

# OBLIGATORIO PROGRAMACIÓN 3

Gimnasios “24”

## ANALISTA PROGRAMADOR

CTC COLONIA. ENSEÑANZA TÉCNICA PROFESIONAL

Alumno: Facundo Tramuja.

Profesor: Agustín Maciel.

Fecha de entrega: mayo 24, 2024.

## Índice:

• Declaración de autoría.....	3
• Problema para resolver.....	3
• Análisis y solución (Relevamiento de Requerimientos).....	3
• Alcances Obtenidos a Consideración del Equipo.....	4
• Diagrama de Clases.....	6
• Diagrama General de Casos de Uso.....	7
• Desarrollo de Dos Casos de Uso.....	7
• Código de las clases.....	8

## Declaración de autoría:

Por este presente, yo Facundo Tramuja declaro que el presente proyecto está realizado por mí.

- Fue realizado en el primer semestre en la institución CTC, en la materia programación 3, dictada por Agustín.
- Aplicando conocimientos adquiridos a lo largo del semestre, como también conocimientos de trabajo de investigación.

## Problema para resolver:

La red de gimnasios "24" necesita modernizar su sistema de gestión para su negocio, para administrar eficientemente sus operaciones, que incluyen la gestión de socios, locales y máquinas. Los socios son usuarios registrados con información personal y tipo de este. Los locales son instalaciones donde se conoce su nombre, ubicación y teléfono, cada una bajo la responsabilidad de una persona, quien es el encargado. Las máquinas en cada local tienen datos de compra, tipo, y estado. El sistema debe proporcionar funcionalidades de administración (ABM) para todas estas entidades y permitir un conjunto de consultas para facilitar las operaciones y las tomas de decisiones de los administradores.

## Análisis y solución (Relevamiento de Requerimientos):

### *Requerimientos Funcionales:*

#### **1. Gestión de Socios:**

- Se requiere alta, baja y modificación de socios.
- Se debe registrar nombre, tipo de usuario (mensual, anual, vitalicio), teléfono, correo y local donde concurra este.
- Se necesita filtrar socios por tipo de usuario a los socios.

#### **2. Gestión de Locales:**

- Se requiere alta, baja y modificación de locales.

- Se debe registrar nombre, ciudad, dirección, teléfono y responsable del local (nombre y teléfono).

### 3. **Gestión de Máquinas:**

- Se requiere alta, baja y modificación de máquinas.
- Se debe registrar local asignado, fecha de compra, precio de compra, vida útil (en años), tipo de máquina y estado.
- Se requiere alta, baja y modificación de tipos de máquinas.
- Se necesita consultar tipos de máquinas existentes en todos los locales y cantidad de cada tipo.
- Se precisa filtrar máquinas por local y tipo.
- Se debe calcular vida útil restante de una máquina específica.
- Se pide ordenar máquinas por fecha de compra (ascendente y descendente).

### *Requerimientos No Funcionales:*

#### 4. **Usabilidad:**

- Interfaz intuitiva y fácil de usar para administradores.

#### 5. **Performance:**

- Respuesta rápida en consultas y operaciones de administración.

#### 6. **Mantenibilidad:**

- Código estructurado y documentado para facilitar futuras modificaciones y mantenimiento.

## Alcances Obtenidos a Consideración del Equipo:

### *Funcionalidad Básica de los ABM:*

#### 1. **Gestionar Socios:**

- **Agregar Socios:** Se permite la creación de nuevos socios con todos los campos requeridos.
- **Modificar Socios:** Se permite la edición de información de los socios.
- **Eliminar Socios:** Se permite la eliminación de socios del sistema.
- **Visualización de Socios:** Se provee una interfaz para listar y buscar socios de acuerdo con distintos criterios, como tipo de usuario.

#### 2. **Gestionar Locales:**

- **Agregar Locales:** Se permite la creación de nuevos locales con campos como nombre, ciudad, dirección, teléfono y responsable.
- **Modificar Locales:** Se permite la edición de información existente de los locales.
- **Eliminar Locales:** Se permite la eliminación de locales del sistema.

### 3. Gestionar Máquinas:

- **Agregar Máquinas:** Se permite la creación de nuevas máquinas con datos como local donde está ubicada, fecha de compra, precio, vida útil, tipo de máquina y su estado.
- **Modificar Máquinas:** Se permite la edición de información de las máquinas.
- **Eliminar Máquinas:** Se permite la eliminación de máquinas del sistema.
- **Visualización de Máquinas:** Se provee una interfaz para listar y buscar máquinas según diferentes criterios, como tipo, local y estado que se encuentra.

### *Consultas Realizadas:*

#### 4. Filtrado y Visualización de Socios por Tipo de Usuario:

- Se provee herramientas para que los administradores puedan filtrar y visualizar a los socios según su tipo de usuario (mensual, anual, vitalicio).

#### 5. Identificación y Conteo de Tipos de Máquinas:

- Se provee una funcionalidad que permita a los administradores identificar los tipos de máquinas existentes en todos los locales y la cantidad de cada tipo.
- Se incluye la posibilidad de filtrar esta información por local para obtener una vista más específica.

#### 6. Cálculo de Vida Útil Restante de una Máquina:

- Se provee una herramienta que calcula y muestra los años de vida útil restantes de una máquina específica basada en su fecha de compra y vida útil total.

#### 7. Ordenación de Máquinas por Fecha de Compra:

- Se permite a los administradores ordenar las máquinas por fecha de compra en orden ascendente y descendente.

### *Interfaz de Usuario:*

#### 8. Interfaz Intuitiva y Accesible:

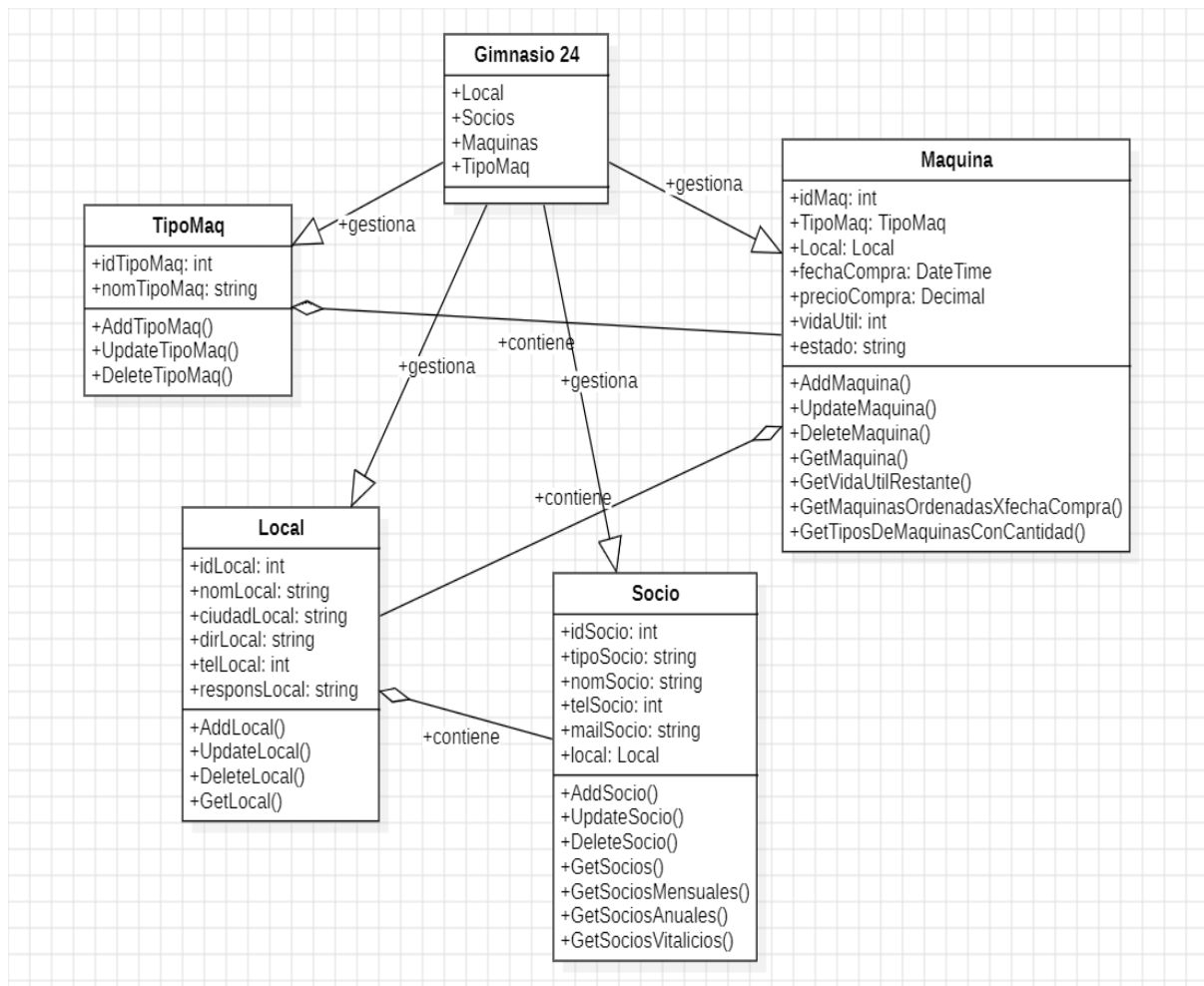
- Diseño de una interfaz de usuario amigable y fácil de navegar, asegurando que las funcionalidades de ABM y consultas sean accesibles.
- Se provee formularios claros y consistentes para la entrada y modificación de datos.
- Se incluye mecanismos de búsqueda y filtrado intuitivos para facilitar la localización de información específica.

## Mantenibilidad:

### 9. Código Estructurado y Documentado:

- Se asegura que el código esté bien estructurado y documentado, facilitando futuras modificaciones y mantenimiento.

## Diagrama de Clases:



## Diagrama General de Casos de Uso:

Administrador

1. Gestionar Socios.
2. Gestionar Locales.
3. Gestionar Máquinas.
4. Consultar Socios por Tipo.
5. Consultar Máquinas por Tipo y Cantidad.
6. Consultar Máquinas por Local.
7. Calcular Vida Útil Restante de Máquina.
8. Ordenar Máquinas por Fecha de Compra.

## Desarrollo de Dos Casos de Uso:

### *Caso de Uso 1: Gestionar Socios:*

- **Actor Principal:** Administrador
- **Propósito:** Permitir al administrador agregar, modificar y eliminar información de los socios.
- **Precondiciones:** El administrador debe estar autenticado.
- **Escenario Principal:**
  - El administrador selecciona la opción de gestionar socios.
  - El administrador elige agregar, modificar o eliminar un socio.
  - Para agregar un socio, ingresa el nombre, tipo, teléfono, email y local.
  - Para modificar un socio, selecciona el socio de la lista y actualiza los datos necesarios.
  - Para eliminar un socio, selecciona el socio y confirma la eliminación.
- **Postcondiciones:** La base de datos se actualiza con la información correspondiente.

### *Caso de Uso 2: Consultar Máquinas por Tipo y Cantidad:*

- **Actor Principal:** Administrador
- **Propósito:** Permitir al administrador ver los tipos de máquinas y la cantidad de cada tipo en todos los locales.
- **Precondiciones:** El administrador debe estar autenticado.
- **Escenario Principal:**

- El administrador selecciona la opción de consultar máquinas por tipo y cantidad.
- El sistema muestra una lista de todos los tipos de máquinas y la cantidad disponible de cada tipo.
- El administrador puede filtrar la lista por local si lo desea.
- **Postcondiciones:** El administrador visualiza la información solicitada.

## Código de las clases:

### *Clase Local:*

La clase Local representa una sucursal de la red de gimnasios "24". Almacena información sobre cada local, como su id, nombre, ciudad, dirección, teléfono y el responsable del local.

```
public class Local
{
    1 referencia
    public int IdLocal { get; set; }
    1 referencia
    public string NomLocal { get; set; }
    1 referencia
    public string CiudadLocal { get; set; }
    1 referencia
    public string DirLocal { get; set; }
    1 referencia
    public int TelLocal { get; set; }
    1 referencia
    public string ResponsLocal { get; set; }

    1 referencia
    public Local(int idLocal, string nomLocal, string ciudadLocal, string dirLocal, int telLocal, string responsLocal)
    {
        IdLocal = idLocal;
        NomLocal = nomLocal;
        CiudadLocal = ciudadLocal;
        DirLocal = dirLocal;
        TelLocal = telLocal;
        ResponsLocal = responsLocal;
    }
}
```

### *Clase Maquina:*

La clase Maquina representa a estas en la red de gimnasios "24". Se almacena información sobre cada máquina, incluyendo su id, el local al que pertenece, el tipo de máquina, la fecha de compra, el precio de compra, la vida útil y su estado actual.



```

public class Maquina
{
    1 referencia
    public int IdMq { get; set; }
    3 referencias
    public Local Local { get; set; }
    3 referencias
    public TipoMq TipoMq { get; set; }
    1 referencia
    public DateTime FechaCompra { get; set; }
    1 referencia
    public decimal PrecioCompra { get; set; }
    3 referencias
    public int VidaUtil { get; set; }
    1 referencia
    public string Estado { get; set; }

    4 referencias
    public Maquina(int idMq, Local local, TipoMq tipoMq, DateTime fechaCompra, decimal precioCompra, int vidaUtil, string estado)
    {
        IdMq = idMq;
        Local = local;
        TipoMq = tipoMq;
        FechaCompra = fechaCompra;
        PrecioCompra = precioCompra;
        VidaUtil = vidaUtil;
        Estado = estado;
    }
}

```

### Clase Socio:

La clase Socio representa a un socio del gimnasio "24". Esta clase almacena información personal y de afiliación de cada socio, incluyendo su identificación, nombre, tipo de membresía, número de teléfono, correo electrónico y el local al que está afiliado.

```

33 referencias
public class Socio
{
    1 referencia
    public int IdSocio { get; set; }
    1 referencia
    public string NomSocio { get; set; }
    1 referencia
    public string TipoSocio { get; set; }
    1 referencia
    public int TelSocio { get; set; }
    1 referencia
    public string MailSocio { get; set; }
    3 referencias
    public Local Local { get; set; }

    10 referencias
    public Socio(int idSocio, string nomSocio, string tipoSocio, int telSocio, string mailSocio, Local local)
    {
        IdSocio = idSocio;
        NomSocio = nomSocio;
        TipoSocio = tipoSocio;
        TelSocio = telSocio;
        MailSocio = mailSocio;
        Local = local;
    }
}

```

### Clase TipoMq:

La clase TipoMq representa al tipo de máquina. Esta clase almacena información básica sobre los diferentes tipos de máquinas disponibles.

```
public class TipoMaq
{
    public int idTipoMaq;
    public string nomTipoMaq;

    0 referencias
    public TipoMaq(int idTipoMaq, string nomTipoMaq)
    {
        this.idTipoMaq = idTipoMaq;
        this.nomTipoMaq = nomTipoMaq;
    }
}
```