

## **Documento Explicativo**

### **Seminario de Taller de lenguajes .NET 2024**

#### **Integrantes:**

- La Porta Franco (02307/2)
- Corbelleri Patricio Jose (02123/5)

# **Funcionalidad SGE.Consola/Program.cs**

## **1.Introducción**

El programa Program.cs está separado por bloques que muestran todo el resultado y sirven para testear cada parte del mismo. Si se quiere probar una parte en concreto y no todo el bloque, la mejor opción es que se pueda agregar o sacar la barra lateral (/) para que esa porción del código esté completamente comentada.

Ejemplo:

- ```
/******  
Bloque de código a ejecutar  
*****
```
- ```
/******  
Bloque de código comentado  
*****
```

## **2. Explicacion delCodigo**

### **2.0 Uso de proyectos SGE.Aplicación y SGE.Repositorios**

Utilizamos la librería de clases SGE.Aplicación donde se encuentran todas las funcionalidades de los Servicios, Interfaces, Casos de Uso, Excepciones, Validaciones y Entidades previamente programadas. Para hacer referencia a este proyecto referenciado de librería de clases utilizaremos la directiva 'using' (línea 1).

*using SGE.Aplicación;*

También utilizamos la librería de clases SGE.Repositorios donde se encuentra el manejo de repositorios de Expedientes y Trámites. Para hacer referencia a este proyecto referenciado de librería de clases utilizaremos la directiva 'using' (línea 2).

*using SGE.Repositorios;*

### **2.1 Instanciaciones**

Al inicio del programa primero se instancian los servicios, excepciones, repositorios y casos de uso (desde líneas 4 a 35) utilizando la metodología de Patrón de Inversión de Dependencias (DIP).

- Instanciación de Servicio de autorización provisorio (línea 7)
- Instanciación de Especificación de cambio de estado de los expedientes (línea 10)
- Instanciación de Repositorios de expedientes y de trámites (línea 13 y 14)
- Instanciación de Excepciones de validación de trámites y expedientes (línea 17)

- Instanciación de Excepciones de los repositorios (línea 18)
- Instanciación de todos los Casos de uso (líneas 20 a 34)

## **2.2 Verificación de Permisos de Usuario**

Hay un pequeño testeo que sirve para ver el funcionamiento del sistema de permisos de usuario y su eventual Excepción en caso de ser usuario inválido (desde líneas 37 a 41).

## **2.3 Alta Expedientes Válidos y no válidos**

Luego de esta parte tenemos la parte de dar de alta los expedientes, en la que se da de alta algunos expedientes y a su vez se intenta dar de alta expedientes que no pasan las validaciones y esto permite que se visualice los mensajes de las excepciones (desde líneas 43 a 80). Conjuntamente (desde líneas 82 a 88) hay una porción de código que permite imprimir un listado de los expedientes y ver cuáles se pudieron dar de alta y cuáles no.

Ejemplo de alta de un expediente válido:

```
// Expediente Valido
altaExpediente.Ejecutar(
    new Expediente(){
        Caratula = "caratula expediente",
        ExpedienteEstado = (EstadoExpediente) (new Random().Next() % 5)
    },
    1 // id usuario
);
```

En `altaExpediente.Ejecutar()` es necesario ingresar un expediente y un id usuario con los permisos requeridos (en este caso provisoriamente pondremos al usuario 1).

Solo agregando los datos de la carátula y el estado son necesarios ya que el resto de atributos se agregan en el caso de uso.

En este ejemplo se agrega un Estado de expediente aleatorio.

## **2.3 Alta Trámites Válidos y no válidos**

Siguiendo con la parte de dar de alta, ahora viene la parte de los trámites (desde líneas 90 a 150) funciona exactamente igual que con los expedientes, tiene trámites que están hechos para que salten las excepciones y se pueden ver las validaciones pertinentes, y como pasa también con los expedientes, hay una porción (desde líneas 152 a 159) que sirve para imprimir todo los trámites y poder visualizarlos.

*Hay que tener en cuenta que al ingresar un trámite válido, éste ocasionará un cambio de estado en su Expediente asociado según el Tipo de Trámite del mismo.*

## 2.3 Bajas

Continuamos con la parte de eliminación, la cual tiene de ambas entidades (para tramites desde líneas 161 a 173) (para expedientes desde líneas 177 a 187), se eliminan varios casos para verlos, cabe aclarar que en el caso de eliminar expediente se puede notar como también se borran los trámites asociados al expediente y para notar estos últimos cambios hay otra sección de impresión en consola de ambas entidades (desde líneas 189 a 195 para expedientes) (desde líneas 197 a 203 para trámites).

## 2.4 Modificaciones

Proseguimos con la sección de modificación de trámites (desde líneas 205 a 223) esta sección se encarga de cambiar atributos de los trámites ya creados, luego tenemos una pequeña visualización para ver los cambios ocurridos (desde líneas 225 a 231). Mismo caso para los expedientes (desde líneas 233 a 269) y su visualización en posterior al cambio para verlos en consola (desde líneas 271 a 277).

Ejemplo de modificación de Trámite:

```
// Id usuario sin permiso
modTramite.Ejecutar(
    new Tramite(){
        IdTramite = 5, // tramite a modificar
        IdExpediente = 3,
        TipoTramite = EtiquetaTramite.PaseAEstudio,
        Contenido = "sssss"
    },
    0
);
```

Esta es una modificación a algunos atributos del trámite con el id = 5

Ejemplo de modificación de Expediente:

```
// Expediente no encontrado
modExpediente.Ejecutar(
    new Expediente(){
        IdExpediente = 25, // expediente a modificar
        ExpedienteEstado = EstadoExpediente.EnNotificacion,
        Caratula = "sssss"
    },
    1
);
```

Esta es una modificación a algunos atributos de manera similar a con los trámites, pero del expediente en este caso con el ID = 25

## **2.5 Consulta**

Implementamos la consulta de un trámite haciendo que retorne un trámite nulo en caso de no encontrarlo o que retorne el trámite en caso de si encontrarlo (línea 218). Lo mismo para la consulta de un expediente (línea 255 y 262).

Ejemplo de consulta y modificación de Trámite:

```
// Consulta y modifica el expediente con id 3 (si lo encuentra)
Tramite? tramiteAux = consTramite.Ejecutar(3);
if (tramiteAux != null) {
    tramiteAux.TipoTramite = EtiquetaTramite.Notificacion;
    modTramite.Ejecutar( (Tramite) tramiteAux, 1);
}
```

En este ejemplo se hace una consulta de Trámite con id del trámite = 3, hace una búsqueda en su repositorio correspondiente y si no lo encuentra retorna un trámite nulo. En caso contrario hace una modificación en ese trámite (en el ejemplo cambia su Etiqueta a 'Notificación') y luego hace una modificación en el repositorio actualizando ese mismo trámite.

Ejemplo de consulta y modificación de Expediente:

```
// Consulta y modifica el expediente con id 1
expedienteAux = consExpediente.Ejecutar(1);
if (expedienteAux != null) {
    expedienteAux.Caratula = "hola";
    modExpediente.Ejecutar( (Expediente) expedienteAux, 1);
}
```

En este ejemplo se hace una consulta de Expediente con id del trámite = 1, hace una búsqueda en su repositorio correspondiente y si no lo encuentra retorna un trámite nulo. En caso contrario hace una modificación en ese expediente (en el ejemplo cambia su carátula a "hola") y luego hace una modificación en el repositorio actualizando ese mismo expediente.

## **2.6 Trámites asociados a un expediente**

Continuamos con la parte de trámites asociados a un expediente en la misma encontramos que al dar un ID de expediente, se ve en pantalla cuales son los trámites asociados al mismo (desde líneas 279 a 291).

## **2.7 Consulta de Etiqueta de trámites**

Llegando al final del Program.cs, tenemos la última parte de consulta de trámites por etiqueta, sirve para consultar todos los trámites con una misma etiqueta, hay 2 ejemplos, pero se pueden agregar más si se busca probar más casos (desde líneas 293 a 305).

### 3. Salida por Consola

Se detallan las salidas por consola de las excepciones y métodos ToString() de los trámites y expedientes ingresados en sus repositorios correspondientes, entre otras cosas.

Salida por consola al verificar si el Usuario con id=1 posee permisos (línea 39):

```
True
```

Salida por consola al verificar si el Usuario con id=0 posee permisos (línea 40):

```
Exception thrown: 'System.Exception' in SGE.Aplicacion.dll
El usuario no cuenta con los permisos adecuados para ejecutar esta accion
False
```

Vemos que a parte de retornar 'False', nos tira una excepción que nos dice que el usuario no posee los permisos adecuados.

Salida por consola de la excepción al dar de alta un Expediente con id de Usuario distinto de 1 (líneas 47 a 53):

```
Exception thrown: 'System.Exception' in SGE.Aplicacion.dll
No se puede agregar el Expediente al repositorio ya que Id de Usuario inválido.
```

Salida por consola de la excepción al dar de alta un Expediente con Carátula = null o sin especificar una Carátula (líneas 56 a 61):

```
Exception thrown: 'System.Exception' in SGE.Aplicacion.dll
No se puede agregar el Expediente al repositorio ya que la Caratula esta Vacía.
```

Salida por consola de todos los Expedientes que estan en el repositorio de expedientes expediente.txt (líneas 85 a 87):

```
***** Listado de todos los Expedientes: *****

Expediente = 0
Caratula = caratula expedietne
Estado del Expediente = ParaResolver
Fecha y hora creacion = 13/05/24 17:17
Fecha y hora modificacion = 13/05/24 17:17
Id Usuario = 1

Expediente = 2
Caratula = caratula expedietne
Estado del Expediente = Finalizado
Fecha y hora creacion = 13/05/24 18:57
Fecha y hora modificacion = 13/05/24 19:08
Id Usuario = 1
```

Salida por consola de la excepción al dar de alta un Expediente con *Contenido=null* o sin especificar una Contenido (líneas 103 a 110):

```
Exception thrown: 'System.Exception' in SGE.Aplicacion.dll
No se puede agregar el Tramite al repositorio ya que el Contenido esta Vacio.
```

Salida por consola de la excepción al dar de alta un Tramite con id de Usuario distinto de 1 (líneas 124 a 131):

```
Exception thrown: 'System.Exception' in SGE.Aplicacion.dll
No se puede agregar el Tramite al repositorio ya que Id de Usuario inválido.
```

Salida por consola de todos los Tramites que estan en el repositorio de tramites *tramites.txt* (líneas 152 a 157):

```
***** Listado de todos los Tramites: *****

Id Tramite = 4
Id Expediente Asociado = 2
Tipo de Tramite = EscritoPresentado
Contenido = contenido tramite
Fecha y hora creacion = 13/05/24 19:08
Fecha y hora modificacion = 13/05/24 19:08
Id Usuario (ultima modificacion) = 1

Id Tramite = 5
Id Expediente Asociado = 1
Tipo de Tramite = Notificacion
Contenido = contenido tramite
Fecha y hora creacion = 13/05/24 19:15
Fecha y hora modificacion = 13/05/24 19:15
Id Usuario (ultima modificacion) = 1
```

Salida por consola de todos los Trámites Asociados a algun expedientes (en este caso al Expediente con Id = 2), (líneas 282 a 284):

```
***** Todos Los Trámites Asociados al Expediente 2: *****

Id Trámite = 4
Id Expediente Asociado = 2
Tipo de Trámite = EscritoPresentado
Contenido = contenido tramite
Fecha y hora creacion = 13/05/24 19:08
Fecha y hora modificacion = 13/05/24 19:08
Id Usuario (ultima modificacion) = 1

Id Trámite = 6
Id Expediente Asociado = 2
Tipo de Trámite = Despachado
Contenido = contenido tramite
Fecha y hora creacion = 13/05/24 19:15
Fecha y hora modificacion = 13/05/24 19:15
Id Usuario (ultima modificacion) = 1
```

Salida por consola de todos los Trámites Asociados a un expedientes que no tiene trámites asociados. En este caso al Expediente con Id = 10). (línea 288 a 290):

```
***** Todos Los Trámites Asociados al Expediente 10: *****
```

Salida por consola de todo los Trámites con una etiqueta específica, en este ejemplo con etiqueta = 'Notificación' (líneas 296 a 298):

```
***** Todos Los Trámites con etiqueta 'Notificacion': *****

Id Trámite = 5
Id Expediente Asociado = 1
Tipo de Trámite = Notificacion
Contenido = contenido tramite
Fecha y hora creacion = 13/05/24 19:15
Fecha y hora modificacion = 13/05/24 19:15
Id Usuario (ultima modificacion) = 1

Id Trámite = 7
Id Expediente Asociado = 3
Tipo de Trámite = Notificacion
Contenido = contenido tramite
Fecha y hora creacion = 13/05/24 19:15
Fecha y hora modificacion = 13/05/24 19:15
Id Usuario (ultima modificacion) = 1
```