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 NutriGym 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
* Consultas de métricas de progreso
* Gráficos dinámicos
* Backup
* Recuperación de datos

# 3. Historias de Usuario y Tareas

# Épica

# Sprint 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Historias de Usuario** | **Observaciones** | **Puntos** | **Inicio** | **Fin** |
| HU-01: Como administrador, quiero registrar y gestionar cuentas de nutriólogos, entrenadores y usuarios. | Configuración de roles y autenticación (Login/Register). | 8 | 25-Set | 4-Oct |
| HU-02: Como administrador, quiero visualizar estadísticas globales del gimnasio. | Dashboard inicial con métricas básicas. | 5 | 25-Set | 9-Oct |
| HU-20: Como administrador, quiero realizar copias de seguridad y restaurar la base de datos. | Backup y restauración manual y automática. | 3 | 1-Oct | 9-Oct |

# 4. Tareas

|  |  |  |  |
| --- | --- | --- | --- |
| **SPRINT 1** | **HU-1** | **HU-2** | **HU-3** |
| Historia | Registro y gestión de cuentas | Estadísticas globales | Copias de seguridad |
| TAREAS | Diseñar esquema de BD para usuarios y roles | Crear consultas SQL de resumen | Crear script de backup automático de BD |
| Implementar autenticación (Login/Register) | Implementar dashboard con métricas (usuarios activos, progreso) | Integrar restauración manual desde panel admin |
| Configurar controladores y rutas en backend | Conectar con frontend (gráficos dinámicos) | Pruebas de recuperación de datos |
| Crear interfaz de registro y login |  |  |
| Pruebas de seguridad y sesión |  |  |

# 5. Evidencias de Código

1. HU – 1
   1. Diseñar esquema para usuarios y roles
      1. Usuario

<?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('usuarios', function (Blueprint $table) {

            $table->id('id');

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

            $table->string('email')->unique();

            $table->date('fecha\_nacimiento');

            $table->string('contrasena');

            $table->timestamp('fecha\_registro')->useCurrent();

            $table->timestamp('fecha\_cese')->nullable();

            $table->foreignId('id\_rol')->default(4)

                ->constrained('roles')

                ->onDelete('restrict');

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('usuario');

    }

};

* + 1. Rol

<?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');

    }

};

<?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()],

        ]);

    }

}

* 1. Autenticación (login/register)
     1. Login

<?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('usuario')->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. Registro

<?php

namespace App\Http\Controllers;

use App\Models\Usuario;

use App\Providers\RouteServiceProvider;

use Illuminate\Http\Request;

use Illuminate\Http\RedirectResponse;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Illuminate\Validation\Rules;

class RegistroUsuarioController extends Controller

{

    public function store(Request $request): RedirectResponse

    {

        //dd($request->all());

        $request->validate([

            'nombre' => ['required', 'string', 'max:255'],

            'email' => ['required', 'string', 'email', 'max:255', 'unique:usuarios,email'],

            'fecha\_nacimiento' => ['required','date'],

            'contrasena' => ['required', 'confirmed', Rules\Password::defaults()],

        ]);

        try {

        $usuario = Usuario::create([

            'nombre' => $request->nombre,

            'email' => $request->email,

            'fecha\_nacimiento' => $request->fecha\_nacimiento,

            'fecha\_registro' => now(),

            'contrasena' => Hash::make($request->contrasena),

            'id\_rol' => 2,

        ]);

        Auth::login($usuario);

        return redirect()->route('registrar\_usuario')->with('success', 'Usuario registrado correctamente.');

    } catch (\Exception $e) {

        // Redirige de vuelta al formulario con un flag de error

        return redirect()->back()->with('error', 'No se pudo crear el usuario. ' . $e->getMessage());

    }

    }

}

* 1. Configurar rutas de los controladores

// Login

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

    return view('usuario.login');

})->name('login');

Route::post('/login', [LoginUsuarioController::class, 'validacion'])

    ->name('login.store');

?>

// 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');

// Ruta para procesar el logout

Route::post('logout', function() {

    //Auth::logout();

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

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

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

})->name('logout');

* 1. Crear interfaz registro y login
     1. Registro

@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="w-full max-w-md p-8 neumorphic">

        <h1 class="text-2xl font-bold text-center mb-6 text-[var(--color-primary)]">Registrar Usuario</h1>

        <!-- Formulario -->

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

            @csrf

            <div class="mb-4">

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

                <input type="text" name="nombre" class="input-neu" value="{{ old('nombre') }}" required>

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

            </div>

            <div class="mb-4">

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

                <input type="email" name="email" class="input-neu" value="{{ old('email') }}" required>

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

            </div>

            <div class="mb-4">

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

                <input type="date" name="fecha\_nacimiento" class="input-neu" value="{{ old('fecha\_nacimiento') }}" required>

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

            </div>

            <div class="mb-4">

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

                <input type="password" name="contrasena" class="input-neu" required>

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

            </div>

            <div class="mb-6">

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

                <input type="password" name="contrasena\_confirmation" class="input-neu" required>

            </div>

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

            <button type="button" @click="openModal = true" class="btn-neu w-full text-center">

                Registrarse

            </button>

        </form>

    </div>

    <!-- 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="w-full max-w-md p-8 neumorphic mx-4">

            <h2 class="text-2xl font-bold text-center mb-4 text-[var(--color-primary)]">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="btn-neu flex-1">

                    Confirmar

                </button>

                <!-- Cancelar -->

                <button type="button"

                        @click="openModal = false"

                        class="btn-neu flex-1 bg-gray-300 text-gray-700 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="w-full max-w-md p-8 neumorphic mx-4">

            <h2 class="text-2xl font-bold text-center mb-4 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="btn-neu bg-red-500 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="w-full max-w-md p-8 neumorphic mx-4">

            <h2 class="text-2xl font-bold text-center mb-4 text-[var(--color-primary)]">¡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 ; window.location.href='{{ route('dashboard') }}'"

                    class="btn-neu" >

                    Cerrar

                </button>

            </div>

        </div>

    </div>

</div>

@endsection

* + 1. Login

@extends('layouts.app')

@section('content')

<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

* + 1. Logout

@extends('layouts.app')

@section('content')

<div class="min-h-screen flex items-center justify-center bg-gray-50 py-12 px-4 sm:px-6 lg:px-8">

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

        <div class="text-center mb-6">

            <div class="w-16 h-16 mx-auto mb-4 rounded-full flex items-center justify-center neumorphic-inset">

                <svg xmlns="http://www.w3.org/2000/svg" class="h-8 w-8 text-yellow-500" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-1.667 1.732-2.5L13.732 4c-.77-.833-1.964-.833-2.732 0L4.082 16.5c-.77.833.192 2.5 1.732 2.5z" />

                </svg>

            </div>

            <h2 class="text-2xl font-bold text-gray-800">Cerrar Sesión</h2>

            <p class="text-gray-600 mt-2">¿Estás seguro de que quieres salir?</p>

        </div>

        <form method="POST" action="{{ route('logout') }}" class="space-y-6">

            @csrf

            <div class="flex flex-col sm:flex-row gap-4 justify-center">

                <button type="submit" class="btn-neu flex items-center justify-center">

                    <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5 mr-2" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3 0 01-3 3H6a3 3 0 01-3-3V7a3 3 0 013-3h4a3 3 0 013 3v1" />

                    </svg>

                    Sí, Cerrar Sesión

                </button>

                <a href="{{ url('/') }}" class="btn-neu-secondary flex items-center justify-center">

                    <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5 mr-2" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12" />

                    </svg>

                    Cancelar

                </a>

            </div>

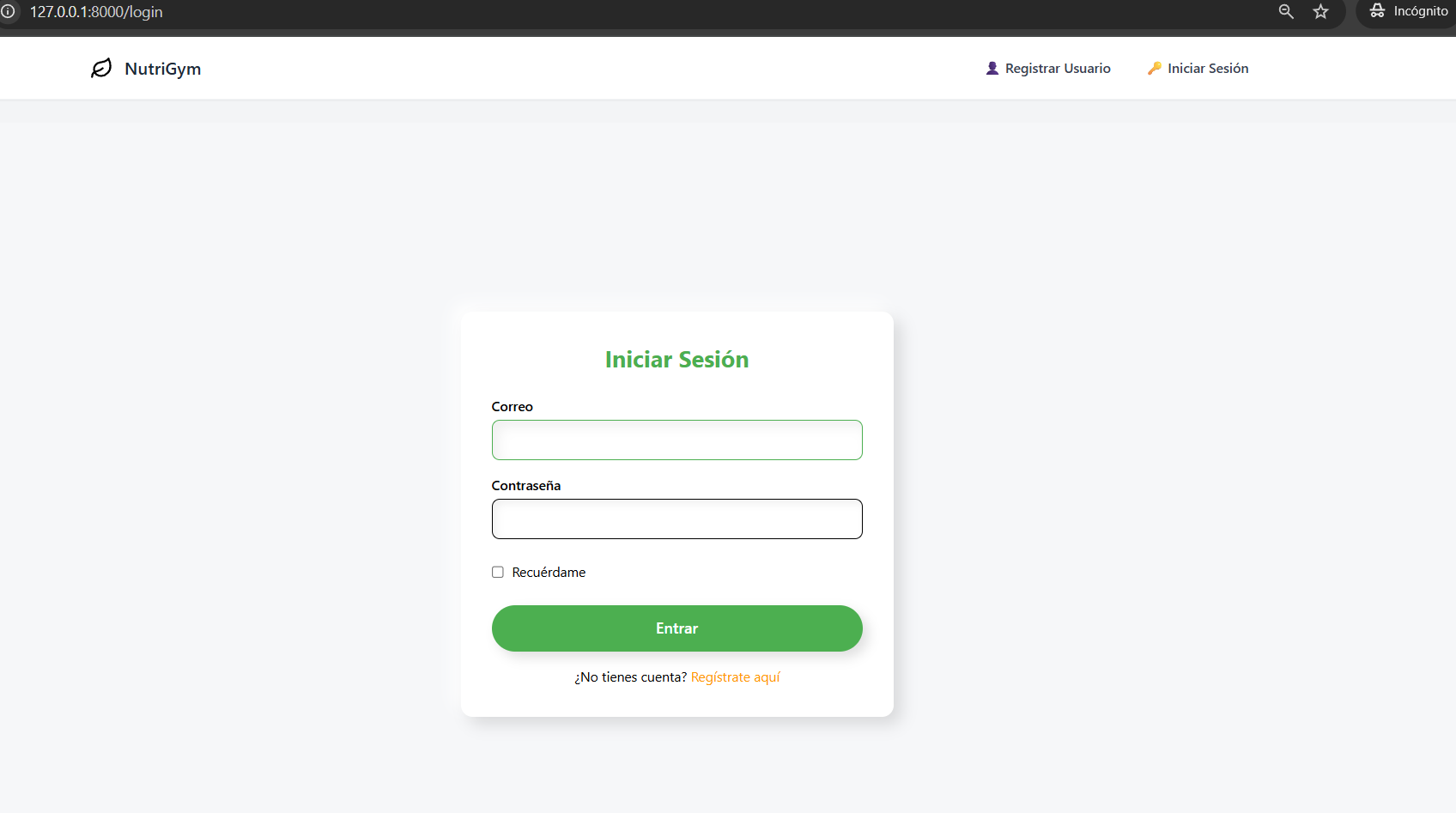
        </form>

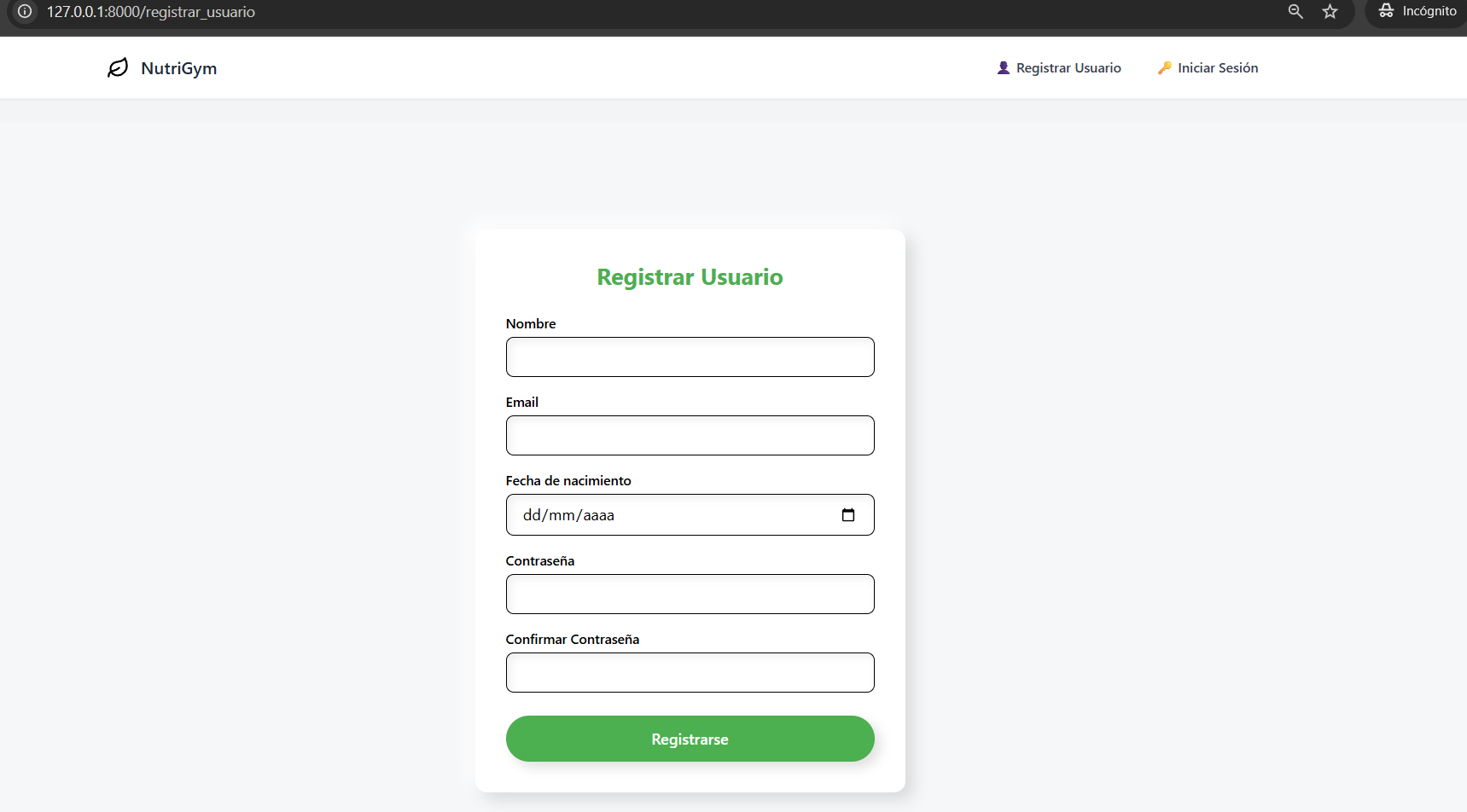
    </div>

</div>

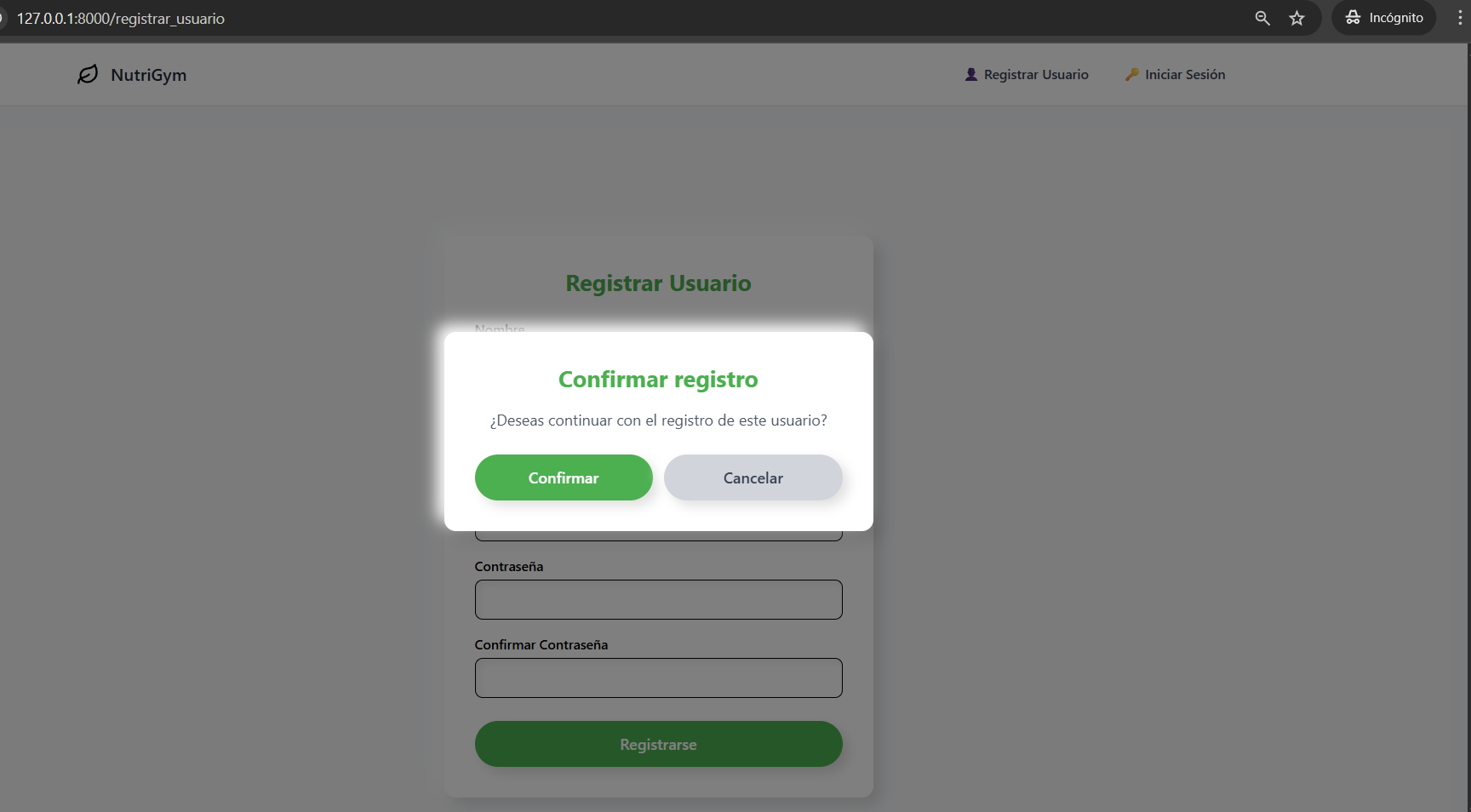
@endsection

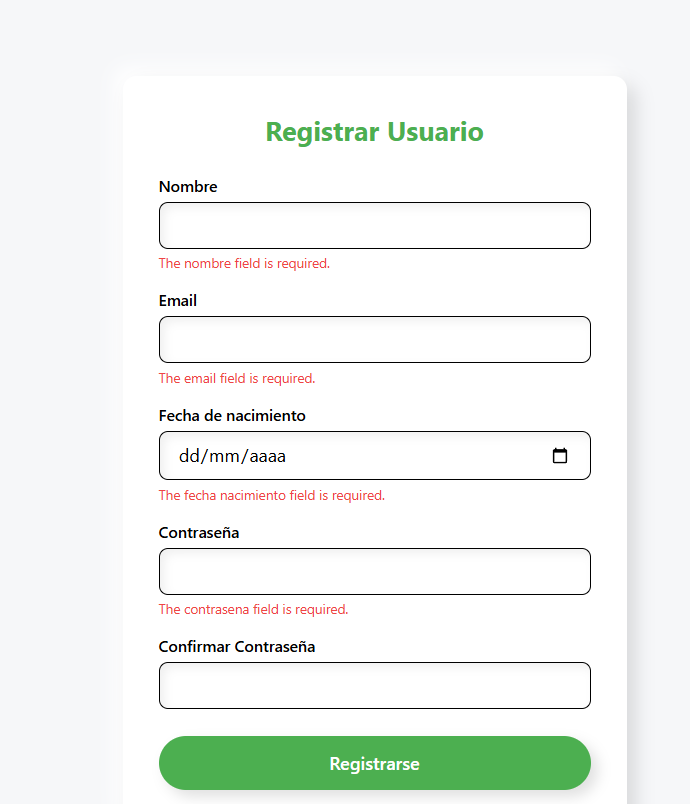
* + 1. Imagen

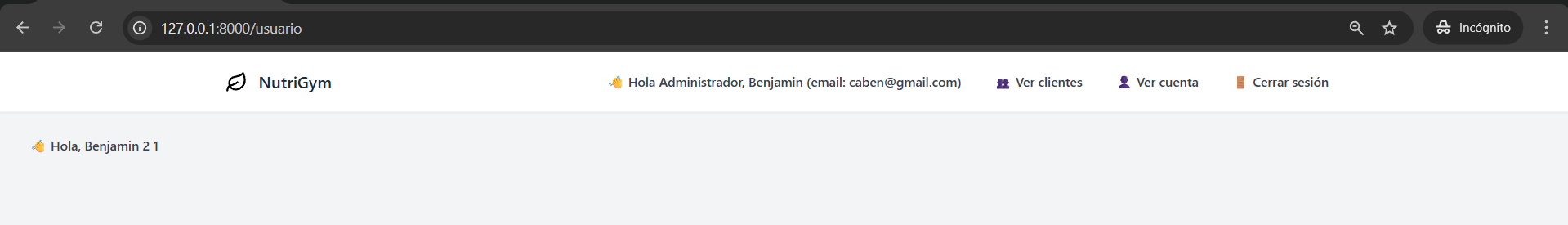




* 1. Pruebas de seguridad







1. HU – 2
   1. Crear consulta de resumen

@extends('layouts.app')

@section('content')

<span class="px-4 py-2 text-sm font-medium text-gray-700">

        Hola, {{ Auth::user()->nombre }}

        {{Auth::user()->id}}

        {{Auth::user()->id\_rol}}

</span>

<!-- component -->

<div class="bg-gray-100">

        <div class="header my-3 h-12 px-10 flex items-center justify-between">

            <h1 class="font-medium text-2xl">Clientes</h1>

        </div>

<div class="flex flex-col lg:flex-row gap-4">

    <!-- Tabla de usuarios -->

    <div class="w-full lg:w-1/3">

        <div class="neumorphic p-6">

            <h3 class="text-xl font-bold text-gray-800 mb-4">Lista de Usuarios</h3>

            <table class="w-full">

                <thead>

                    <tr class="border-b-2 border-gray-200">

                        <th class="text-left p-4 font-semibold text-gray-700">ID</th>

                        <th class="text-left p-4 font-semibold text-gray-700">Nombre</th>

                        <th class="text-left p-4 font-semibold text-gray-700">Rol</th>

                    </tr>

                </thead>

                <tbody id="tablaUsuarios">

                    @foreach($usuarios as $usuario)

                    <tr class="border-b border-gray-100 hover:bg-gray-50 transition duration-200 cursor-pointer"

                        onclick="mostrarUsuario(event)"

                        data-usuario='@json($usuario)'>

                        <td class="p-4 text-gray-600">{{ $usuario->id }}</td>

                        <td class="p-4 text-gray-800 font-medium">{{ $usuario->nombre }}</td>

                        <td class="p-4">

                            <span class="px-3 py-1 rounded-full text-sm font-medium

                                @if($usuario->id\_rol == 2) bg-green-100 text-green-800

                                @elseif($usuario->id\_rol == 3) bg-blue-100 text-blue-800

                                @else bg-gray-100 text-gray-800 @endif">

                                {{ $usuario->nombre\_rol ?? 'Rol ' . $usuario->id\_rol }}

                            </span>

                        </td>

                    </tr>

                    @endforeach

                </tbody>

            </table>

        </div>

    </div>

    <!-- Panel de detalles -->

    <div class="w-full lg:w-2/3">

        <div class="neumorphic p-6">

            <div id="detallesUsuario">

                <div id="mensajeVacio" class="text-center py-8 text-gray-500">

                    Selecciona un usuario para ver sus detalles

                </div>

            </div>

        </div>

    </div>

</div>

<script>

function mostrarUsuario(event) {

    // Obtener la fila clickeada

    const fila = event.currentTarget;

    const usuario = JSON.parse(fila.getAttribute('data-usuario'));

    // Ocultar mensaje vacío

    document.getElementById('mensajeVacio').style.display = 'none';

    // Mostrar detalles del usuario

    document.getElementById('detallesUsuario').innerHTML = `

        <div class="bg-white p-6 rounded-lg">

            <div class="flex justify-between items-center mb-4">

                <h3 class="text-2xl font-bold text-gray-800">Detalles del Usuario</h3>

                <button onclick="limpiarSeleccion()" class="btn-neu bg-gray-500 hover:bg-gray-600 text-white text-sm px-4 py-2">

                    Cerrar

                </button>

            </div>

            <div class="grid grid-cols-1 md:grid-cols-2 gap-4 mb-6">

                <div>

                    <label class="block text-sm font-medium text-gray-600">ID</label>

                    <p class="text-lg text-gray-800">${usuario.id}</p>

                </div>

                <div>

                    <label class="block text-sm font-medium text-gray-600">Nombre</label>

                    <p class="text-lg text-gray-800">${usuario.nombre}</p>

                </div>

                <div>

                    <label class="block text-sm font-medium text-gray-600">Email</label>

                    <p class="text-lg text-gray-800">${usuario.email}</p>

                </div>

                <div>

                    <label class="block text-sm font-medium text-gray-600">Rol</label>

                    <span class="px-3 py-1 rounded-full text-sm font-medium ${

                        usuario.id\_rol == 2 ? 'bg-green-100 text-green-800' :

                        usuario.id\_rol == 3 ? 'bg-blue-100 text-blue-800' :

                        'bg-gray-100 text-gray-800'

                    }">

                        ${usuario.nombre\_rol || 'Rol ' + usuario.id\_rol}

                    </span>

                </div>

                <div>

                    <label class="block text-sm font-medium text-gray-600">Fecha de Nacimiento</label>

                    <p class="text-lg text-gray-800">${new Date(usuario.fecha\_nacimiento).toLocaleDateString()}</p>

                </div>

                <div>

                    <label class="block text-sm font-medium text-gray-600">Fecha de Registro</label>

                    <p class="text-lg text-gray-800">${new Date(usuario.fecha\_registro).toLocaleDateString()}</p>

                </div>

            </div>

            <!-- Botones de acción -->

            <div class="flex space-x-3 mt-6">

                <button class="btn-neu bg-blue-500 hover:bg-blue-600 text-white">

                    Editar Usuario

                </button>

                <button class="btn-neu bg-green-500 hover:bg-green-600 text-white">

                    Asignar Dieta

                </button>

                <button class="btn-neu bg-red-500 hover:bg-red-600 text-white">

                    Eliminar

                </button>

            </div>

        </div>

    `;

    // Remover highlight de todas las filas y agregar a la seleccionada

    document.querySelectorAll('#tablaUsuarios tr').forEach(tr => {

        tr.classList.remove('bg-blue-50', 'border-l-4', 'border-blue-500');

    });

    fila.classList.add('bg-blue-50', 'border-l-4', 'border-blue-500');

}

function limpiarSeleccion() {

    // Mostrar mensaje vacío

    document.getElementById('detallesUsuario').innerHTML = `

        <div id="mensajeVacio" class="text-center py-8 text-gray-500">

            Selecciona un usuario para ver sus detalles

        </div>

    `;

    // Remover highlight de todas las filas

    document.querySelectorAll('#tablaUsuarios tr').forEach(tr => {

        tr.classList.remove('bg-blue-50', 'border-l-4', 'border-blue-500');

    });

}

</script>

@endsection

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\DB;

class AdminController extends Controller

{

    public function index()

    {

        // Listar usuarios

        // Select con join para obtener información del rol

        $usuarios = DB::table('usuarios')

            ->join('roles', 'usuarios.id\_rol', '=', 'roles.id')

            ->where('usuarios.id\_rol', '!=', 1)

            ->select(

                'usuarios.\*',

                'roles.nombre\_rol as nombre\_rol'

            )

            ->get();

        return view('ui\_dashboard.admin', compact('usuarios'));

    }

}

* 1. Dashboard con métricas de progreso

@extends('layouts.app')

@section('content')

<span class="px-4 py-2 text-sm font-medium text-gray-700">

        Hola, {{ Auth::user()->nombre }}

        {{Auth::user()->id}}

        {{Auth::user()->id\_rol}}

</span>

<!-- component -->

<div class="bg-gray-100">

        <div class="header my-3 h-12 px-10 flex items-center justify-between">

            <h1 class="font-medium text-2xl">Clientes</h1>

        </div>

<div class="flex flex-col lg:flex-row gap-4">

    <!-- Tabla de usuarios -->

    <div class="w-full lg:w-1/3">

        <div class="neumorphic p-6">

            <h3 class="text-xl font-bold text-gray-800 mb-4">Lista de Usuarios</h3>

            <table class="w-full">

                <thead>

                    <tr class="border-b-2 border-gray-200">

                        <th class="text-left p-4 font-semibold text-gray-700">ID</th>

                        <th class="text-left p-4 font-semibold text-gray-700">Nombre</th>

                        <th class="text-left p-4 font-semibold text-gray-700">Rol</th>

                    </tr>

                </thead>

                <tbody id="tablaUsuarios">

                    @foreach($usuarios as $usuario)

                    <tr class="border-b border-gray-100 hover:bg-gray-50 transition duration-200 cursor-pointer"

                        onclick="mostrarUsuario(event)"

                        data-usuario='@json($usuario)'>

                        <td class="p-4 text-gray-600">{{ $usuario->id }}</td>

                        <td class="p-4 text-gray-800 font-medium">{{ $usuario->nombre }}</td>

                        <td class="p-4">

                            <span class="px-3 py-1 rounded-full text-sm font-medium

                                @if($usuario->id\_rol == 2) bg-green-100 text-green-800

                                @elseif($usuario->id\_rol == 3) bg-blue-100 text-blue-800

                                @else bg-gray-100 text-gray-800 @endif">

                                {{ $usuario->nombre\_rol ?? 'Rol ' . $usuario->id\_rol }}

                            </span>

                        </td>

                    </tr>

                    @endforeach

                </tbody>

            </table>

        </div>

    </div>

    <!-- Panel de detalles -->

    <div class="w-full lg:w-2/3">

        <div class="neumorphic p-6">

            <div id="detallesUsuario">

                <div id="mensajeVacio" class="text-center py-8 text-gray-500">

                    Selecciona un usuario para ver sus detalles

                </div>

            </div>

        </div>

    </div>

</div>

<script>

    function mostrarUsuario(event) {

        // Obtener la fila clickeada

        const fila = event.currentTarget;

        const usuario = JSON.parse(fila.getAttribute('data-usuario'));

        // Ocultar mensaje vacío

        document.getElementById('mensajeVacio').style.display = 'none';

        // Mostrar detalles del usuario

        document.getElementById('detallesUsuario').innerHTML = `

            <div class="bg-white p-6 rounded-lg">

                <div class="flex justify-between items-center mb-4">

                    <h3 class="text-2xl font-bold text-gray-800">Detalles del Usuario</h3>

                    <button onclick="limpiarSeleccion()" class="btn-neu bg-gray-500 hover:bg-gray-600 text-white text-sm px-4 py-2">

                        Cerrar

                    </button>

                </div>

                <div class="grid grid-cols-1 md:grid-cols-2 gap-4 mb-6">

                    <div>

                        <label class="block text-sm font-medium text-gray-600">ID</label>

                        <p class="text-lg text-gray-800">${usuario.id}</p>

                    </div>

                    <div>

                        <label class="block text-sm font-medium text-gray-600">Nombre</label>

                        <p class="text-lg text-gray-800">${usuario.nombre}</p>

                    </div>

                    <div>

                        <label class="block text-sm font-medium text-gray-600">Email</label>

                        <p class="text-lg text-gray-800">${usuario.email}</p>

                    </div>

                    <div>

                        <label class="block text-sm font-medium text-gray-600">Rol</label>

                        <span class="px-3 py-1 rounded-full text-sm font-medium">

                            ${usuario.nombre\_rol || 'Rol ' + usuario.id\_rol}

                        </span>

                    </div>

                    <div>

                        <label class="block text-sm font-medium text-gray-600">Fecha de Nacimiento</label>

                        <p class="text-lg text-gray-800">${new Date(usuario.fecha\_nacimiento).toLocaleDateString()}</p>

                    </div>

                    <div>

                        <label class="block text-sm font-medium text-gray-600">Fecha de Registro</label>

                        <p class="text-lg text-gray-800">${new Date(usuario.fecha\_registro).toLocaleDateString()}</p>

                    </div>

                </div>

                <!-- Botones de acción -->

                <div class="flex space-x-3 mt-6">

                    <button onclick="openModal('progresoModal')" class="btn-neu bg-blue-500 hover:bg-blue-600 text-white">

                        Ver progreso

                    </button>

                    <button class="btn-neu bg-red-500 hover:bg-red-600 text-white">

                        Eliminar

                    </button>

                </div>

            </div>

        `;

        // Remover highlight de todas las filas y agregar a la seleccionada

        document.querySelectorAll('#tablaUsuarios tr').forEach(tr => {

            tr.classList.remove('bg-blue-50', 'border-l-4', 'border-blue-500');

        });

        fila.classList.add('bg-blue-50', 'border-l-4', 'border-blue-500');

    }

    // Funciones para el modal de progreso

    function openModal(modalId) {

        document.getElementById(modalId).classList.remove('hidden');

    }

    function closeModal(modalId) {

        document.getElementById(modalId).classList.add('hidden');

    }

    function limpiarSeleccion() {

        // Mostrar mensaje vacío

        document.getElementById('detallesUsuario').innerHTML = `

            <div id="mensajeVacio" class="text-center py-8 text-gray-500">

                Selecciona un usuario para ver sus detalles

            </div>

        `;

        // Remover highlight de todas las filas

        document.querySelectorAll('#tablaUsuarios tr').forEach(tr => {

            tr.classList.remove('bg-blue-50', 'border-l-4', 'border-blue-500');

        });

    }

</script>

<div id="progresoModal" class="fixed inset-0 z-50 hidden">

    <div class="modal-backdrop fixed inset-0" onclick="closeModal('progresoModal')"></div>

    <div class="fixed inset-0 flex items-center justify-center p-4">

        <div class="modal-content neumorphic w-full max-w-2xl">

            <div class="p-6">

                <!-- Header -->

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

                    <h3 class="text-xl font-bold text-gray-800 flex items-center">

                        <svg class="w-6 h-6 mr-2 text-purple-500" fill="none" stroke="currentColor" viewBox="0 0 24 24">

                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2 0 00-2 2v6a2 2 0 002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0 012 2v10m-6 0a2 2 0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0 012 2v14a2 2 0 01-2 2h-2a2 2 0 01-2-2z"/>

                        </svg>

                        Mi Progreso

                    </h3>

                    <button onclick="closeModal('progresoModal')" class="text-gray-500 hover:text-gray-700">

                        <svg class="w-6 h-6" fill="none" stroke="currentColor" viewBox="0 0 24 24">

                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12"/>

                        </svg>

                    </button>

                </div>

                <!-- Contenido -->

                <div class="space-y-6">

                    <!-- Resumen de progreso -->

                    <div class="grid grid-cols-2 md:grid-cols-4 gap-4">

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-green-600">-5kg</div>

                            <div class="text-sm text-gray-600">Peso perdido</div>

                        </div>

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-blue-600">12</div>

                            <div class="text-sm text-gray-600">Sesiones/mes</div>

                        </div>

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-orange-600">85%</div>

                            <div class="text-sm text-gray-600">Asistencia</div>

                        </div>

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-purple-600">+8%</div>

                            <div class="text-sm text-gray-600">Fuerza</div>

                        </div>

                    </div>

                    <!-- Gráfico simulado -->

                    <div class="neumorphic-inset p-4 rounded-lg">

                        <h4 class="font-semibold text-gray-700 mb-3">Evolución de peso (últimos 30 días)</h4>

                        <div class="h-32 flex items-end justify-between space-x-1">

                            <div class="flex-1 bg-green-200 rounded-t" style="height: 60%"></div>

                            <div class="flex-1 bg-green-300 rounded-t" style="height: 70%"></div>

                            <div class="flex-1 bg-green-400 rounded-t" style="height: 65%"></div>

                            <div class="flex-1 bg-green-500 rounded-t" style="height: 55%"></div>

                            <div class="flex-1 bg-green-600 rounded-t" style="height: 50%"></div>

                        </div>

                    </div>

                    <!-- Metas -->

                    <div>

                        <h4 class="font-semibold text-gray-700 mb-3">Metas alcanzadas</h4>

                        <div class="space-y-2">

                            <div class="flex items-center">

                                <div class="w-3 h-3 bg-green-500 rounded-full mr-3"></div>

                                <span class="text-sm">Bajar 2kg este mes ✅</span>

                            </div>

                            <div class="flex items-center">

                                <div class="w-3 h-3 bg-yellow-500 rounded-full mr-3"></div>

                                <span class="text-sm">Asistir 15 sesiones (12/15)</span>

                            </div>

                            <div class="flex items-center">

                                <div class="w-3 h-3 bg-red-500 rounded-full mr-3"></div>

                                <span class="text-sm">Correr 5km sin parar</span>

                            </div>

                        </div>

                    </div>

                </div>

                <!-- Footer -->

                <div class="flex gap-3 mt-6">

                    <button onclick="closeModal('progresoModal')" class="btn-neu bg-gray-500 hover:bg-gray-600 text-white flex-1">

                        Cerrar

                    </button>

                    <button class="btn-neu bg-blue-500 hover:bg-blue-600 text-white flex-1">

                        Exportar Reporte

                    </button>

                </div>

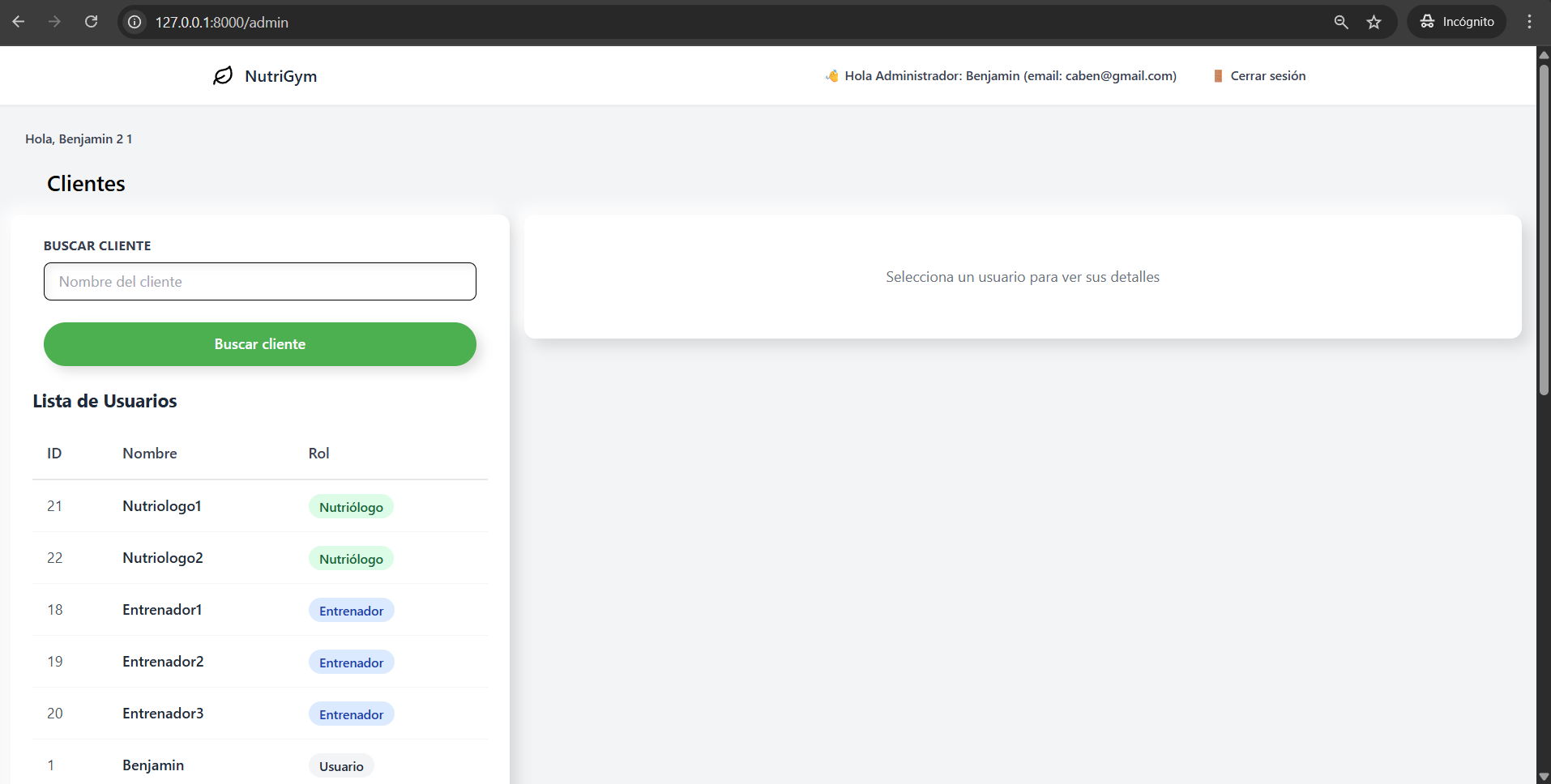
            </div>

        </div>

    </div>

</div>

@endsection



* 1. Gráficos dinámicos

<!-- Modal de progreso (fuera de la función) -->

<div id="progresoModal" class="fixed inset-0 z-50 hidden">

    <div class="modal-backdrop fixed inset-0" onclick="closeModal('progresoModal')"></div>

    <div class="fixed inset-0 flex items-center justify-center p-4">

        <div class="modal-content neumorphic w-full max-w-2xl">

            <div class="p-6">

                <!-- Header -->

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

                    <h3 class="text-xl font-bold text-gray-800 flex items-center">

                        <svg class="w-6 h-6 mr-2 text-purple-500" fill="none" stroke="currentColor" viewBox="0 0 24 24">

                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9 19v-6a2 2 0 00-2-2H5a2 2 0 00-2 2v6a2 2 0 002 2h2a2 2 0 002-2zm0 0V9a2 2 0 012-2h2a2 2 0 012 2v10m-6 0a2 2 0 002 2h2a2 2 0 002-2m0 0V5a2 2 0 012-2h2a2 2 0 012 2v14a2 2 0 01-2 2h-2a2 2 0 01-2-2z"/>

                        </svg>

                        Mi Progreso

                    </h3>

                    <button onclick="closeModal('progresoModal')" class="text-gray-500 hover:text-gray-700">

                        <svg class="w-6 h-6" fill="none" stroke="currentColor" viewBox="0 0 24 24">

                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M6 18L18 6M6 6l12 12"/>

                        </svg>

                    </button>

                </div>

                <!-- Contenido -->

                <div class="space-y-6">

                    <!-- Resumen de progreso -->

                    <div class="grid grid-cols-2 md:grid-cols-4 gap-4">

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-green-600">-5kg</div>

                            <div class="text-sm text-gray-600">Peso perdido</div>

                        </div>

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-blue-600">12</div>

                            <div class="text-sm text-gray-600">Sesiones/mes</div>

                        </div>

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-orange-600">85%</div>

                            <div class="text-sm text-gray-600">Asistencia</div>

                        </div>

                        <div class="neumorphic-inset p-4 text-center rounded-lg">

                            <div class="text-2xl font-bold text-purple-600">+8%</div>

                            <div class="text-sm text-gray-600">Fuerza</div>

                        </div>

                    </div>

                    <!-- Gráfico simulado -->

                    <div class="neumorphic-inset p-4 rounded-lg">

                        <h4 class="font-semibold text-gray-700 mb-3">Evolución de peso (últimos 30 días)</h4>

                        <div class="h-32 flex items-end justify-between space-x-1">

                            <div class="flex-1 bg-green-200 rounded-t" style="height: 60%"></div>

                            <div class="flex-1 bg-green-300 rounded-t" style="height: 70%"></div>

                            <div class="flex-1 bg-green-400 rounded-t" style="height: 65%"></div>

                            <div class="flex-1 bg-green-500 rounded-t" style="height: 55%"></div>

                            <div class="flex-1 bg-green-600 rounded-t" style="height: 50%"></div>

                        </div>

                    </div>

                    <!-- Metas -->

                    <div>

                        <h4 class="font-semibold text-gray-700 mb-3">Metas alcanzadas</h4>

                        <div class="space-y-2">

                            <div class="flex items-center">

                                <div class="w-3 h-3 bg-green-500 rounded-full mr-3"></div>

                                <span class="text-sm">Bajar 2kg este mes ✅</span>

                            </div>

                            <div class="flex items-center">

                                <div class="w-3 h-3 bg-yellow-500 rounded-full mr-3"></div>

                                <span class="text-sm">Asistir 15 sesiones (12/15)</span>

                            </div>

                            <div class="flex items-center">

                                <div class="w-3 h-3 bg-red-500 rounded-full mr-3"></div>

                                <span class="text-sm">Correr 5km sin parar</span>

                            </div>

                        </div>

                    </div>

                </div>

                <!-- Footer -->

                <div class="flex gap-3 mt-6">

                    <button onclick="closeModal('progresoModal')" class="btn-neu bg-gray-500 hover:bg-gray-600 text-white flex-1">

                        Cerrar

                    </button>

                    <button class="btn-neu bg-blue-500 hover:bg-blue-600 text-white flex-1">

                        Exportar Reporte

                    </button>

                </div>

            </div>

        </div>

    </div>

</div>



1. HU - 3
   1. Script de backup

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\File;

use Carbon\Carbon;

class BackupController extends Controller

{

    public function createBackup(Request $request)

    {

        try {

            $backupName = 'backup\_' . Carbon::now()->format('Y-m-d\_H-i-s');

            $backupPath = storage\_path('app/backups/' . $backupName);

            // Crear directorio

            if (!File::exists($backupPath)) {

                File::makeDirectory($backupPath, 0755, true);

            }

            // Backup de archivos

            $this->backupFiles($backupPath);

            // SIN compresión - solo copiamos los archivos

            $backupSize = $this->calculateFolderSize($backupPath);

            return response()->json([

                'success' => true,

                'message' => 'Backup creado exitosamente (sin compresión)',

                'backup\_folder' => $backupName,

                'backup\_path' => $backupPath,

                'backup\_size' => $this->formatBytes($backupSize),

                'note' => 'Los archivos están en: storage/app/backups/' . $backupName

            ]);

        } catch (\Exception $e) {

            return response()->json([

                'success' => false,

                'message' => 'Error al crear backup: ' . $e->getMessage()

            ], 500);

        }

    }

    private function backupFiles($backupPath)

    {

        $filesPath = $backupPath . '/files';

        File::makeDirectory($filesPath);

        // Archivos importantes

        $importantFiles = [

            '.env' => base\_path('.env'),

            'composer.json' => base\_path('composer.json'),

            'composer.lock' => base\_path('composer.lock'),

            'package.json' => base\_path('package.json'),

        ];

        foreach ($importantFiles as $name => $path) {

            if (File::exists($path)) {

                File::copy($path, $filesPath . '/' . $name);

            }

        }

        // Directorios importantes

        $importantDirs = [

            'app' => base\_path('app'),

            'config' => base\_path('config'),

            'database' => base\_path('database'),

            'resources' => base\_path('resources'),

            'routes' => base\_path('routes'),

            'public' => base\_path('public'),

        ];

        foreach ($importantDirs as $name => $path) {

            if (File::exists($path)) {

                File::copyDirectory($path, $filesPath . '/' . $name);

            }

        }

    }

    private function calculateFolderSize($path)

    {

        $size = 0;

        $files = new \RecursiveIteratorIterator(

            new \RecursiveDirectoryIterator($path),

            \RecursiveIteratorIterator::LEAVES\_ONLY

        );

        foreach ($files as $file) {

            if ($file->isFile()) {

                $size += $file->getSize();

            }

        }

        return $size;

    }

    private function formatBytes($size, $precision = 2)

    {

        if ($size == 0) return '0 B';

        $units = ['B', 'KB', 'MB', 'GB'];

        $bytes = max($size, 0);

        $pow = floor(($bytes ? log($bytes) : 0) / log(1024));

        $pow = min($pow, count($units) - 1);

        $bytes /= pow(1024, $pow);

        return round($bytes, $precision) . ' ' . $units[$pow];

    }

}

* 1. Integrar restauración

<div class="card">

    <div class="card-header">

        <h5>🔄 Sistema de Backup</h5>

    </div>

    <div class="card-body">

        <p class="text-muted">Haz clic en el botón para crear un backup completo del sistema.</p>

        <button id="backupBtn" class="btn btn-primary">

            <i class="fas fa-database"></i> Generar Backup Ahora

        </button>

        <div id="backupResult" class="mt-3" style="display: none;"></div>

        <div id="backupLoading" class="mt-3" style="display: none;">

            <div class="spinner-border text-primary" role="status">

                <span class="visually-hidden">Generando backup...</span>

            </div>

            <span class="ms-2">Generando backup, por favor espere...</span>

        </div>

    </div>

</div>

document.getElementById('backupBtn').addEventListener('click', function() {

    const btn = this;

    const resultDiv = document.getElementById('backupResult');

    const loadingDiv = document.getElementById('backupLoading');

    btn.disabled = true;

    btn.innerHTML = '<i class="fas fa-spinner fa-spin"></i> Generando...';

    loadingDiv.style.display = 'block';

    resultDiv.style.display = 'none';

    fetch('{{ route("backup.create") }}', {

        method: 'POST',

        headers: {

            'Content-Type': 'application/json',

            'X-CSRF-TOKEN': '{{ csrf\_token() }}',

            'Accept': 'application/json'

        }

    })

    .then(response => response.json())

    .then(data => {

        loadingDiv.style.display = 'none';

        resultDiv.style.display = 'block';

        if (data.success) {

            resultDiv.innerHTML = `

                <div class="alert alert-success">

                    <h6>✅ Backup Completado Exitosamente</h6>

                    <p>${data.message}</p>

                    <div class="mt-2">

                        <strong>Carpeta:</strong> ${data.backup\_folder}<br>

                        <strong>Tamaño:</strong> ${data.backup\_size}

                    </div>

                    <small class="text-muted">${data.note}</small>

                </div>

            `;

        } else {

            resultDiv.innerHTML = `

                <div class="alert alert-danger">

                    <h6>❌ Error en el Backup</h6>

                    <p>${data.message}</p>

                </div>

            `;

        }

    })

    .catch(error => {

        loadingDiv.style.display = 'none';

        resultDiv.style.display = 'block';

        resultDiv.innerHTML = `

            <div class="alert alert-danger">

                <h6>❌ Error de Conexión</h6>

                <p>${error.message}</p>

            </div>

        `;

    })

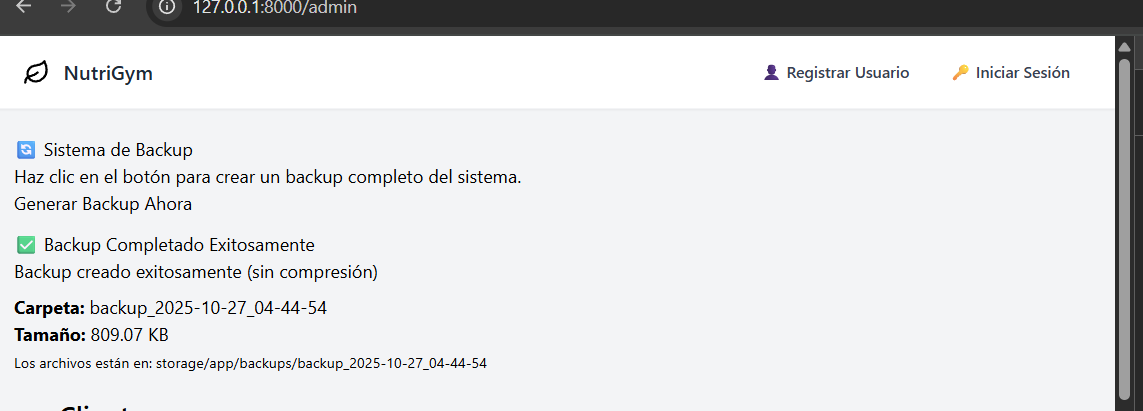
    .finally(() => {

        btn.disabled = false;

        btn.innerHTML = '<i class="fas fa-database"></i> Generar Backup Ahora';

    });

});



* 1. Pruebas de recuperación de datos

# 

# 6. Burndown Chart

# 

# 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 mediante 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.