****

**Trabajo de Diploma**

**Sistema Artec**

**Índice**

[**5.3.21 Gestión de Perfiles de Usuario** 3](#_Toc9347765)

[**5.3.22 Gestión de Encriptado** 6](#_Toc9347766)

[**5.3.23 Gestión de Backup** 7](#_Toc9347767)

[**5.3.24 Gestión de Bitácora y Control de Cambios** 9](#_Toc9347768)

[**5.3.25 Gestión de Login/Logout** 11](#_Toc9347769)

[**5.3.26 Gestión de Dígitos Verificadores** 12](#_Toc9347770)

[**5.3.27 Gestión de Múltiples Idiomas** 14](#_Toc9347771)

**3.8.1 Gestión de Perfiles de Usuario**

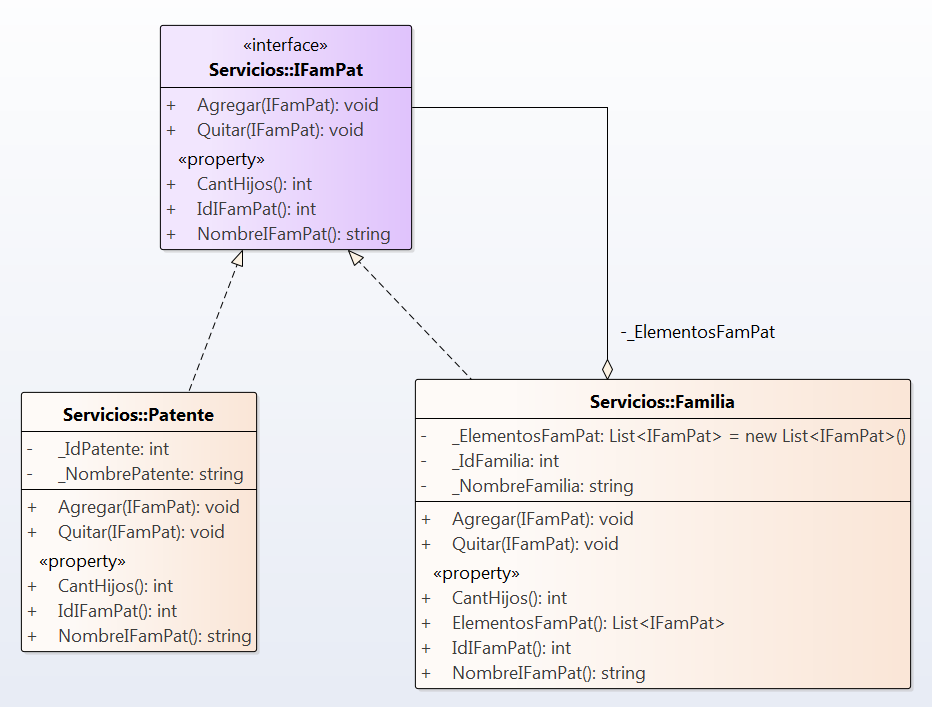
El objetivo de este aspecto técnico es brindar la posibilidad de otorgar permisos atómicos y/o compuestos, sin la necesidad de generar un nuevo rol al momento de tener que agregarle un permiso adicional a un usuario del sistema.

**Funcionamiento:**

Está realizado utilizando el patrón Composite para poder realizar la composición de objetos, en este caso Patentes y Familias, la Patente es el permiso atómico, y la Familia el compuesto, que puede contener tanto Patentes como Familias.

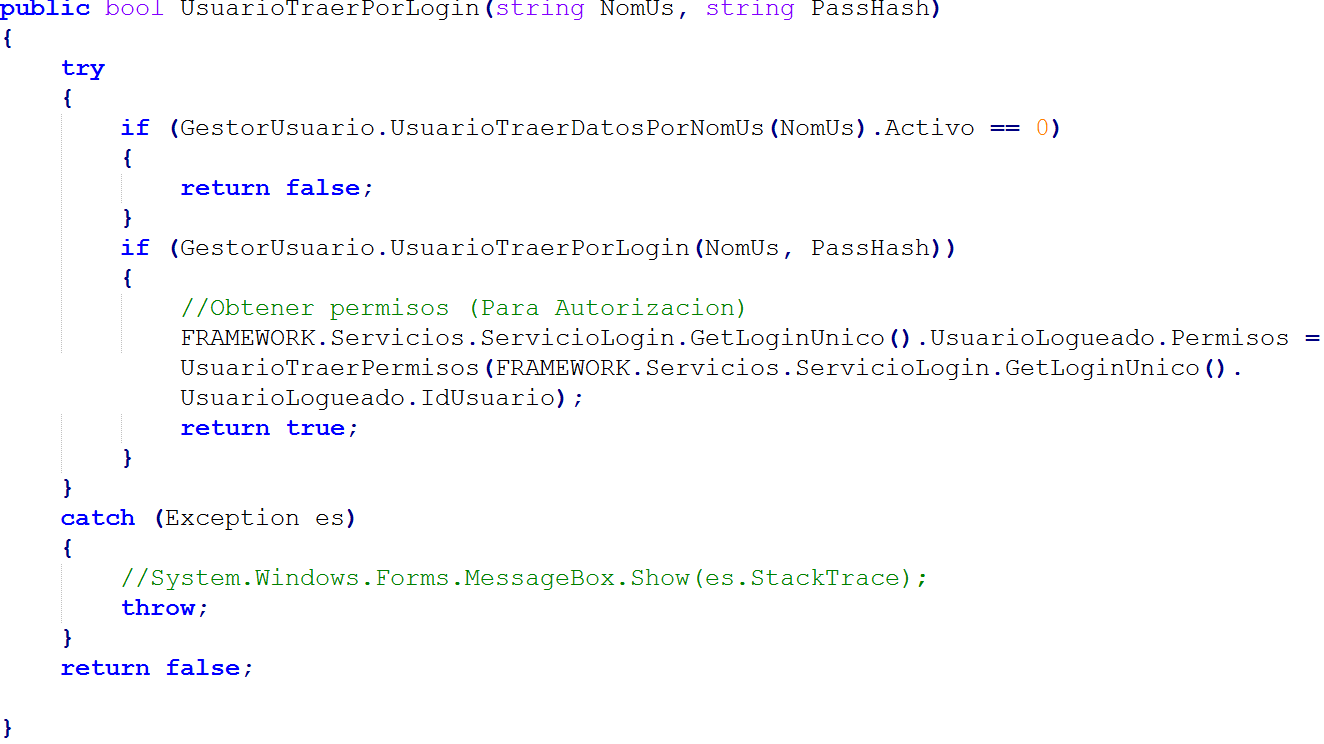
Para obtener los permisos de un usuario, a nivel de programación se realizaron funciones recursivas.

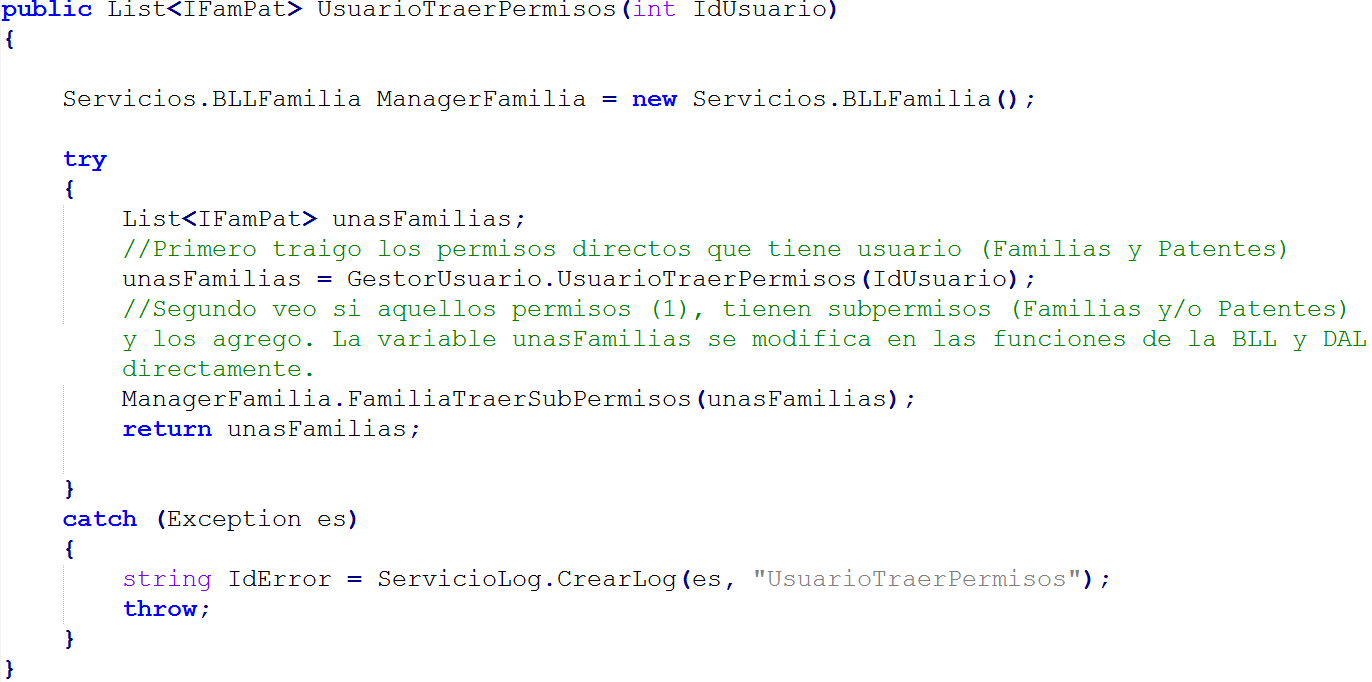
**Diagrama de Clases - Gestión de Encriptado:**



Los permisos de un usuario se consultan a la base de datos y se traen a memoria al loguearse en el sistema, posteriormente por cada formulario que se abre se realiza una inspección entre los permisos que tiene el usuario y los controles del formulario que tienen restricciones de permisos, utilizando recursividad para recorrer los subpermisos cuando se trata de Familias, en caso de tener permisos para el control determinado se lo habilita en tiempo de ejecución.

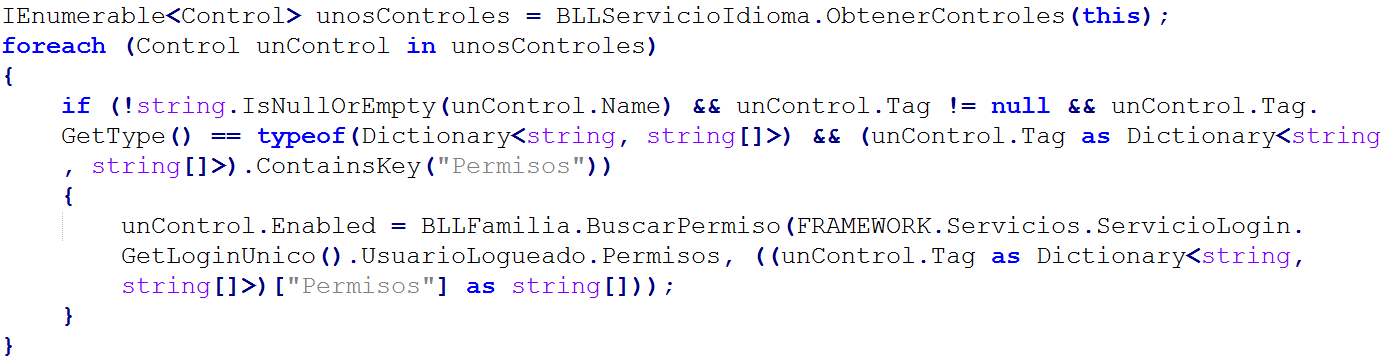
**Código que se ejecuta ante un logueo del usuario exitoso:**





Para identificar qué control debe mostrarse de acuerdo al permiso que posee el usuario logueado, se utilizó la propiedad “tag” de los controles de formulario .Net.

Debido al uso de dicha propiedad por el aspecto técnico “Múltiples Idiomas”, se conformó un diccionario para contener los valores para los permisos y los textos traducidos en la misma propiedad, y así desacoplar también los controles con respecto a los permisos.

****

**DER - Gestión de Perfiles de Usuario:**



**3.8.2 Gestión de Encriptado**

El objetivo de este aspecto técnico comprende la securización de los datos sensibles que el sistema puede llegar a persistir, obtener y transmitir.

El módulo de securización está compuesto por 3 funciones:

* AplicarHash:

A partir de la recepción de un texto, genera y retorna un código alfanumérico, desde el cual no puede obtenerse el argumento original.

* Encriptar:

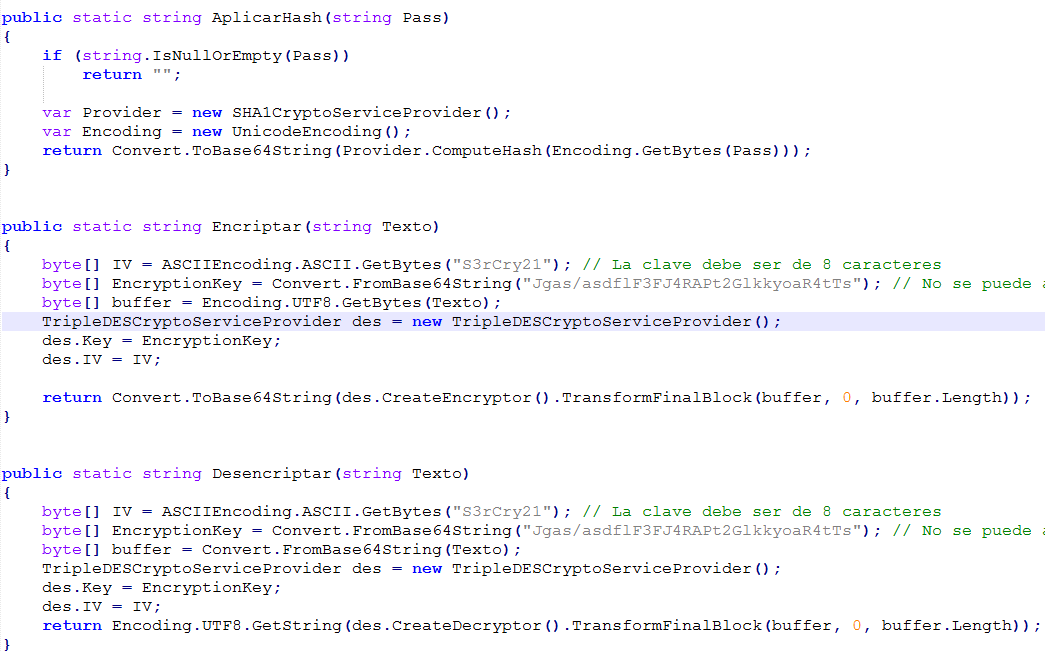
A partir de la recepción de un texto, convierte el texto en una cadena ilegible de texto, utilizando una clave y un algoritmo de encriptado.

* Desencriptar:

A partir de la recepción de un texto encriptado, retorna el texto original.

En el sistema, por ejemplo, para almacenar y obtener las contraseñas se utiliza una combinación de las funciones descriptas, brindando confidencialidad a través de la función “Encriptar”, e integridad a través de la función “AplicarHash”.

**Código fuente de las funciones del módulo de securización:**



**Diagrama de Clases - Gestión de Encriptado:**

****

**3.8.3 Gestión de Backup**

El objetivo de este aspecto técnico es permitir el resguardo de la base de datos desde la aplicación en funcionamiento y a su vez, la restauración de la misma a un estado anterior a partir de un archivo de backup realizado anteriormente.

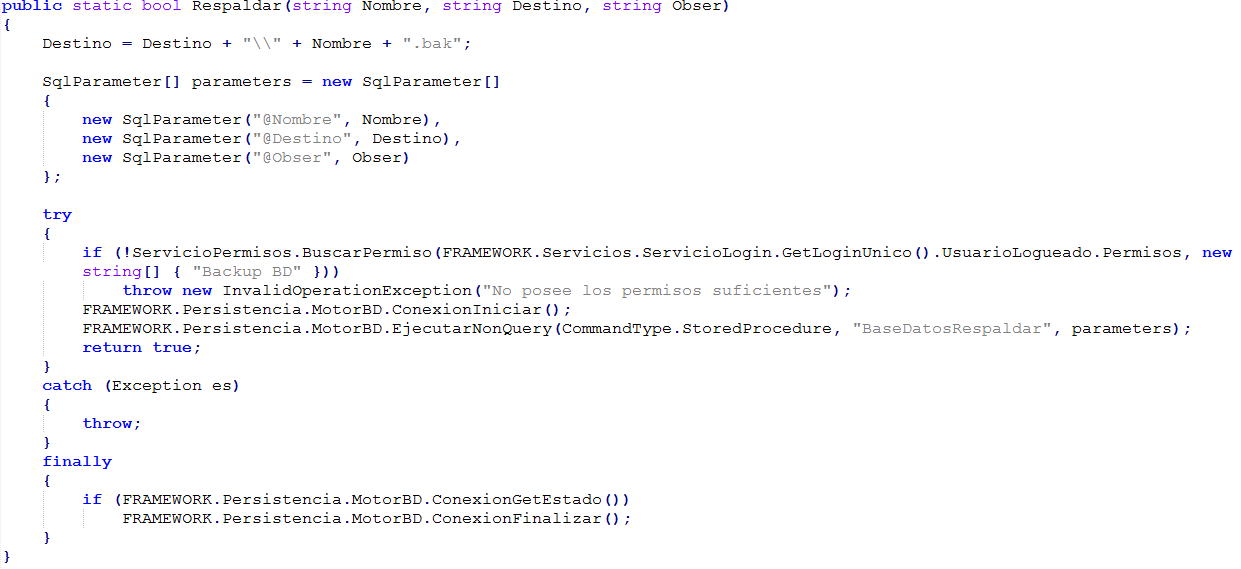
**Diagrama de Clases - Gestión de Backup:**



**Respaldar:**

La función para realizar un backup de la base de datos recibe 3 argumentos:

“Nombre” en donde se indica el nombre del archivo que se creará, “Destino” para indicar la ruta del archivo a crearse y “Observación” que permite agregar una descripción a la operación.

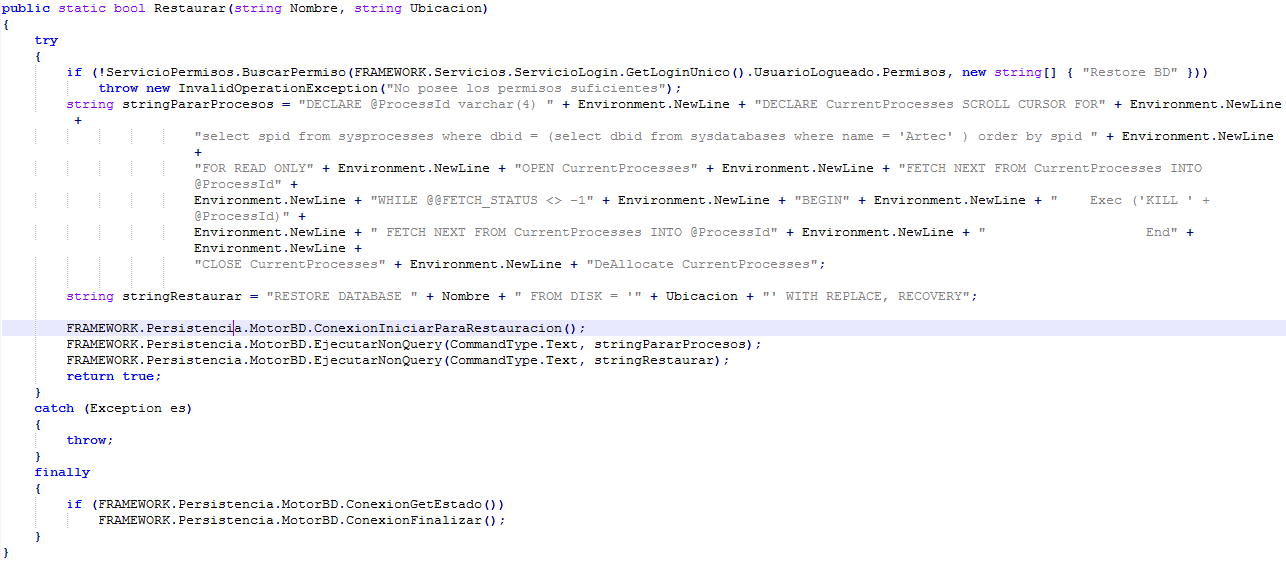


**Restaurar:**

La función para restaurar a un estado anterior la base de datos recibe el nombre de la base a restaurar y la ubicación del archivo con el backup.

En este procedimiento la aplicación se desconecta de la base de datos para permitir eliminarla y realizar la correspondiente restauración.

Una vez finalizada la restauración, la aplicación indica que se debe iniciar nuevamente la misma y se cierra.



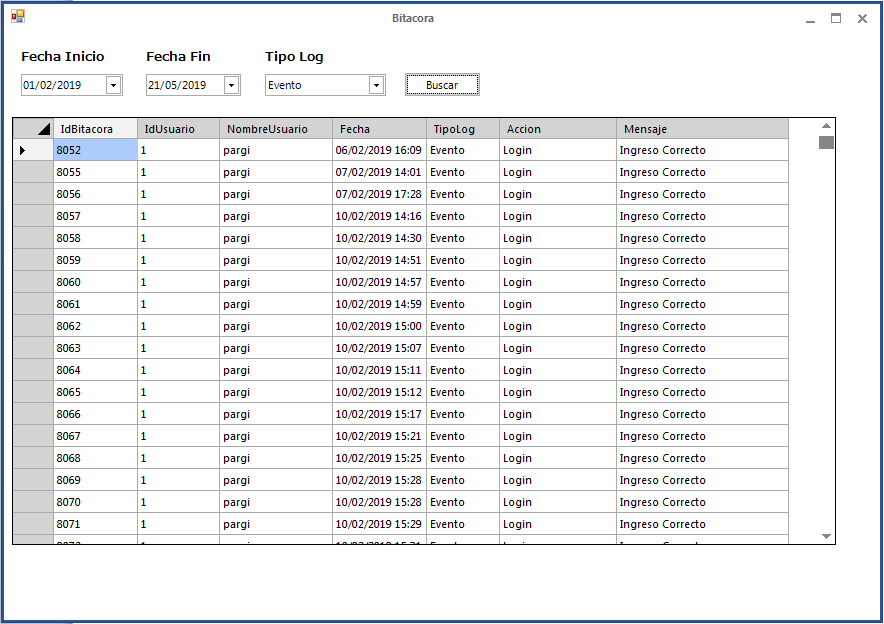
**3.8.4 Gestión de Bitácora y Control de Cambios**

El objetivo de este aspecto técnico es realizar un seguimiento de la actividad que realizan los usuarios en el sistema, como a su vez registrar los errores que acontecen durante la ejecución del mismo.

Los errores que ocurren durante la ejecución del sistema, se registran en la base de datos de la aplicación y en los eventos de Windows, tal que llegado el caso en que haya un error en la base de datos, se realice un formateo de la información brindada en la excepción y pueda visualizarse el mensaje que el sistema emitió dentro del visor de eventos de Windows.

En cambio, las acciones que los usuarios realizan en la aplicación se registran únicamente en la base de datos, ya que la utilidad de esta funcionalidad es la de permitir un control y en tal caso auditar las acciones realizadas por los usuarios.

Los registros en la base de datos de las acciones realizadas por los usuarios y los errores que ocurren en la ejecución del sistema pueden visualizarse desde la misma aplicación, permitiendo realizar filtros de eventos y errores y rango de fechas.



**Diagrama de Clases - Gestión de Bitácora y Control de Cambios:**

****

**DER - Gestión de Bitácora y Control de Cambios:**



**3.8.5 Gestión de Login/Logout**

El objetivo de este aspecto técnico es brindar un control de acceso al sistema y registrar quién ingresó.

Para ello se mantiene una base de usuarios que utilizarán el sistema cada uno con su propia contraseña de acceso.

El procedimiento de Login se realiza de la siguiente manera:

* + - 1. El usuario ingresa su nombre de usuario y contraseña
      2. Se verifica que el nombre de usuario ingresado exista y esté activo
      3. En caso positivo se comprueba si la contraseña ingresada es correcta
      4. En caso positivo se obtienen los datos del usuario (Nombre, Apellido) y se alojan los mismos en memoria a través de una clase utilizada para mantener los datos de la sesión, conformada por un patrón Singleton, a efectos de asegurar la instanciación única de la clase. Por lo que a partir de ahí, los datos se obtienen desde el objeto “ServicioLogin” instanciado que contiene una variable de tipo “Usuario”.
      5. Se realiza la autorización obteniendo los permisos de acceso
      6. Se registra en la bitácora el ingreso del usuario

El procedimiento de Logout se realiza de la siguiente manera:

1. El usuario cierra la aplicación
2. Se registra en la bitácora la salida del usuario de la aplicación

**Diagrama de Clases - Gestión de Login/Logout:**

****

**3.8.6 Gestión de Dígitos Verificadores**

El objetivo de este aspecto técnico es realizar una comprobación de la integridad de la base de datos, tanto si se agregaron o quitaron registros o se modificó el orden de los mismos.

Este aspecto está conformado por una tabla en la base de datos dedicada “DVV”, la cual contiene un hash conformado por cada uno de los hasheos realizados a los registros de una tabla determinada, hay tantos registros como tablas se considere debe asegurarse la integridad.



Por otro lado, dentro de las tablas que se considerada debe realizarse este proceso, se encuentra un campo “DVH” para obtener un número que se conforma a partir de los campos contenidos en el registro.

Cada vez que se modifica un dato en un registro de alguna de las tablas indicadas, o se agrega un nuevo registro, se realiza el proceso de cálculo y registro de DVH y DVV.

**Verificación al Inicio del Sistema**

Además, cada vez que se inicia el sistema, y antes de que aparezca la ventana de acceso al Login, se realiza la verificación de integridad de la base de datos a través de la comprobación de los dígitos verificadores DVH y DVV.

En caso de existir discrepancias entre los valores DVH y DVV almacenados en la base de datos contra los calculados en tiempo de ejecución, el sistema únicamente permite que se loguee un usuario con permisos de Administrador.

Al respecto, una vez logueado un Administrador, el sistema ofrece dos opciones:

* + - * 1. Realizar la recomposición de los dígitos verificadores (persistir y sobrescribir los valores calculados en tiempo de ejecución).
        2. Realizar la restauración de la base de datos a partir de un archivo de backup.

**Métodos de cálculo de DVH y DVV:**

**DVH:**

El método utilizado para conformar el valor DVH comprende:

1. Realizar un hash de los valores concatenados de los campos de un registro de una tabla.
2. Por cada caracter del valor hash obtenido, se realiza una sumatoria de acuerdo al valor long del carácter, hasta completar todos los caracteres.

Este proceso se realiza por cada registro que se encuentra en la tabla y por cada tabla comprendida en la verificación de integridad por DVH y DVV.

A su vez, este proceso ejecuta la actualización del DVV.

**DVV:**

El método utilizado para conformar el valor DVV comprende:

1. Se obtienen los DVH asociados a la tabla implicada.
2. Se concatenan los mismos anexándoles un “;” como separador entre cada valor DVH.
3. Se aplica un hash al valor concatenado

**Diagrama de Clases - Gestión de Dígitos Verificadores:**

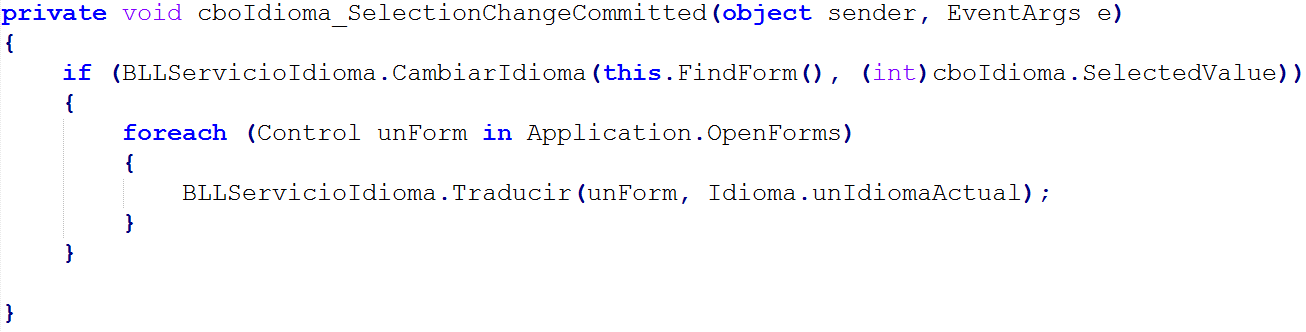


**3.8.7 Gestión de Múltiples Idiomas**

El objetivo de este aspecto técnico es permitirle al usuario modificar el idioma en tiempo de ejecución en forma dinámica, es decir, sin la necesidad de cerrar el sistema y/o las pantallas dispuestas en vista.

Para lograr el cometido de traducción de los controles de formulario de .Net, se utilizó la propiedad “tag” de los mismos, de manera de desacoplar el control con respecto al valor traducido del idioma persistido en la base de datos.

En base a la dinámica para poder traducir entre distintos idiomas, se creó una función que recibe un control de formulario y el idioma actual en que se encuentra el sistema, colocando la misma dentro de un “foreach” con la condición de que se realice para todos los formularios abiertos.



Para lograr que el sistema sea veloz para mostrar los valores traducidos de los controles, se obtienen todos los valores del idioma seleccionado y se los guarda en memoria para ir a buscarlos y comparar con la propiedad “tag” en tiempo de ejecución.

**Diagrama de Clases - Gestión de Múltiples Idiomas:**



**DER - Gestión de Múltiples Idiomas:**

