Registro Diario de Avances – Sprint 1 Nutrigym

**Fecha:**

**Autor:** Benjamin Miranda Quispe

Brandon Jair Chipana Salazar

John Manuel Villagarcia Mendoza

Victor Fernando Masias Baca

**Versión:** 1.0

# 1. Introducción

Este documento se evidenciara todo el proceso y desarrollo del sistema NutryGym específicamente del Sprint 1.

# 2. Lineamientos de Diseño

Desarrollar una primera versión funcional de la aplicación NutriGym que permita a los usuarios:

* Registrarse e iniciar sesión.
* Asignar roles para el ingreso a los dashboards
* Registrar preferencias alimenticias
* Gestión y validación de la información (CRUD)

# 3. Historias de Usuario y Tareas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Backlog ID | Historia de Usuario | Tiempo Estimado | Fecha de Inicio | Fecha de Finalización |
| HU-1 | Registro de usuario | 5 | 2/10/2025 | 3/10/2025 |
| HU-2 | Asignación de rol | 3 | 2/10/2025 | 3/10/2025 |
| HU-3 | Registrar preferencias alimenticias | 5 | 2/10/2025 | 3/10/2025 |

# 4. Tareas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | HU ID | Tarea | Responsable | Estimación (horas) | Estado |
| 1 | HU1 | Diseñar formulario de registro de usuario | Benjamin | 3 | [✅](https://eltiotech.com/checkbox-con-formato-condicional/) |
| 2 | HU1 | Implementar validaciones de registro | Fernando | 3 | [✅](https://eltiotech.com/checkbox-con-formato-condicional/) |
| 3 | HU1 | Crear migración y modelo de usuarios | Benjamin | 2 | [✅](https://eltiotech.com/checkbox-con-formato-condicional/) |
| 4 | HU2 | Implementar asignación de rol en backend | Benjamin | 2 | pendiente |
| 5 | HU2 | Crear tabla de roles y relaciones | Benjamin | 2 | [✅](https://eltiotech.com/checkbox-con-formato-condicional/) |
| 6 | HU3 | Diseñar interfaz para registrar preferencias | John | 3 | pendiente |
| 7 | HU3 | Implementar persistencia de preferencias | Brandon | 4 | pendiente |
| 8 | HU1-HU3 | Pruebas unitarias e integración Sprint 1 | Brandon | 3 | En curso |

# 5. Evidencias de Código

Formulario registro Usuario

@extends('layouts.app')

@section('content')

<div x-data="{

        openModal: false,

        openErrorModal: {{ session('error') ? 'true' : 'false' }},

        openSuccessModal: {{ session('success') ? 'true' : 'false' }}

    }" class="max-w-md mx-auto bg-white p-6 rounded shadow mt-10">

    <h1 class="text-2xl font-bold mb-6 text-center">Registrar Usuario</h1>

    <!-- Formulario -->

    <form x-ref="registroForm" method="POST" action="{{ route('registrar\_usuario.store') }}">

        @csrf

        <div class="mb-4">

            <label class="block mb-1 font-medium">Nombre</label>

            <input type="text" name="nombre" class="w-full border rounded px-3 py-2" value="{{ old('nombre') }}" required>

            @error('nombre') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Email</label>

            <input type="email" name="email" class="w-full border rounded px-3 py-2" value="{{ old('email') }}" required>

            @error('email') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Fecha de nacimiento</label>

            <input type="date" name="fecha\_nacimiento" class="w-full border rounded px-3 py-2" value="{{ old('fecha\_nacimiento') }}" required>

            @error('fecha\_nacimiento') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Contraseña</label>

            <input type="password" name="contrasena" class="w-full border rounded px-3 py-2" required>

            @error('contrasena') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Confirmar Contraseña</label>

            <input type="password" name="contrasena\_confirmation" class="w-full border rounded px-3 py-2" required>

        </div>

        <!-- Botón que abre el modal de confirmación -->

        <button type="button" @click="openModal = true" class="w-full bg-green-500 text-white py-2 rounded hover:bg-green-600">

            Registrarse

        </button>

    </form>

    <!-- Modal de confirmación -->

    <div x-show="openModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center">Confirmar registro</h2>

            <p class="text-gray-600 mb-6 text-center">¿Deseas continuar con el registro de este usuario?</p>

            <div class="flex space-x-3">

                <!-- Confirmar -->

                <button type="button"

                        @click="$refs.registroForm.submit(); openModal = false"

                        class="flex-1 p-3 bg-green-500 text-white rounded hover:bg-green-600">

                    Confirmar

                </button>

                <!-- Cancelar -->

                <button type="button"

                        @click="openModal = false"

                        class="flex-1 p-3 bg-gray-300 rounded hover:bg-gray-400">

                    Cancelar

                </button>

            </div>

        </div>

    </div>

    <!-- Modal de error -->

    <div x-show="openErrorModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center text-red-600">Error</h2>

            <p class="text-gray-700 mb-6 text-center">{{ session('error') }}</p>

            <div class="flex justify-center">

                <button type="button" @click="openErrorModal = false"

                    class="px-4 py-2 bg-red-500 text-white rounded hover:bg-red-600">

                    Cerrar

                </button>

            </div>

        </div>

    </div>

    <!-- Modal de éxito -->

    <div x-show="openSuccessModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center text-green-600">¡Éxito!</h2>

            <p class="text-gray-700 mb-6 text-center">{{ session('success') }}</p>

            <div class="flex justify-center">

                <button type="button" @click="openSuccessModal = false"

                    class="px-4 py-2 bg-green-500 text-white rounded hover:bg-green-600">

                    Cerrar

                </button>

            </div>

        </div>

    </div>

</div>

@endsection

Validaciones de registro

@extends('layouts.app')

@section('content')

<div x-data="{

        openModal: false,

        openErrorModal: {{ session('error') ? 'true' : 'false' }},

        openSuccessModal: {{ session('success') ? 'true' : 'false' }}

    }" class="max-w-md mx-auto bg-white p-6 rounded shadow mt-10">

    <h1 class="text-2xl font-bold mb-6 text-center">Registrar Usuario</h1>

    <!-- Formulario -->

    <form x-ref="registroForm" method="POST" action="{{ route('registrar\_usuario.store') }}">

        @csrf

        <div class="mb-4">

            <label class="block mb-1 font-medium">Nombre</label>

            <input type="text" name="nombre" class="w-full border rounded px-3 py-2" value="{{ old('nombre') }}" required>

            @error('nombre') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Email</label>

            <input type="email" name="email" class="w-full border rounded px-3 py-2" value="{{ old('email') }}" required>

            @error('email') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Fecha de nacimiento</label>

            <input type="date" name="fecha\_nacimiento" class="w-full border rounded px-3 py-2" value="{{ old('fecha\_nacimiento') }}" required>

            @error('fecha\_nacimiento') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Contraseña</label>

            <input type="password" name="contrasena" class="w-full border rounded px-3 py-2" required>

            @error('contrasena') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Confirmar Contraseña</label>

            <input type="password" name="contrasena\_confirmation" class="w-full border rounded px-3 py-2" required>

        </div>

        <!-- Botón que abre el modal de confirmación -->

        <button type="button" @click="openModal = true" class="w-full bg-green-500 text-white py-2 rounded hover:bg-green-600">

            Registrarse

        </button>

    </form>

    <!-- Modal de confirmación -->

    <div x-show="openModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center">Confirmar registro</h2>

            <p class="text-gray-600 mb-6 text-center">¿Deseas continuar con el registro de este usuario?</p>

            <div class="flex space-x-3">

                <!-- Confirmar -->

                <button type="button"

                        @click="$refs.registroForm.submit(); openModal = false"

                        class="flex-1 p-3 bg-green-500 text-white rounded hover:bg-green-600">

                    Confirmar

                </button>

                <!-- Cancelar -->

                <button type="button"

                        @click="openModal = false"

                        class="flex-1 p-3 bg-gray-300 rounded hover:bg-gray-400">

                    Cancelar

                </button>

            </div>

        </div>

    </div>

    <!-- Modal de error -->

    <div x-show="openErrorModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center text-red-600">Error</h2>

            <p class="text-gray-700 mb-6 text-center">{{ session('error') }}</p>

            <div class="flex justify-center">

                <button type="button" @click="openErrorModal = false"

                    class="px-4 py-2 bg-red-500 text-white rounded hover:bg-red-600">

                    Cerrar

                </button>

            </div>

        </div>

    </div>

    <!-- Modal de éxito -->

    <div x-show="openSuccessModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center text-green-600">¡Éxito!</h2>

            <p class="text-gray-700 mb-6 text-center">{{ session('success') }}</p>

            <div class="flex justify-center">

                <button type="button" @click="openSuccessModal = false"

                    class="px-4 py-2 bg-green-500 text-white rounded hover:bg-green-600">

                    Cerrar

                </button>

            </div>

        </div>

    </div>

</div>

@endsection

Modelos y migración tablas : usuario – preferencia – rol

@extends('layouts.app')

@section('content')

<div x-data="{

        openModal: false,

        openErrorModal: {{ session('error') ? 'true' : 'false' }},

        openSuccessModal: {{ session('success') ? 'true' : 'false' }}

    }" class="max-w-md mx-auto bg-white p-6 rounded shadow mt-10">

    <h1 class="text-2xl font-bold mb-6 text-center">Registrar Usuario</h1>

    <!-- Formulario -->

    <form x-ref="registroForm" method="POST" action="{{ route('registrar\_usuario.store') }}">

        @csrf

        <div class="mb-4">

            <label class="block mb-1 font-medium">Nombre</label>

            <input type="text" name="nombre" class="w-full border rounded px-3 py-2" value="{{ old('nombre') }}" required>

            @error('nombre') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Email</label>

            <input type="email" name="email" class="w-full border rounded px-3 py-2" value="{{ old('email') }}" required>

            @error('email') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Fecha de nacimiento</label>

            <input type="date" name="fecha\_nacimiento" class="w-full border rounded px-3 py-2" value="{{ old('fecha\_nacimiento') }}" required>

            @error('fecha\_nacimiento') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Contraseña</label>

            <input type="password" name="contrasena" class="w-full border rounded px-3 py-2" required>

            @error('contrasena') <p class="text-red-500 text-sm mt-1">{{ $message }}</p> @enderror

        </div>

        <div class="mb-4">

            <label class="block mb-1 font-medium">Confirmar Contraseña</label>

            <input type="password" name="contrasena\_confirmation" class="w-full border rounded px-3 py-2" required>

        </div>

        <!-- Botón que abre el modal de confirmación -->

        <button type="button" @click="openModal = true" class="w-full bg-green-500 text-white py-2 rounded hover:bg-green-600">

            Registrarse

        </button>

    </form>

    <!-- Modal de confirmación -->

    <div x-show="openModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center">Confirmar registro</h2>

            <p class="text-gray-600 mb-6 text-center">¿Deseas continuar con el registro de este usuario?</p>

            <div class="flex space-x-3">

                <!-- Confirmar -->

                <button type="button"

                        @click="$refs.registroForm.submit(); openModal = false"

                        class="flex-1 p-3 bg-green-500 text-white rounded hover:bg-green-600">

                    Confirmar

                </button>

                <!-- Cancelar -->

                <button type="button"

                        @click="openModal = false"

                        class="flex-1 p-3 bg-gray-300 rounded hover:bg-gray-400">

                    Cancelar

                </button>

            </div>

        </div>

    </div>

    <!-- Modal de error -->

    <div x-show="openErrorModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center text-red-600">Error</h2>

            <p class="text-gray-700 mb-6 text-center">{{ session('error') }}</p>

            <div class="flex justify-center">

                <button type="button" @click="openErrorModal = false"

                    class="px-4 py-2 bg-red-500 text-white rounded hover:bg-red-600">

                    Cerrar

                </button>

            </div>

        </div>

    </div>

    <!-- Modal de éxito -->

    <div x-show="openSuccessModal" x-transition class="fixed inset-0 flex items-center justify-center bg-black bg-opacity-50 z-50">

        <div class="bg-white max-w-md w-full rounded-xl overflow-hidden p-6">

            <h2 class="text-2xl font-bold mb-4 text-center text-green-600">¡Éxito!</h2>

            <p class="text-gray-700 mb-6 text-center">{{ session('success') }}</p>

            <div class="flex justify-center">

                <button type="button" @click="openSuccessModal = false"

                    class="px-4 py-2 bg-green-500 text-white rounded hover:bg-green-600">

                    Cerrar

                </button>

            </div>

        </div>

    </div>

</div>

@endsection

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('preferencias', function (Blueprint $table) {

            $table->id('id');

            $table->enum('tipo', ['dieta','alergia']);

            $table->string('descripcion');

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('preferencia');

    }

};

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('roles', function (Blueprint $table) {

            $table->id('id');

            $table->string('nombre\_rol', 100);

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('roles');

    }

};

Rutas actuales

1. Raiz principal login

// Redirigir la raíz al login

Route::get('/', function () {

    return redirect('/login');

});

// Registro

Route::get('/registrar\_usuario', function () {

    return view('usuario.registrar\_usuario');

})->name('registrar\_usuario');

Route::post('/registrar\_usuario', [RegistroUsuarioController::class, 'store'])

    ->name('registrar\_usuario.store');

// Dashboard

Route::get('/dashboard', function() {

    return view('dashboard'); // o tu vista correspondiente

})->name('dashboard');

Interfaz login

@extends('layouts.app')

@section('content')

<div x-data="{

        openModal: false,

        openErrorModal: {{ session('error') ? 'true' : 'false' }},

        openSuccessModal: {{ session('success') ? 'true' : 'false' }}

    }" class="min-h-screen flex items-center justify-center bg-[var(--color-surface)] py-8">

<div class="min-h-screen flex items-center justify-center bg-[var(--color-surface)]">

    <div class="w-full max-w-md p-8 neumorphic">

        <h2 class="text-2xl font-bold text-center mb-6 text-[var(--color-primary)]">

            Iniciar Sesión

        </h2>

        <!-- Formulario -->

        <form method="POST" action="{{ route('login') }}">

            @csrf

            <!-- Email -->

            <div class="mb-4">

                <label for="email" class="block text-sm font-medium mb-1">Correo</label>

                <input type="email" id="email" name="email"

                       class="input-neu"

                       value="{{ old('email') }}" required autofocus>

                @error('email')

                    <p class="text-red-500 text-xs mt-1">{{ $message }}</p>

                @enderror

            </div>

            <!-- Password -->

            <div class="mb-6">

                <label for="password" class="block text-sm font-medium mb-1">Contraseña</label>

                <input type="password" id="password" name="contrasena"

                       class="input-neu"

                       required>

                @error('password')

                    <p class="text-red-500 text-xs mt-1">{{ $message }}</p>

                @enderror

            </div>

            <!-- Remember me -->

            <div class="flex items-center mb-6">

                <input type="checkbox" id="remember" name="remember"

                       class="mr-2">

                <label for="remember" class="text-sm">Recuérdame</label>

            </div>

            <!-- Botón -->

            <div class="flex justify-center">

                <button type="submit" class="btn-neu w-full text-center">

                    Entrar

                </button>

            </div>

            <!-- Link de registro -->

            <div class="mt-4 text-center">

                <p class="text-sm">

                    ¿No tienes cuenta?

                    <a href="{{ route('registrar\_usuario') }}" class="text-[var(--color-accent)] hover:underline">

                        Regístrate aquí

                    </a>

                </p>

            </div>

        </form>

    </div>

</div>

@endsection

Validación de registro

1. Controller

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Usuario;

use Illuminate\Http\RedirectResponse;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Redirect;

use Illuminate\Validation\Rules;

class LoginUsuarioController extends Controller

{

    // Manejo de login para los usuarios

    public function validacion(Request $request) : RedirectResponse {

        // credenciales de ingreso

        $credenciales = $request->validate([

            'email'=>['required','email'],

            'contrasena'=>['required'],

        ]);

        // Busqueda del email

        $usuario  = Usuario::where('email',$credenciales['email'])->first();

        //Verificar contrasena y email

        if ($usuario && Hash::check($credenciales['contrasena'], $usuario->contrasena))

        {

            Auth::login($usuario);

            // Redirigir y enviar mensaje de ingreso

            return redirect()->intended('/dashboard')->with('succes','Bienvenido'.$usuario->nombre);

        }

        // Si falla

        return back()->withErrors([

            'email' => "Las credenciales son errores",

        ])->onlyInput('email');

    }

    public function cerrar(Request $request): RedirectResponse

    {

        Auth::logout();

        $request->session()->invalidate();

        $request->session()->regenerateToken();

        return redirect('/login')->with('success', 'Sesión cerrada correctamente.');

    }

}

1. Model

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable; //  importante

use Illuminate\Notifications\Notifiable;

class Usuario extends Authenticatable

{

    use HasFactory, Notifiable;

    protected $table = 'usuarios';

    protected $fillable = [

        'nombre',

        'email',

        'contrasena',

        'id\_rol',

        'fecha\_nacimiento',

        'fecha\_registro',

    ];

    protected $hidden = [

        'contrasena'

    ];

    // Relación: usuario pertenece a un rol

    public function rol()

    {

        return $this->belongsTo(Rol::class, 'id\_rol');

    }

    // Relación: usuario tiene muchas preferencias asignadas

    public function preferencias()

    {

        return $this->belongsToMany(

            Preferencia::class,

            'asignacion\_preferencia',

            'id\_usuario',

            'id\_preferencia'

        )->withTimestamps();

    }

}

Asignación de rol

1. Mgration

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('asignacion\_preferencia', function (Blueprint $table) {

            $table->id('id\_asignacion');

            $table->foreignId('id\_usuario')

                ->constrained('usuarios')

                ->onDelete('cascade');

            $table->foreignId('id\_preferencia')

                ->constrained('preferencias')

                ->onDelete('cascade');

            $table->date('fecha')->useCurrent();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('asignacion\_preferencia');

    }

};

1. Sedd

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class RolesSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*/

    public function run(): void

    {

        //

            DB::table('roles')->insert([

            ['nombre\_rol' => 'Administrador', 'created\_at' => now(), 'updated\_at' => now()],

            ['nombre\_rol' => 'Nutriólogo', 'created\_at' => now(), 'updated\_at' => now()],

            ['nombre\_rol' => 'Entrenador', 'created\_at' => now(), 'updated\_at' => now()],

            ['nombre\_rol' => 'Usuario', 'created\_at' => now(), 'updated\_at' => now()],

        ]);

    }

}

Asignación preferencia

1. Migration

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('asignacion\_preferencia', function (Blueprint $table) {

            $table->id('id\_asignacion');

            $table->foreignId('id\_usuario')

                ->constrained('usuarios')

                ->onDelete('cascade');

            $table->foreignId('id\_preferencia')

                ->constrained('preferencias')

                ->onDelete('cascade');

            $table->date('fecha')->useCurrent();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('asignacion\_preferencia');

    }

};

1. Seed

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class PreferenciasSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*/

    public function run(): void

    {

        //

            DB::table('preferencias')->insert([

            ['tipo' => 'dieta', 'descripcion' => 'Vegetariano', 'created\_at' => now(), 'updated\_at' => now()],

            ['tipo' => 'dieta', 'descripcion' => 'Vegano', 'created\_at' => now(), 'updated\_at' => now()],

            ['tipo' => 'dieta', 'descripcion' => 'Sin Gluten', 'created\_at' => now(), 'updated\_at' => now()],

            ['tipo' => 'dieta', 'descripcion' => 'Bajo en Azúcar', 'created\_at' => now(), 'updated\_at' => now()],

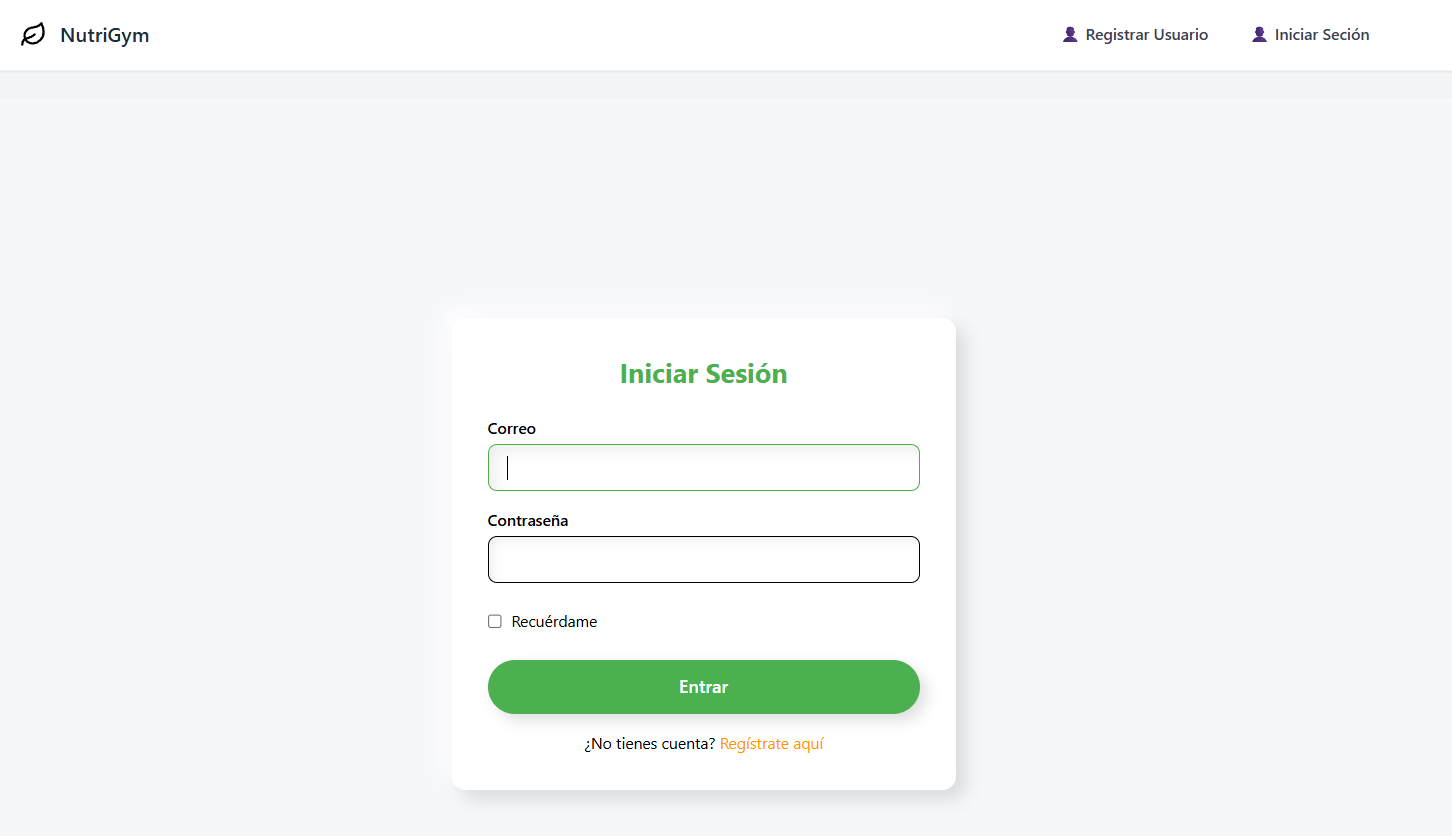
            ['tipo' => 'alergia', 'descripcion' => 'Sin Lactosa', 'created\_at' => now(), 'updated\_at' => now()],

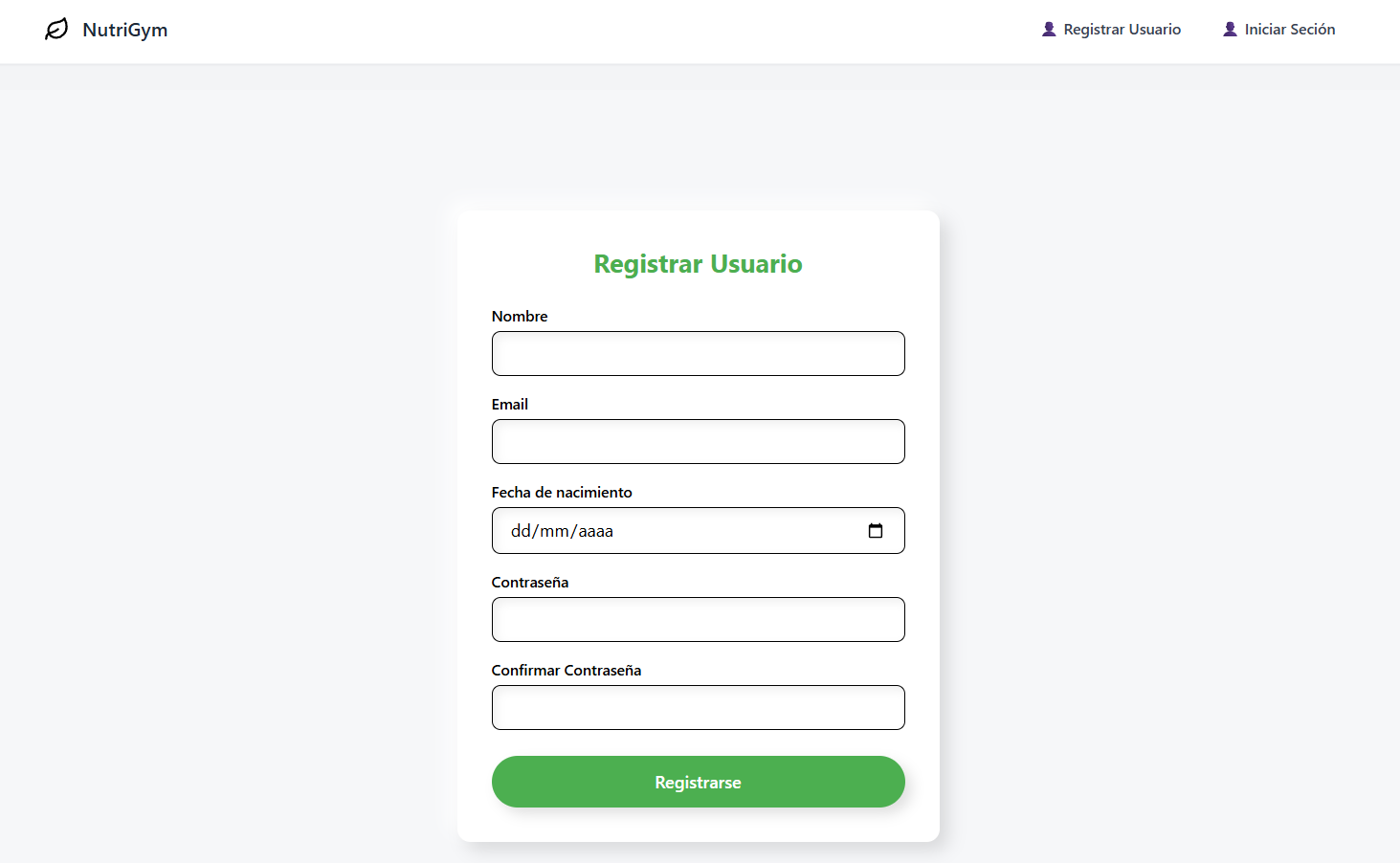
        ]);

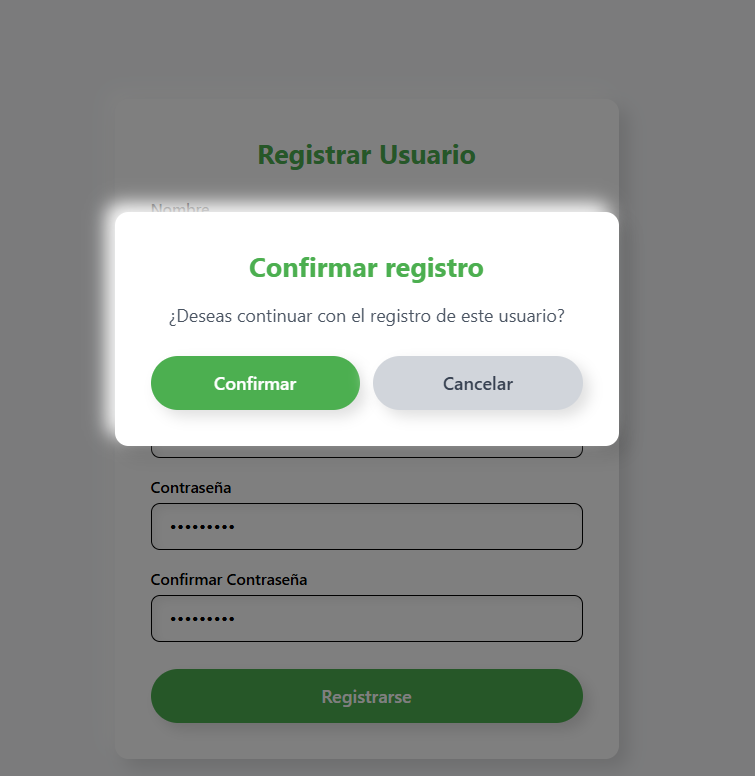
    }

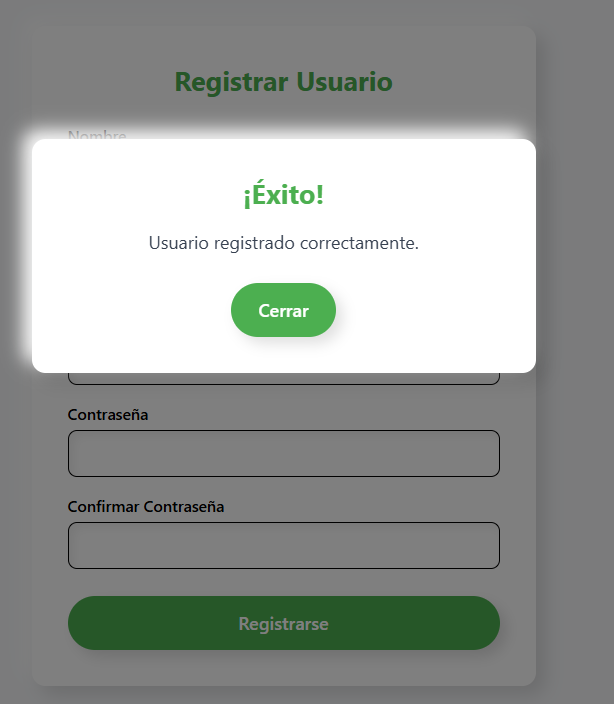
}

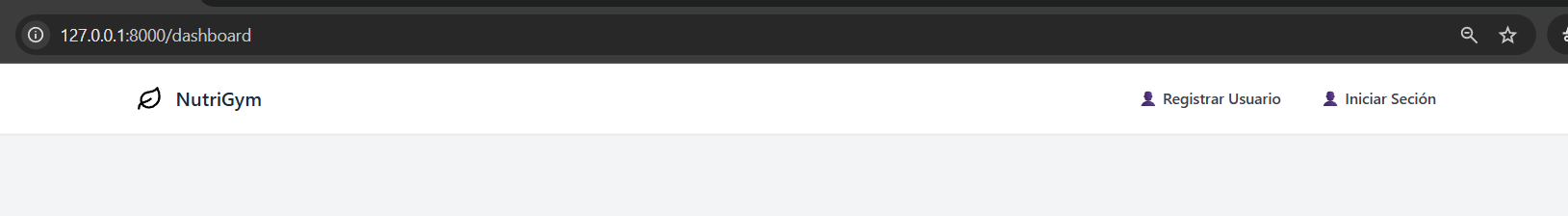
Resultados

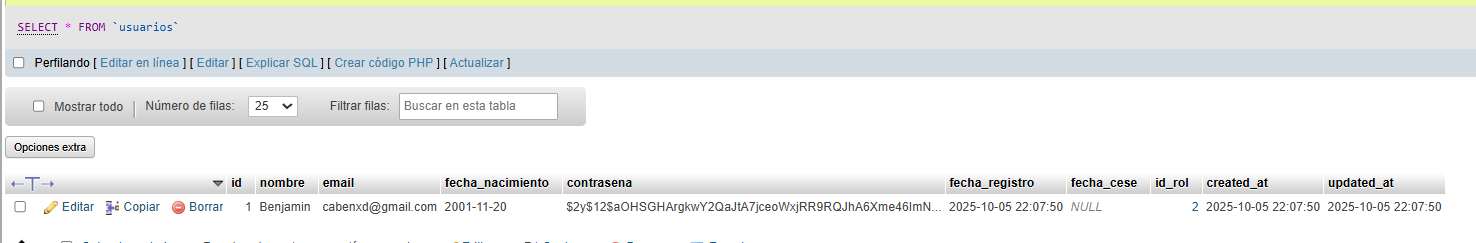






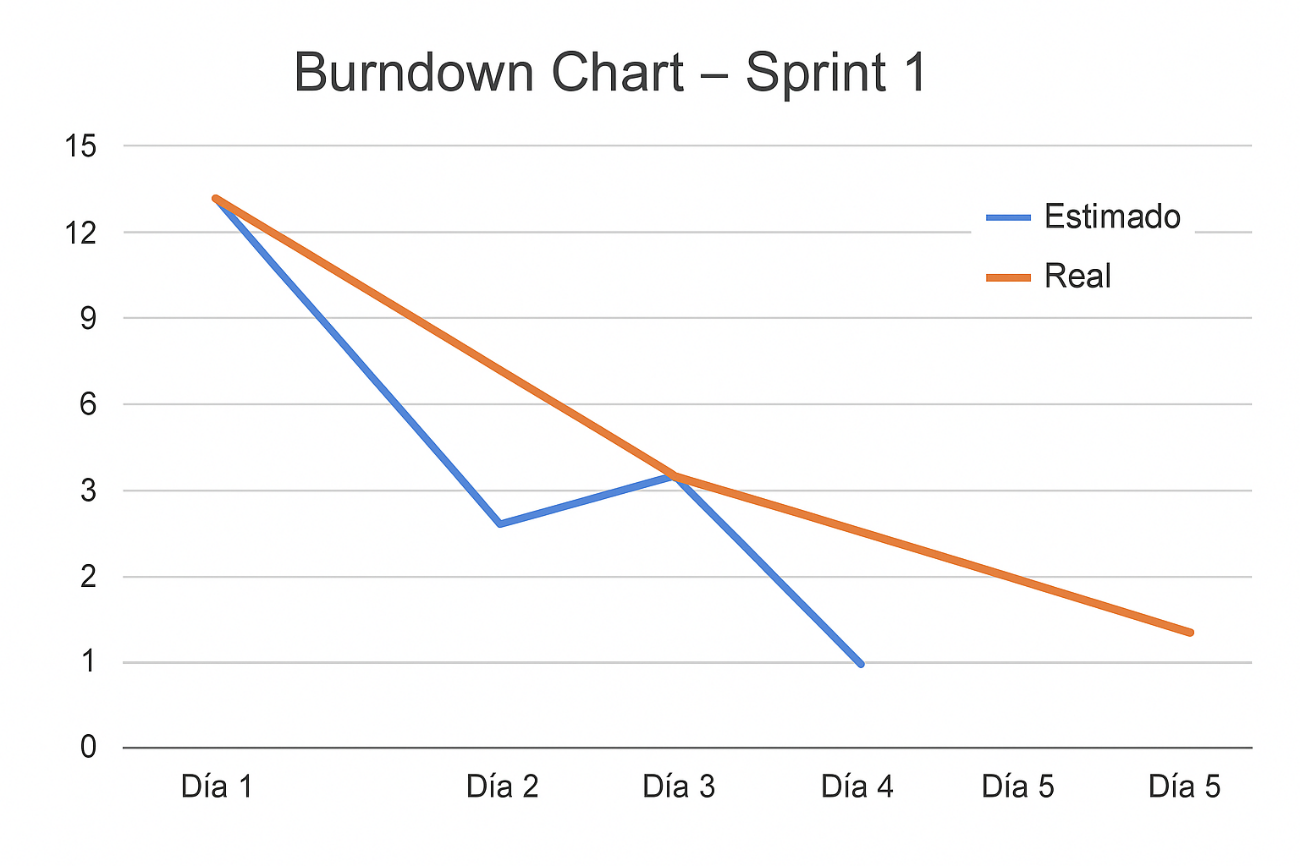






# 6. Burndown Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Backlog ID | Historia de Usuario | Tiempo Estimado | 02/10/2025 | 03/10/2025 |
| **Dia1** | **Dia2** |
| HU-1 | Registro de usuario | 5 | 2 | 3 |
| HU-2 | Asignación de rol | 3 | 2 | 0 |
| HU-3 | Registrar preferencias alimenticias | 5 | 3 | 0 |
| Tiempo de trabajo | | 13 | 5 | 3 |



# 6. Análisis del Sprint

* Se cumplió toda la generación de toda la base de datos del sistema con la inserción de datos
* Se estiman más horas desarrollo por las relaciones y el trato de toda la información
* Se validan los resultados mendiante modals
* Se usa alpine Js para el dinamismo de los botones
* Se presenta npm run dev para el desarrollo y actualización de cambios.

# 7. Conclusiones

El Sprint 1 demoro más tiempo del que debía y esta en proceso de desarrollo sin embargo la seguridad que tiene y el diseño que presenta son potentes y todos verificados.