = StyleSheet.create({
  button: {
    borderRadius: 8,
    height: 40,
    padding: 10,
    width: '100%',
    marginTop: 20,
    marginBottom: 20
  },
  text: {
    fontSize: 16,
    color: 'white',
    textAlign: 'center',
    marginLeft: 5
  },
  imagePicker: {
    height: 40,
    paddingLeft: 10,
  },
});
```

```
        marginTop: 20,  
        marginBottom: 80  
    },  
    image: {  
        width: 100,  
        height: 100,  
        borderWidth: 1,  
        alignSelf: 'center',  
        marginTop: 5  
    }  
  })  
})
```


CreateRestaurantCategoryScreen

// SOLUCIÓN-----

```
/* eslint-disable react/react-in-jsx-scope */
import { useState } from 'react'
import { createRestaurantCategory } from '../../api/RestaurantEndpoints'
import { Pressable, StyleSheet, View } from 'react-native'
import { MaterialCommunityIcons } from '@expo/vector-icons'
import * as yup from 'yup'
import InputItem from '../../components/InputItem'
import TextRegular from '../../components/TextRegular'
import * as GlobalStyles from '../../styles/GlobalStyles'
import { showMessage } from 'react-native-flash-message'
import { Formik } from 'formik'
import TextError from '../../components/TextError'

export default function CreateRestaurantCategoryScreen ({ navigation }) {
  const [backendErrors, setBackendErrors] = useState()
  const initialRestaurantValues = { name: null, createdAt: null, updatedAt:
null }

  const validationSchema = yup.object().shape({
    name: yup
      .string()
      .max(50, 'Name too long')
      .required('Name is required')
  })

  const createCategory = async (values) => {
    setBackendErrors([])
    try {
      const createdCategory = await createRestaurantCategory(values)
      showMessage({
        message: `Restaurant category ${createdCategory.name} succesfully
created`,
        type: 'success',
        style: GlobalStyles.flashStyle,
        titleStyle: GlobalStyles.flashTextStyle
      })
      navigation.navigate('CreateRestaurantScreen', { dirty: true })
    }
  }
}
```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

```
    } catch (err) {
      console.log(err)
      setBackendErrors(err.errors)
    }
  }

  return <Formik
    validationSchema={validationSchema}
    initialValues={initialRestaurantValues}
    onSubmit={createCategory}
  >
    ({ handleSubmit, setFieldValue, values }) => (
      <>
        <View style={{ alignItems: 'center' }}>
          <View style={{ width: '60%' }}>
            <InputItem
              name='name'
              label='Name:'
            />
            <Pressable
              onPress={handleSubmit}
              style={({ pressed }) => [
                {
                  backgroundColor: pressed
                    ? GlobalStyles.brandSuccessTap
                    : GlobalStyles.brandSuccess
                },
                styles.button
              ]
            >
              <View style={{ flex: 1, flexDirection: 'row', justifyContent:
                'center' }}}>
                <MaterialCommunityIcons name='content-save' color={'white'}
                  size={20}/>
                <TextRegular textStyle={styles.text}>
                  Save
                </TextRegular>
              </View>
            </Pressable>
            {backendErrors &&
              backendErrors.map((error, index) => <TextError
                key={index}>{error.msg}</TextError>)
            }
          </View>
        </View>
      </>
    )
  )
```

```
        </View>
    </View>

    </>
  })

  </Formik>
}

const styles = StyleSheet.create({
  button: {
    borderRadius: 8,
    height: 40,
    padding: 10,
    width: '100%',
    marginTop: 20,
    marginBottom: 20
  }
})

// -----SOLUCIÓN
```

RestaurantStack

```
import { createNativeStackNavigator } from '@react-navigation/native-stack'
import React from 'react'
import CreateProductScreen from './CreateProductScreen'
import CreateRestaurantScreen from './CreateRestaurantScreen'
import EditProductScreen from './EditProductScreen'
import EditRestaurantScreen from './EditRestaurantScreen'
import RestaurantDetailScreen from './RestaurantDetailScreen'
import RestaurantsScreen from './RestaurantsScreen'
import CreateRestaurantCategoryScreen from './CreateRestaurantCategoryScreen'
```

```
const Stack = createNativeStackNavigator()
```

```
export default function RestaurantsStack () {
  return (
```

```
    <Stack.Navigator>
```

```
      <Stack.Screen
```

```
        name='RestaurantsScreen'
```

```
        component={RestaurantsScreen}
```

```
        options={{
```

```
          title: 'My Restaurants'
```

```
        }} />
```

```
      <Stack.Screen
```

```
        name='RestaurantDetailScreen'
```

```
        component={RestaurantDetailScreen}
```

```
        options={{
```

```
          title: 'Restaurant Detail'
```

```
        }} />
```

```
      <Stack.Screen
```

```
        name='CreateRestaurantScreen'
```

```
        component={CreateRestaurantScreen}
```

```
        options={{
```

```
          title: 'Create Restaurant'
```

```
        }} />
```

```
// SOLUCIÓN-----
```

```
      <Stack.Screen
```

```
        name='CreateRestaurantCategoryScreen'
```

```
        component={CreateRestaurantCategoryScreen}
```

```
        options={{
```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH



```
        title: 'Create Restaurant Category'
      }} />
// -----SOLUCIÓN-----
<Stack.Screen
  name='CreateProductScreen'
  component={CreateProductScreen}
  options={{
    title: 'Create Product'
  }} />
<Stack.Screen
  name='EditRestaurantScreen'
  component={EditRestaurantScreen}
  options={{
    title: 'Edit Restaurant'
  }} />
<Stack.Screen
  name='EditProductScreen'
  component={EditProductScreen}
  options={{
    title: 'Edit Product'
  }} />
</Stack.Navigator>
)
}
```