

EXAMEN PRODUCTOS DESTACADOS

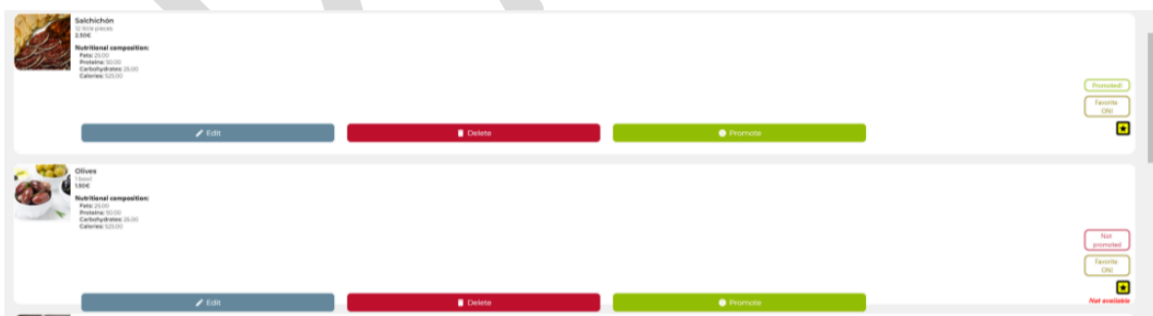
- OCTUBRE 2023

Se desea permitir a los dueños de restaurantes destacar los productos de sus restaurantes. Un dueño podrá destacar un producto dentro de un restaurante de dos maneras distintas:

- En el formulario de creación y/o edición de producto. Por defecto, se seleccionará la opción de “no destacado”. Si el propietario indica que el producto debe estar destacado, pero ya existían cinco productos destacados del mismo restaurante, al pulsar el botón Save se mostrará un error y no se creará o editará el producto. El mecanismo de destacar o no un producto, en este caso, será un deslizable. Nótese: deberá introducir un mensaje avisando del error.
- En la pantalla de "Detalles del restaurante", mediante un botón mostrado junto a cada producto, que permitirá mediante su pulsación destacar el producto en cuestión. Si el propietario pulsa el botón para destacar un nuevo producto y ya existían cinco productos destacados del mismo restaurante, se procederá a destacar el producto indicado y se marcará como "no destacado" el producto más antiguo que fue destacado con anterioridad. La aplicación debe pedir confirmación al propietario cuando se pulse el botón; utilice para ello el componente suministrado ConfirmationModal, similar al componente DeleteModal utilizado en la asignatura.

Además, los productos destacados aparecerán siempre al principio de los listados de productos que se le presenten tanto a los dueños como a los clientes dentro de un cierto restaurante. Nótese que el orden de aparición de los productos destacados irá en función de la fecha de destaque de más reciente a más antigua.

Además de presentarse al principio, los productos promocionados deben destacarse visualmente, por lo que aparecerá una insignia: la insignia será una estrella de color negro sin contorno cuando un producto no esté destacado o una estrella encuadrada de color negro con contorno amarillo cuando un producto sea destacado. A su vez, encima de la insignia, aparecerá un texto que cambiará entre “Not favorite” y “Favorite ON!” en función de si el producto esté o no destacado; el color del texto cuando el producto esté destacado será “dorado” y “negro” cuando no lo esté.



BACKEND

Añadir la nueva propiedad:

Tendremos que crear una nueva propiedad booleana que tome valor true si está destacado ese producto o false si no lo esta.

Añadimos la propiedad al modelo de Product:

```
isHighlight: DataTypes.BOOLEAN
```

Y al create Product:

```
isHighlight: {  
  type: Sequelize.BOOLEAN  
},
```

Dicen que:

- *En el formulario de creación y/o edición de producto. Por defecto, se seleccionará la opción de “no destacado”. Si el propietario indica que el producto debe estar destacado, pero ya existían cinco productos destacados del mismo restaurante, al pulsar el botón Save se mostrará un error y no se creará o editará el producto. El mecanismo de destacar o no un producto, en este caso, será un deslizable. Nótese: deberá introducir un mensaje avisando del error.*

Luego tenemos que crear una validación de que no existan mas de 5 productos promocionados:

```
const checkOnly5HighlightedProducts = async (value, { req }) => {  
  try {  
    const products = Product.count({ where: { restaurantId:  
req.body.restaurantId, isHighlight: true } })  
    if (products >= 5) {  
      return Promise.reject(new Error('Maximum number of highlighted  
products is 5.'))  
    } else {  
      return Promise.resolve('OK')  
    }  
  } catch (err) {  
    return Promise.reject(new Error(err))  
  }  
}
```

Y la añadimos al create y update:

```
check('isHighlight').custom(checkOnly5HighlightedProducts).withMessage(  
'Maximum number of highlighted products is 5.')
```

Luego vamos a crear la función que nos permite destacar o no los productos en el controlador de productos:

```
const highlight = async function (req, res) {
  const t = await sequelizeSession.transaction()
  try {
    const totalHighlited = await Product.count({ where: { isHighlight:
true } })

    if (totalHighlited >= 5) {
      const firstHighlighted = await Product.findOne({ where: {
isHighlight: true }, order: [['createdAt'], ['DESC']] })
      await Product.update(
        { isHighlight: false },
        { where: { id: firstHighlighted.id }, transaction: t }
      )
    }

    await Product.update(
      { isHighlight: true },
      { where: { id: req.params.productId }, transaction: t }
    )
    await t.commit()
    res.status(200).send('Product highlighted successfully')
  } catch (err) {
    await t.rollback()
    res.status(500).send(err)
  }
}
```

Nótese que el orden de apariencia de los productos destacados irá en función de la fecha de destaque de más reciente a más antigua.

```
const show = async function (req, res) {
  // Only returns PUBLIC information of restaurants
  try {
    const restaurant = await
Restaurant.findByPk(req.params.restaurantId, {
  attributes: { exclude: ['userId'] },
  include: [{
    model: Product,
    as: 'products',
    include: { model: ProductCategory, as: 'productCategory' }
  },
  {
    model: RestaurantCategory,
    as: 'restaurantCategory'
  }
  ],
  order: [[{ model: Product, as: 'products' }, 'isHighlight',
'DESC'], [{ model: Product, as: 'products' }, 'createdAt', 'DESC']]
  )
  res.json(restaurant)
} catch (err) {
  res.status(500).send(err)
}
}
```

Y por ultimo la ruta:

```
app.route('/products/:productId/highlight')
  .patch(
    isLoggedIn,
    hasRole('owner'),
    checkEntityExists(Product, 'productId'),
    ProductMiddleware.checkProductOwnership,
    ProductController.highlight
  )
```

FRONTEND

Tenemos que añadir el endpoint con la ruta que hemos creado:

```
function highlight (id) {  
  return patch(`products/${id}/highlight`)  
}
```

Como vamos a crear un modal, creamos un useState:

```
const [productToBeHighlighted, setProductToBeHighlighted] =  
useState(null)
```

Vamos a crear una vista que contenga el badge y la estrella:

```
<View style={[{ flexDirection: 'column', justifyContent:  
'space-between', alignItems: 'flex-end', marginBottom: 100 }]}>  
  {item.isHighlight &&  
    <TextRegular textStyle={[styles.badge, { color:  
GlobalStyles.brandGreen, borderColor: GlobalStyles.brandSuccess }]}>  
      ¡Favourite ON!  
    </TextRegular>  
  }  
  {!item.isHighlight &&  
    <TextRegular textStyle={[styles.badge, { color:  
GlobalStyles.brandPrimary, borderColor: GlobalStyles.brandPrimary }]}>  
      ¡Favourite OFF!  
    </TextRegular>  
  }  
  <Pressable  
    onPress={() => { setProductToBeHighlighted(item) }}  
    style={() => [  
      {  
        backgroundColor: item.isHighlight ?  
GlobalStyles.yellow : GlobalStyles.white,  
        alignItems: 'center',  
        justifyContent: 'center',  
        borderRadius: 4,  
        borderWidth: item.isHighlight ? 4 : 0, // tamaño de  
relleno. Se rellena sólo si el producto está destacado.  
        height: 30,  
        width: 30  
      }  
    ]}>  
    <MaterialCommunityIcons name='star' color={'black'}  
size={20}  
    />  
  </Pressable>  
</View>
```

Y luego el botón que destaca el producto, creando un Modal:

```
<Pressable
  onPress={() => { setProductToBeHighlighted(item) }}
  style={({ pressed }) => [
    {
      backgroundColor: pressed
        ? GlobalStyles.brandGreenTap
        : GlobalStyles.brandGreen
    },
    styles.actionButton
  ]}>
  <View style={[{ flex: 1, flexDirection: 'row',
justifyContent: 'center' }]}>
    <MaterialCommunityIcons name='star' color={'white'}
size={20}/>
    <TextRegular textStyle={styles.text}>
      Highlight product
    </TextRegular>
  </View>
</Pressable>
```

Y por último la función que promociona el producto:

```
const highlightProduct = async (product) => {
  try {
    await highlight(product.id)
    await fetchRestaurantDetail()
    setProductToBeHighlighted(null)
    showMessage({
      message: `Product ${product.name} succesfully highlighted`,
      type: 'success',
      style: GlobalStyles.flashStyle,
      titleStyle: GlobalStyles.flashTextStyle
    })
  } catch (error) {
    console.log(error)
    setProductToBeHighlighted(null)
    showMessage({
      message: `Product ${product.name} could not be highlighted.`,
      type: 'error',
      style: GlobalStyles.flashStyle,
      titleStyle: GlobalStyles.flashTextStyle
    })
  }
}
```

```
}
```

Con su modal:

```
<HighlightModal  
  isVisible={productToBeHighlighted !== null}  
  onCancel={() => setProductToBeHighlighted(null)}  
  onConfirm={() => highlightProduct(productToBeHighlighted)}>  
</HighlightModal>
```