

Sistema de programas de fidelización

Una compañía maneja programas de fidelización (clase `ProgramaFidelizacion`) para empresas (clase `ProgramaSocio`) que ofrecen a sus clientes varios tipos de bonos.

Se obtienen puntos de bonificación o millas aéreas, u otras bonificaciones posibles. Cualquier cosa que una compañía esté dispuesta a ofrecer puede ser un servicio (clase `Servicio`) prestado en un programa de fidelización. Cada cliente puede entrar en el programa de fidelización obteniendo una tarjeta de afiliación (clase `TarjetaCliente`). Los objetos de la clase `Cliente` representan a las personas que han entrado en el programa. La tarjeta de miembro se emite a una persona, pero puede ser utilizada para toda una familia o negocio.

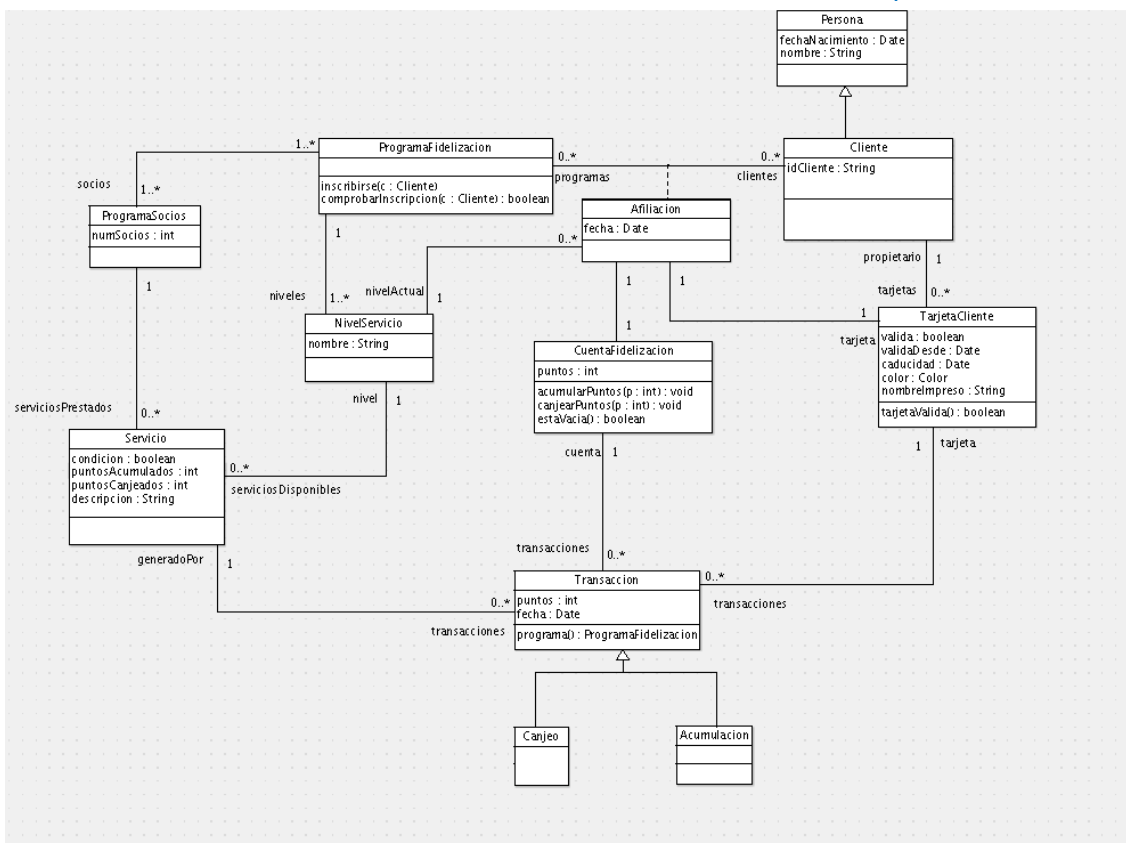
Los programas de fidelización pueden permitir a los clientes ahorrar puntos de bonificación (clase `CuentaFidelizacion`), con los que pueden "comprar" servicios de los socios del programa.

Se emite una cuenta de fidelización por cada cliente que se adhiere a un programa de este tipo y que está representada por (**clase de asociación** `Afiliacion`). Esta clase sustituye a la asociación definida entre el programa y el cliente. De esta forma se pueden añadir más propiedades a la afiliación¹. Por eso `Afiliacion` tendrá asociado el programa de fidelización y el cliente. Y en las clases `ProgramaFidelizacion` estarán asociadas todas las posibles afiliación que implementan de forma indirecta la relación con el Cliente y la clase `Cliente` tendrá asociadas todas las afiliaciones que implementa la relación con los programas de forma indirecta.

Las transacciones (clase `Transaccion`) en las cuentas de fidelización implican varios servicios proporcionados por los socios del programa y se realizan por una sola tarjeta. Hay dos tipos de transacciones: `Acumulacion` y `Canjeo`. La duración de la afiliación determina varios niveles de servicios (clase `NivelServicio`).

Un aspecto importante, es como implementar el método `programa()` de la clase `Transaccion`. Aunque esta clase no cuenta con un enlace directo con la clase `ProgramaFidelizacion`, este se puede obtener navegando por las asociaciones `cuenta` y la que esta tiene con `Afiliacion`

¹ En lo ejemplos que se os han planteado en la actividad, no es necesario que intentéis reconocer una clases de este tipo pero es interesante conocer su utilidad.



Algunas clases especificadas²

```
public class ProgramaFidelizacion {
```

```

/**
 *
 * @element-type ProgramaSocios
 */
private Vector socios;
/**
 *
 * @element-type NivelServicio
 */
private Vector niveles;
/**
 *
 * @element-type Afiliacion
 */
private Vector clientes;

public void inscribirse(Cliente c) {
}

public boolean comprobarInscripcion(Cliente c) {
return false;
}
}

```

² No se hacen referencia a los constructores porque es irrelevante ya que la asociación de objetos no produce dependencia vital

- socios se refiere a la asociación con cardinalidad múltiple con la clase ProgramaSocios, un programa de fidelización puede ser ofrecido a distintas empresas asociadas.
- niveles se refiere a la asociación con cardinalidad múltiple con la clase NivelServicio, cada programa puede tener diferentes niveles de servicio.
- clientes se refiere a la asociación con cardinalidad múltiple con la clase Afiliacion, esta implementa de forma indirecta la relación con los clientes afiliados a través de la clase Afiliacion.

```
public class Cliente extends Persona {  
  
    private String idCliente;  
  
    /**  
     *  
     * @element-type TarjetaCliente  
     */  
    private Vector  tarjetas;  
    /**  
     *  
     * @element-type Afiliacion  
     */  
    private Vector  programas;  
}
```

- tarjetas se refiere a la asociación con cardinalidad múltiple con la clase TarjetaCliente
- programas se refiere a la asociación con cardinalidad múltiple con la clase Afiliacion que indirectamente asocia al cliente los programas a los que está afiliado

```
public class Afiliacion {  
  
    private Date fecha;  
    /**  
     *  
     * los siguientes atributos son para modelar la asociacion a la que  
sustituye  
     */  
    private Cliente elcliente;  
    private ProgramaFidelizacion elprograma  
    private TarjetaCliente tarjeta;  
    /**  
     *  
     *  
     */  
    private NivelServicio nivelActual;  
    /**  
     *  
     *  
     */  
    private CuentaFidelizacion cuenta;  
  
    public void inscribirse(Cliente c) {  
    }  
  
    public boolean comprobarInscripcion(Cliente c) {  
    return false;  
    }  
}
```

- `elcliente` se refiere a la asociación con el `Cliente` afiliado que es único para cada afiliación. `elprograma` se refiere a la asociación con el `ProgramaFidelizacion` al que pertenece la afiliación. Estos dos atributos se necesitan para completar la asociación que había entre programa y cliente
- `tarjeta` se refiere a la asociación con `TarjetaCliente`, cada afiliación tiene una tarjeta.
- `nivelActual` se refiere a la asociación con `NivelServicio`, cada afiliación tiene un nivel de servicio según el momento.
- `cuenta` se refiere a la asociación con `CuentaFidelizacion`, cada afiliación esta asociada con una única cuenta.

```
public class CuentaFidelizacion {  
  
    private int puntos;  
  
    /**  
     *  
     * @element-type Transaccion  
     */  
    private Vector transacciones;  
  
    public Afiliacion laAfiliacion;  
  
    public void acumularPuntos(int p) {  
    }  
  
    public void canjearPuntos(int p) {  
    }  
  
    public boolean estaVacía() {  
        return false;  
    }  
}
```

- `transacciones` se refiere a la asociación con cardinalidad múltiple con la clase `Transaccion`
- `laAfiliacion` se refiere a la asociación con la clase `Afiliacion`, cada `CuentaFidelizacion` solo se puede asociar a una afiliación, es decir, a un programa

```
public class NivelServicio {  
  
    private String nombre;  
  
    private ProgramaFidelizacion myProgramaFidelizacion;  
    /**  
     *  
     * @element-type Afiliacion  
     */  
    private Vector lasAfiliaciones;  
    /**  
     *  
     * @element-type Servicio  
     */  
    private Vector serviciosDisponibles;  
}
```

- `lasAfiliacioness` se refiere a la asociación con cardinalidad múltiple con la clase `Afiliacion`, un nivel de servicio puede estar vigente en varias afiliaciones. Los niveles de servicio se definen para cada programa.
- `serviciosDisponibles` se refiere a la asociación múltiple con la clase `Servicio`.