

Proyecto Entorns Sana y Marc



Funcionalidad

Este proyecto es una web dedicada al entretenimiento, centrada en los gatos, tiene un diseño simple amigable y está enfocado especialmente para niños.

La aplicación carga una **imagen aleatoria de un gato** y un **dato curioso** usando dos APIs diferentes. Además, incluye una pequeña funcionalidad que reproduce un sonido al hacer clic derecho y un botón que muestra instrucciones.

Usa **C# y .NET Core MVC**, organizando el proyecto en Modelo, Vista y Controlador, y usando Razor para generar la interfaz con los datos recibidos.

¿Cómo funciona?

• Imágenes

Desde el controlador (HomeController.cs), se hace una petición a la api <https://api.thecatapi.com/v1/images/search>, que devuelve una lista con una imagen aleatoria.

Se accede al campo url de la primera imagen, y se guarda en el modelo en la clase Gato. En la vista Razor y luego se muestra con `@Model.Url` reemplazando el src de la imagen con él.

• Textos

Otra llamada, pero esta vez a <https://catfact.ninja/fact>, que devuelve directamente una frase curiosa esta la guarda en la misma clase y modelo, luego se muestra con `@Model.Fact`.

• Sonido

Es un archivo de audio local que se reproduce al hacer clic derecho usando JS.

Explicación detallada del código

En el HomeController.cs, que controla la lógica principal, se hacen las llamadas a las APIs y se envían los datos a la vista.

```
0 references
public IActionResult Index() // Función que carga la página
{
    const string apiImage = "https://api.thecatapi.com/v1/images/search"; // API de imagen
    const string apiFact = "https://catfact.ninja/fact"; // API de frase

    var client = new HttpClient(); // Cliente para hacer las peticiones

    var responseImagen = client.GetAsync(apiImage).Result; // Pido la imagen
    var contentImagen = responseImagen.Content.ReadAsStringAsync().Result; // Guardo la respuesta como texto
    var imagenconlista = JsonConvert.DeserializeObject<List<Gato>>(contentImagen); // Convierto el texto en lista
    var imagenFinal = imagenconlista[0]; // Cojo la primera imagen

    var responseFact = client.GetAsync(apiFact).Result; // Pido la frase
    var contentFact = responseFact.Content.ReadAsStringAsync().Result; // Guardo la respuesta como texto
    var factFinal = JsonConvert.DeserializeObject<Gato>(contentFact); // Convierto el texto en objeto

    var model = new Gato // Junto la imagen y la frase en un solo modelo
    {
        Url = imagenFinal.Url,
        Fact = factFinal.Fact,
    };

    return View(model); // Mando el modelo a la vista
}
```

Gato.cs, controla el modelo de datos, es donde se guardan la imagen del gato y el texto.

```
Models / Gato.cs / Gato
6 references
1 public class Gato // Clase con los datos del gato
2 {
3     public string Url { get; set; } // Dirección de la imagen
4     public string Fact { get; set; } // Frase del gato
5 }
```

Index.cshtml es donde va todo el código de la parte que se ve en la web

```
ews > Home > Index.cshtml
1  @{
2      ViewData["Title"] = "Cat Project"; // Título que se ve en la pestaña del navegador
3  }
4
5  @model Gato // Se indica que esta vista usará el modelo Gato
6
```

```
<div id="tarjeta">
   <!-- Imagen del gato que referencia con el model y luego le indico el dato que debe usar con el .url-->
  <p class="bold">Cat fact!</p>
  <p>@Model.Fact</p> <!-- Texto de gato que referencia con el model y luego le indico el dato que debe usar con el .Fact-->
</div>
```

El archivo site.js, controla las funciones interactivas como activar el sonido al hacer clic derecho y crear la nota al pulsar el botón.

```
wroot > js > JS site.js > RightClicado
1  window.onload = function () {
2      RightClicado(); // Activa el sonido al hacer clic derecho
3      let boton = document.querySelector("#instrucciones"); // Busca el botón de instrucciones
4      boton.addEventListener('click', generarInstrucciones); // Si se hace clic, genera las instrucciones
5  };
6
7  function RightClicado() {
8      let audio = new Audio('/attachments/miau.mp3'); // Carga el audio del maullido
9
10     document.addEventListener('contextmenu', function(event) {
11         event.preventDefault(); // Evita que se abra el menú del clic derecho
12         audio.play(); // Reproduce el maullido
13     });
14 }
15
16 function generarInstrucciones() {
17     var button = document.getElementById("instrucciones"); // Busca el botón
18     button.disabled = true; // Desactiva el botón después de clicarlo
19
20     let contenedorTarjeta = document.querySelector('#container'); // Busca el contenedor lateral
21
22     let tarjeta = document.createElement('div'); // Crea un nuevo div
23     tarjeta.classList.add('nota'); // Le aplica la clase con fondo de nota
24     contenedorTarjeta.append(tarjeta); // Lo mete dentro del contenedor
25
26     let hint = document.createElement('p'); // Crea un párrafo
27     hint.classList.add('textnota'); // Le aplica estilo
28     hint.id = 'hint'; // Le pone un ID
29     hint.innerHTML = `Right click to meow!`; // Le pone el texto
30     tarjeta.append(hint); // Añade el texto a la nota
31 }
```

Posibles mejoras

1. **Recargar contenido por separado:** Añadir botones para actualizar solo la imagen o solo la frase sin recargar toda la página.
2. **Modo favorito:** Poder guardar gatos que nos gusten (imagen + frase) en un historial.
3. **Responsividad:** Mejorar la adaptación a pantallas pequeñas o móviles.
4. **Código más limpio:** Comentar mejor el código y usar nombres más descriptivos en funciones y clases.
5. **Interacciones aleatorias:** Añadir maullidos aleatorios o animaciones al pasar el ratón.