

Universidad Americana

Estudiante: Javier Carballo Granados

Andrey Matarrita Vasques

Gabriel Najera Fernandez

Tye Robinson Brooks

Profesor: Andrey Padias Calvo

Programacion III

Proyecto Final

Tema: Sistema de Gestion de Gymnacios



Sistema de Gestion de Gimnacios

Descripción del sistema y sus funcionalidades

1. Resumen ejecutivo

GymManagementSystem es una aplicación web ASP.NET Core MVC con Entity Framework Core y ASP.NET Core Identity para la autenticación y autorización. El sistema permite administrar miembros, entrenadores, clases de gimnasio, reservas y equipos. Incluye inicio de sesión, registro de usuarios, gestión de roles (Admin, Manager, Trainer, Member), y reglas de negocio para disponibilidad de cupos y cancelaciones.

2. Arquitectura y tecnologías

- ASP.NET Core MVC (patrón Modelo–Vista–Controlador).
- Entity Framework Core (migraciones, DbContext y relaciones).
- ASP.NET Core Identity con la entidad personalizada Member.
- SQL Server como base de datos relacional.
- Razor Views y ViewModels fuertemente tipados.
- Inyección de dependencias (userManager, signInManager, ApplicationDbContext).

3. Seguridad y autenticación

El sistema utiliza ASP.NET Core Identity con la clase Member como usuario principal. La configuración establece políticas de contraseña (mínimo 6 caracteres, dígito y mayúscula requeridos, símbolos opcionales) y exige correos únicos. Se habilitan roles: Admin, Manager, Trainer y Member. El pipeline define rutas por defecto hacia Home/Index y una ruta adicional para forzar el inicio en Account/Login; además, un middleware redirige cualquier intento de acceder a /Identity hacia /Account/Login. Durante el primer arranque, se crean (seed) los roles mencionados y un usuario administrador por defecto (admin@gym.com) con roles Admin y Manager.

Permisos principales por rol

Rol	Permisos destacados
Admin	CRUD de miembros; CRUD de entrenadores; CRUD de clases; Configura
Manager	CRUD de entrenadores; CRUD de clases; consulta general.
Trainer	Acceso a listado de clases; interacción operativa relacionada con clases a
Member	Ver clases disponibles; reservar (enroll); ver Mis reservas; cancelar con 24h de anticipación.

4. Módulos y funcionalidades

• Autenticación y cuentas (AccountController)

Flujos de autenticación: login, registro, perfil, cambio de contraseña y logout con ASP.NET Core Identity. Soporta redirecciones a 2FA cuando está habilitado, bloqueo por intentos, y vistas para AccessDenied. El perfil permite actualizar nombre y apellido.

• Página principal (HomeController)

Controlador con [Authorize] global que expone Index y Privacy, y maneja errores con respuesta no cacheada (Error + ErrorViewModel).

• Gestión de miembros (MembersController)

CRUD completo de miembros, restringido a Admin. Crea usuarios con rol Member por defecto, valida concurrencia y usa TempData para mensajes.

• Gestión de entrenadores (TrainersController)

CRUD de entrenadores restringido a Admin y Manager. Impide eliminar entrenadores con clases asignadas.

• Clases de gimnasio (GymClassesController)

Listado filtrado para Members (solo futuras y con cupo). Crear/editar con validación de no programar en el pasado; eliminar solo si no hay reservas activas. Detalle de clases con asistentes, estado, acciones permitidas y estadísticas. Enroll permite reservar; MyBookings lista reservas activas y permite cancelar con 24h de anticipación.

• Reservas (BookingsController)

Alias de navegación hacia GymClasses/MyBookings. Requiere autenticación y roles válidos.

• Equipos (Equipment)

Modelo para controlar inventario y mantenimiento de equipos: nombre, descripción, fechas de compra/mantenimiento y estado (Operational/Maintenance/Out of Service).

5. Modelos de dominio

• Member

Entidad principal de Identity extendida: Name, LastName, RegistrationDate, MembershipExpiration, MembershipType, relación con MemberClasses.

- **Trainer**

Entrenador con datos de contacto, especialización, Bio, HireDate, IsActive y relación 1–N con GymClass. Propiedad calculada FullName y método IsAvailable(date).

- **GymClass**

Clase con Name, Description, Schedule, DurationMinutes, MaxParticipants, InstructorId/Instructor y relación N–N (vía MemberClass) con Member. Propiedades calculadas EndTime, AvailableSpots, Status.

- **MemberClass**

Tabla puente enriquecida que almacena BookingDate, Attended, Rating, Feedback, IsCancelled/CancellationDate; FK a Member y a GymClass.

- **Equipment**

Equipo del gimnasio con información de estado y mantenimiento.

6. ViewModels principales

Autenticación y perfil

- LoginViewModel – email, password, remember me, returnUrl.
- RegisterViewModel – email, password/confirm, name/lastName, membershipType (lista: Basic, Premium, VIP).
- ProfileViewModel – email (solo lectura), name, lastName.
- ChangePasswordViewModel – current/new/confirm password.

Clases y reservas

- GymClassViewModel – datos principales, EndTime, AvailableSpots, Status, InstructorName; método estático GetStatus(...).
- GymClassCreateEditViewModel – datos para crear/editar clases y SelectList de entrenadores.
- GymClassDetailViewModel – detalle con asistentes y estadísticas; flags CanEdit/CanCancel/CanMarkAttendance.
- MemberBookingViewModel – reservas del usuario: clase, horario, duración, instructor, estado y posibilidad de cancelar.
- TrainerViewModel – FullName, Contacto, Especialización, HireDate, ActiveClassesCount, IsActive/Status.
- AttendeeViewModel – datos de asistentes a una clase, contacto, estado de inscripción y feedback.

7. Base de datos y migraciones

El modelo de datos incluye tablas de Identity (AspNetUsers/AspNetRoles y auxiliares) y tablas de dominio:

Trainers, GymClasses, MemberClasses y Equipment. Relaciones destacadas: MemberClass enlaza

Member→GymClass con eliminación en cascada; GymClass→Trainer usa eliminación restringida (no borra el entrenador si hay clases). Migraciones registradas: InitialCreate (creación de tablas e índices principales), LOGIN

(valor por defecto 'Basic' para MembershipType), así como migraciones base/placeholder (Inicio, INICIADOR, SetDefaultMembershipType) sin operaciones aplicadas. El snapshot del modelo refleja el estado actual de la base de datos y sus restricciones.

8. Reglas de negocio clave

- Los Members solo ven clases futuras con cupo disponible.
- No se pueden programar clases en el pasado (validación en Create/Edit).
- No se puede eliminar una clase con reservas activas.
- Reserva (Enroll) impide duplicados y respeta capacidad máxima.
- Cancelación de reserva permitida solo hasta 24 horas antes del inicio.
- Estados de clase calculados: Disponible, Llena, En curso, Finalizada.

9. Ruteo y pipeline HTTP

Se definen rutas convencionales para controladores/acciones. Se incluye una ruta para Account/Login y otra para Home/Index, y middleware que redirige /Identity a /Account/Login. El pipeline incorpora HTTPS redirection, archivos estáticos, autenticación y autorización. En desarrollo, UseMigrationsEndPoint; en producción, manejo de errores y HSTS.

10. Requisitos y configuración

- Configurar cadena de conexión 'DefaultConnection' hacia SQL Server.
- Ejecutar migraciones de EF Core para crear/actualizar el esquema.
- Asegurar variables/secrets para credenciales de BD en entornos productivos.
- Opcional: habilitar 2FA en Identity y configurar proveedor de correo/SMS para confirmaciones.

11. Observaciones finales

El sistema está orientado a operación diaria de un gimnasio: alta y administración de socios, programación y control de clases, gestión de entrenadores y control básico de equipos. La seguridad se sustenta en Identity con roles y políticas de contraseñas, y se proporcionan modelos de vista específicos para separar responsabilidades entre dominio, validación y presentación.