

Explicación del archivo emociones.js

1. ¿Qué hace este archivo?

El archivo emociones.js permite que cuando un usuario seleccione una emoción en el formulario del dashboard, se le asigne automáticamente un color de fondo que refleja esa emoción. Además, al enviar el formulario, los datos se mandan al backend usando AJAX (fetch) sin recargar la página.

2. Paso a paso del código

1. Creamos un objeto llamado emocionesColores que actúa como un "diccionario" con claves (emociones) y valores (colores HEX). Ejemplo:

```
const emocionesColores = {  
  feliz: "#FFE066",  
  triste: "#A3C4F3",  
  ...  
};
```

2. Esperamos a que el DOM esté completamente cargado:

```
document.addEventListener("DOMContentLoaded", function () {  
  // Todo el código irá aquí dentro  
});
```

3. Capturamos el <select> con id="select-emocion". Si no existe, salimos con return.

4. Escuchamos cuando el usuario cambia su emoción:

```
selectEmocion.addEventListener("change", function () {  
    const emocionSeleccionada = selectEmocion.value;  
    const color = emocionesColores[emocionSeleccionada];  
    if (color) {  
        selectEmocion.style.backgroundColor = color;  
        selectEmocion.style.color = "#000";  
    } else {  
        selectEmocion.style.backgroundColor = "#fff";  
    }  
});
```

5. Capturamos el formulario con id="form-estado-animo" y evitamos su envío tradicional con event.preventDefault().

6. Obtenemos los valores del select y del textarea:

```
const emocion = document.getElementById("select-emocion").value;  
const texto = document.getElementById("texto-diario").value;
```

7. Empaquetamos estos datos en un objeto:

```
const datos = {  
    emocion: emocion,  
    texto: texto  
};
```

8. Enviamos este objeto con fetch() al backend (moodController.php):

```
fetch("../controladores/moodController.php", {  
    method: "POST",
```

```
headers: {  
    "Content-Type": "application/json"  
},  
body: JSON.stringify(datos)  
})  
  
.then(response => response.json())  
  
.then(data => console.log(data))  
  
.catch(error => console.error("Error al guardar", error));
```

3. ¿Por qué lo hacemos así?

- Separar lógica JS del HTML es buena práctica.
- Usar JSON permite estandarizar el envío de datos.
- Fetch es una forma moderna de hacer peticiones HTTP sin recargar la página.
- Asociar colores a emociones da una experiencia visual intuitiva y emocional.

4. Próximos pasos

Ahora toca recibir los datos en moodController.php, validarlos, y pasarlos al modelo EstadoAnimo.php para guardarlos en la base de datos.

¡Vamos paso a paso como en una empresa profesional!