

EXAMEN MENSAJES PARA LOS FANS

- JUNIO 2023

Se desea que los dueños de restaurantes puedan enviar mensajes a sus fans, a modo de anuncios. Para ello, deberás añadir el campo correspondiente a los formularios de creación y edición de restaurantes, tal y como aparece en una de las fotografías adjuntas. El mensaje para los fans aparecerá dentro de la ventana “Restaurant Detail”, encima del botón de crear producto. Su estilo, en negrita, variará de manera intermitente entre los colores negro y azul, esto es, dichos colores se alternarán por segundo. El mensaje para los fans tendrá una longitud máxima de 500 caracteres y, obviamente, no se mostrará ni ocupará espacio en pantalla si un dueño de restaurante no ha puesto un mensaje para los fans.



Segundo 1



Segundo 2

← Edit Restaurant

Name:

100 montaditos

Introduce a message for the fans:

Miércoles y Domingos, todo al 50%

Description:

Una forma divertida y variada de disfrutar de la comida. Un lugar para compartir experiencias y dejarse llevar por el momento.

Address:

Av. de la Reina Mercedes, 43, Sevilla

Postal code:

41012

Url:

http://span.100montaditos.com/

BACKEND

Añadir la nueva propiedad:

Tendremos que añadir una propiedad de tipo String que permita dar el mensaje.

Añadimos la propiedad al modelo de Restaurant:

```
messageToFans: {  
  allowNull: true,  
  type: DataTypes.STRING  
}
```

Y al create Restaurant:

```
messageToFans: {  
  allowNull: true,  
  type: Sequelize.STRING  
},
```

Luego la validación para el texto:

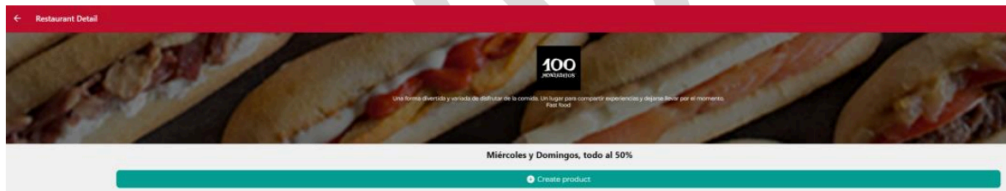
El mensaje para los fans tendrá una longitud máxima de 500 caracteres y, obviamente, no se mostrará ni ocupará espacio en pantalla si un dueño de restaurante no ha puesto un mensaje para los fans.

Añadimos la validación en create y update:

```
check('messageToFans').optional({ nullable: true, checkFalsy: true  
}).isLength({ max: 500 }).isString().trim()
```

FRONTEND

Tenemos que añadir el texto:



Segundo 1



Segundo 2

```
const AnimatedText = () => {
  const [color, setColor] = useState('blue')

  useEffect(() => {
    const interval = setInterval(() => {
      setColor((prevColor) => (prevColor === 'black' ? 'blue' :
'black'))
    }, 750) // Change color every x milliseconds

    return () => clearInterval(interval) // Clean up the interval
when the component unmounts
  }, [])

  return <TextRegular style={[styles.text, { color, fontSize: 20,
fontWeight: 'bold', marginTop: 10, marginBottom: 10 }]}>
    {restaurant.messageToFans}</TextRegular>
  }
}
```

Y lo añadimos al renderHeader:

```
        <Image style={styles.image} source={restaurant.logo ? { uri:
process.env.API_BASE_URL + '/' + restaurant.logo, cache: 'force-cache'
} : undefined} />

        <TextRegular
textStyle={styles.description}>{restaurant.description}</TextRegular>

        <TextRegular
textStyle={styles.description}>{restaurant.restaurantCategory ?
restaurant.restaurantCategory.name : ''}</TextRegular>

        </View>
    </ImageBackground>

    <AnimatedText>
    </AnimatedText>

    <Pressable
        onPress={() => navigation.navigate('CreateProductScreen', {
id: restaurant.id })
        }
        style={({ pressed }) => [
            {
                backgroundColor: pressed
                ? GlobalSt
```

Y en el edit restaurant screen añadimos:

```
const [initialRestaurantValues, setInitialRestaurantValues] =
useState({ name: null, description: null, address: null, postalCode:
null, url: null, shippingCosts: null, email: null, phone: null,
restaurantCategoryId: null, logo: null, heroImage: null, messageToFans:
null })
```

Y al validation esquema:

```
messageToFans: yup
    .string()
    .nullable()
    .max(500, 'Message to fans too long')
```

Y añadimos el input Item:

```
<InputItem
    name='description'
    label='Description:'
/>
```

Name:

Casa Félix

Introduce a message to your fans:

Descuentos!

Description:

Cocina Tradicional

Address:

Y saldrá:

Descuentos!

+ Create product

Ensaladilla

Tus salados favoritos