

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN – LEÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN



AÑO LECTIVO: 2025
SEMESTRE: II

Componente Curricular: PROGRAMACION ORIENTADA A LA
WEB II

Grupo: 1

Profesor(a): Lic. Juan Carlos Leyton Briones

Autor:

Kevin Alejandro Sánchez Machado.

Carnet: 22-01062-0.

León, Nicaragua, 2025.

“¡A la Libertad por la Universidad!”

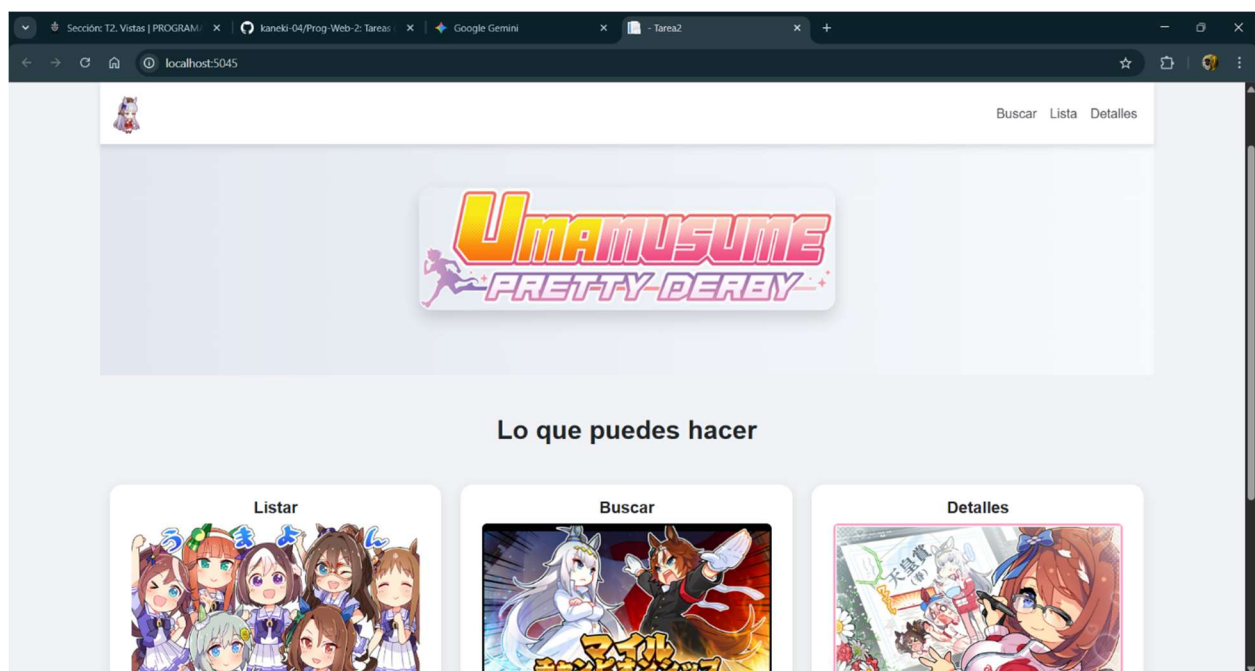
Documentación Técnica del Proyecto: Umamusume: Pretty Derby

1. Introducción

Este documento detalla el desarrollo de la aplicación web

Umamusume: Pretty Derby, creada para la Tarea 2 del curso de Programación Web 2. El proyecto, construido sobre la arquitectura

Modelo-Vista-Controlador (MVC), tiene como propósito principal gestionar y presentar información sobre los personajes del popular universo de Umamusume. Las funcionalidades clave incluyen la visualización de una lista completa de personajes, un sistema de búsqueda dinámico y una página de detalles del proyecto.



2. Tecnologías y Componentes Clave

El proyecto se sustenta en una sólida combinación de tecnologías modernas para garantizar un desarrollo eficiente y una interfaz de usuario atractiva:

- **ASP.NET Core 9 MVC:** Es el framework principal de desarrollo. Proporciona una estructura robusta que separa la aplicación en tres componentes lógicos: el Modelo (datos), la Vista (interfaz de usuario) y el Controlador (lógica de negocio).
- **Razor Syntax (.cshtml):** Utilizada para la creación de las vistas, permite incrustar código C# directamente en el HTML, facilitando el renderizado de datos dinámicos del servidor en la página web.

- **C#:** Es el lenguaje de programación del lado del servidor. Se utiliza para la lógica del Controlador y del Modelo, manipulando los datos de los personajes.
- **Bootstrap 5:** Framework de CSS utilizado para el diseño responsivo de la interfaz. Aporta un aspecto moderno y asegura que la aplicación se adapte a cualquier tamaño de pantalla. Elementos como la barra de navegación (.navbar), las tarjetas (.card) y los botones (.btn) están basados en Bootstrap.
- **CSS Personalizado:** Se ha implementado código CSS adicional para personalizar los estilos de la aplicación, como los efectos de sombra (box-shadow), transiciones suaves (transition) y transformaciones (transform) que se activan al pasar el cursor sobre los elementos.

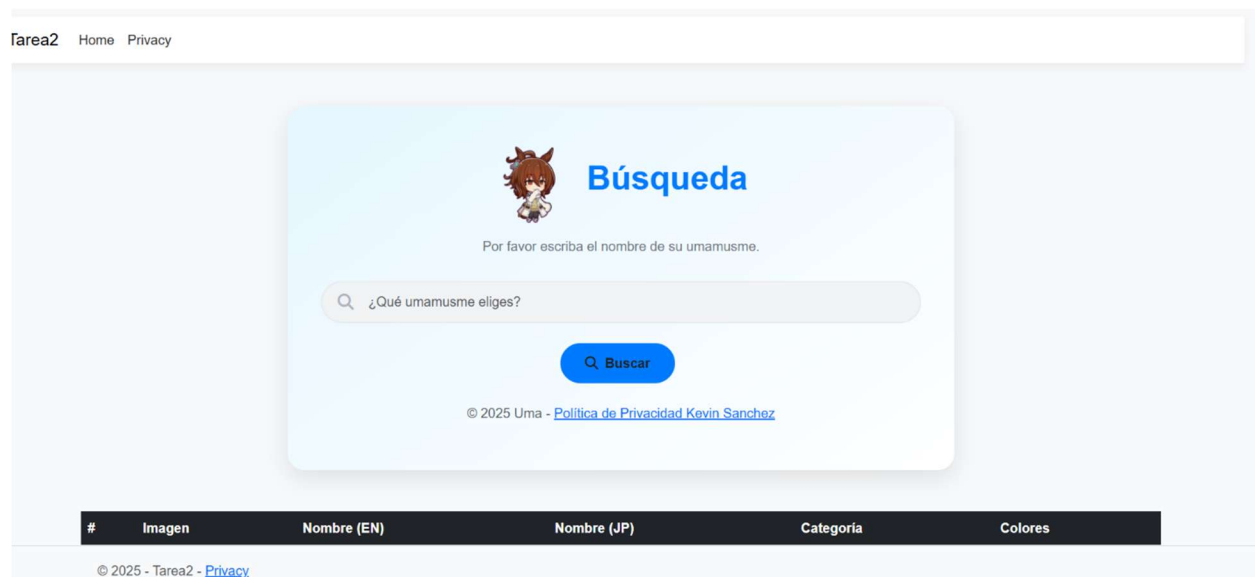
3. Estructura de Vistas (.cshtml)

Las vistas son la capa de presentación de la aplicación. A continuación, se detalla la función y el diseño de cada archivo de vista proporcionado.

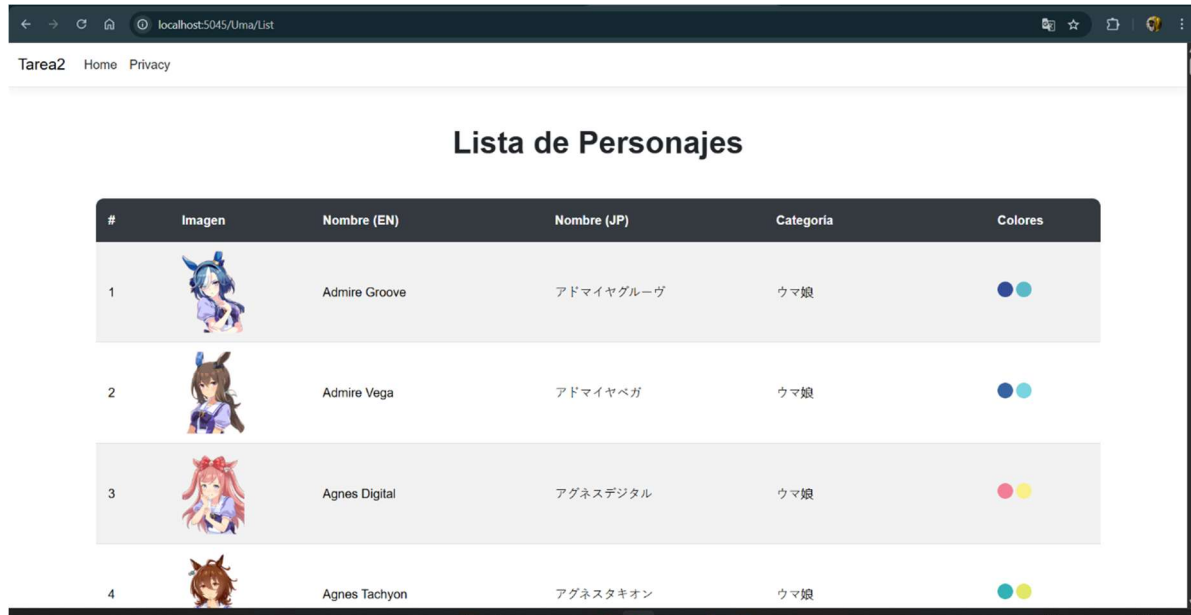
3.1. Index.cshtml - Página Principal







Este archivo sirve como la puerta de entrada a la aplicación.

- **Estructura y Contenido:** La página presenta el logo de Umamusume y tres tarjetas de servicio (.service-card) en una sección central. Cada tarjeta representa una funcionalidad clave:
 - **Listar:** Dirige a la vista List para ver todos los personajes.
 - **Buscar:** Dirige a la vista Find para buscar un personaje específico.



- **Detalles:** Dirige a la vista Details para obtener información del proyecto.
- **Barra de Navegación (.navbar):** Una barra de navegación persistente en la parte superior facilita la navegación entre las secciones Buscar, Lista y Detalles.
- **Estilo:** El diseño utiliza un degradado (linear-gradient) en la sección principal y efectos visuales como sombras en las tarjetas (`box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08)`) y transformaciones (`transform: translateY(-10px)`) al pasar el cursor sobre ellas.



#	Imagen	Nombre (EN)	Nombre (JP)	Categoría	Colores
1		Admire Groove	アドマイヤグルーヴ	ウマ娘	
2		Admire Vega	アドマイヤベガ	ウマ娘	
3		Agnes Digital	アグネスデジタル	ウマ娘	
4		Agnes Tachyon	アグネスタキオン	ウマ娘	

API

Un componente fundamental de este proyecto es el consumo de una API RESTful externa para obtener la información de los personajes de Umamusume. Este enfoque es crucial porque separa la lógica de presentación de los datos de su fuente, lo que hace que la aplicación sea más flexible y fácil de mantener.

1. ¿Qué es una API y por qué se usa?

Una API (Application Programming Interface) es un conjunto de reglas y protocolos que permite a diferentes aplicaciones comunicarse entre sí. En este caso, la API de `umapyoi.net` actúa como una base de datos pública a la que nuestra aplicación puede hacer una "solicitud" (request) para obtener los datos de los personajes.

El uso de una API es beneficioso por varias razones:

- **Separación de responsabilidades:** La API se encarga de almacenar, gestionar y actualizar los datos, mientras que tu aplicación solo se encarga de solicitarlos y mostrarlos.

- **Escalabilidad:** Si la lista de personajes crece, no necesitas modificar el código de tu aplicación, ya que la API se encargará de proporcionar la lista actualizada automáticamente.
- **Eficiencia:** En lugar de mantener una base de datos local, tu aplicación solo necesita una conexión a internet para obtener la información más reciente.

3.2. List.cshtml - Lista de Personajes

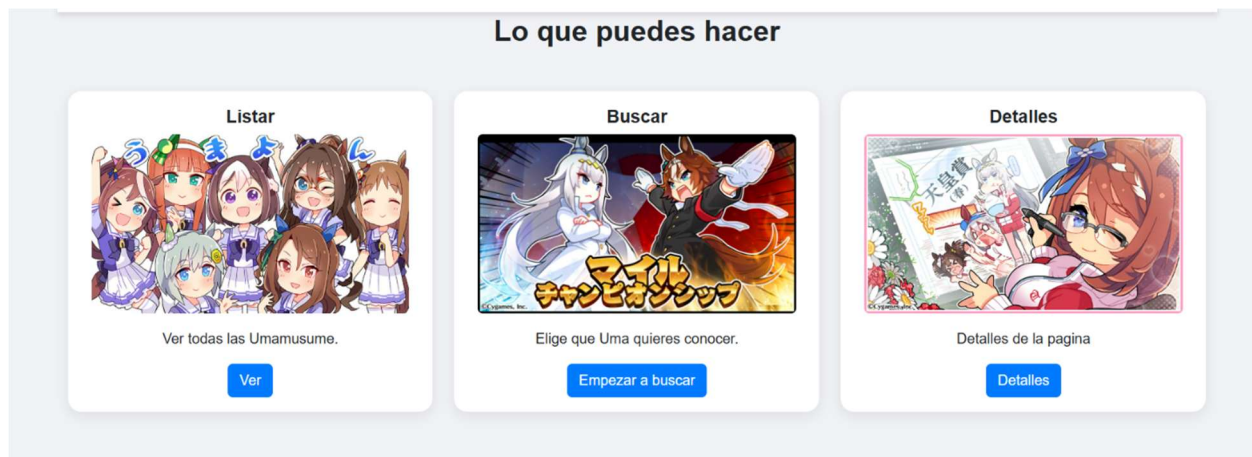
Esta vista se encarga de mostrar una tabla completa con todos los personajes del modelo.

- **Procesamiento de Datos:** El archivo utiliza la directiva `@model` `List<Tarea2.Models.CharacterInfo>` para recibir una colección de objetos `CharacterInfo` desde el controlador.
- **Renderizado Dinámico:** A través de un bucle `@foreach`, la vista itera sobre la colección del modelo para generar una fila de tabla (`<tr>`) por cada personaje. La información mostrada en la tabla incluye:
 - Número de fila (`@c.row_number`)
 - Imagen del personaje (`@c.thumb_img`)
 - Nombre en inglés (`@c.name_en`) y japonés (`@c.name_jp`)
 - Categoría (`@c.category_label`)
 - Círculos de colores (``) para los colores principales y secundarios (`@c.color_main` y `@c.color_sub`).
- **Estilo:** La tabla tiene un estilo moderno con bordes redondeados, sombras y un encabezado oscuro. Los efectos de hover resaltan las filas y las imágenes al pasar el cursor.

3.3. Find.cshtml - Búsqueda de Personajes

Esta vista implementa la funcionalidad de búsqueda a través de una combinación de HTML, CSS y JavaScript del lado del cliente.

- **Procesamiento de Datos:** Al igual que la vista `List`, recibe una lista de personajes en el modelo. La diferencia es que este modelo se serializa en un objeto JSON (`umas`) en JavaScript para su procesamiento en el navegador.
- **Funcionalidad de Búsqueda (`Search()`):** La lógica principal reside en una función de JavaScript llamada `Search()`. Esta función:
 1. Obtiene el valor del cuadro de texto de búsqueda (`SearchBox`).
 2. Filtra el array `umas` para encontrar coincidencias en el nombre en inglés (`name_en`).
 3. Limpia el contenido de la tabla de resultados.
 4. Itera sobre los resultados del filtro para construir y añadir dinámicamente nuevas filas (`<tr>`) a la tabla, mostrando la información del personaje encontrado.
- **Estilo:** La página de búsqueda tiene un diseño minimalista con un contenedor principal estilizado con un degradado y una sombra pronunciada (`box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1)`).



3.4. Details.cshtml - Detalles del Proyecto

Esta vista es una página informativa estática que describe el propósito y las tecnologías del proyecto.

- **Contenido:** El documento detalla que el proyecto es una aplicación **ASP.NET Core MVC** diseñada para la **Tarea 2**. Resalta las funcionalidades clave de búsqueda y listado y menciona el uso de **Bootstrap 5**.



- **Estilo:** El diseño utiliza un contenedor principal con un fondo claro, bordes redondeados y una sombra para un efecto de profundidad. La información se presenta en tarjetas (.info-card) con un fondo más claro para una mejor organización visual.

Conclusión

El proyecto

Umamusume: Pretty Derby es un ejemplo bien estructurado del patrón **MVC** aplicado al desarrollo web. La separación clara entre el Modelo, la Vista y el Controlador facilita la mantenibilidad del código y la implementación de nuevas funcionalidades. La combinación de **.NET 9** con **Bootstrap** demuestra un enfoque moderno y práctico para la creación de aplicaciones web interactivas y visualmente atractivas.