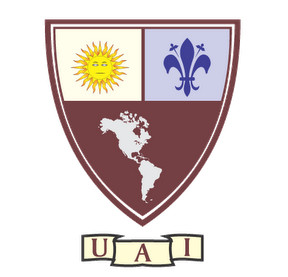
Universidad Abierta Interamericana

Facultad de Tecnología Informática

Modelos Computacionales de Gestión Administrativa



Trabajo Práctico Integrador

Docente: Milio, Roberto Claudio

Alumnos:

Dome, Martin

Rodriguez Arata, Damian

## índice

[índice 1](#_Toc118671585)

[1. Introducción 3](#_Toc118671586)

[1.1 Descripción del negocio 3](#_Toc118671587)

[2. Casos de uso 5](#_Toc118671588)

[2.1. Inicio de sesión 5](#_Toc118671589)

[2.1.1. Ciclo de vida 5](#_Toc118671590)

[2.1.2. Secuencia de navegación 6](#_Toc118671591)

[2.1.3. Código 6](#_Toc118671592)

[2.2 Copia de seguridad y restauración 16](#_Toc118671593)

[2.2.1 Ciclo de vida 17](#_Toc118671594)

[2.2.2. Secuencia de navegación 19](#_Toc118671595)

[2.2.3. Código 19](#_Toc118671596)

[2.3. Integridad del sistema 24](#_Toc118671597)

[2.3.1. Ciclo de vida 24](#_Toc118671598)

[2.3.2. Secuencia de navegación 25](#_Toc118671599)

[2.3.3. Código 27](#_Toc118671600)

[3. Estrategia de programación 37](#_Toc118671601)

[3.1. Programación en capas 37](#_Toc118671602)

[3.2. Control de acceso por URL 38](#_Toc118671603)

[3.3. Encriptación 39](#_Toc118671604)

[3.4. Frecuencia de copia de seguridad 42](#_Toc118671605)

[3.5 Tipo de conexión 42](#_Toc118671606)

[4. Árbol de navegación 42](#_Toc118671607)

# 1. Introducción

## 1.1 Descripción del negocio

**Veterinaria “Cachorros SA”,** es una empresa familiar ubicada en Burzaco, con sucursales en Quilmes y Wilde, fundada en 1996 por el doctor Carlos A. Gomez y Juan Pablo Viña, quienes actualmente siguen brindando su atención especializada en perros y gatos. El uso de internet se realiza principalmente para expandir más su mercado, dar a conocer sus servicios y captar nuevos clientes de zonas nuevas zonas geográficas.

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Forma, Rectángulo

Descripción generada automáticamente

# 2. Casos de uso

## 2.1. Inicio de sesión

## 2.1.1. Ciclo de vida

## Texto Descripción generada automáticamente con confianza baja

### 2.1.2. Secuencia de navegación

1- Usuario selecciona login en pantalla (UI)

2- El usuario ingresa los datos (UI)

3- Se verifican los datos ingresados en (BLL)

4- Encriptación de la clave de usuario y comprobación con la clave guardada y se obtienen los datos y permisos del usuario (DAL)

5- Se guardan los datos del usuario en variables de sesión (UI)

Si el usuario es cliente:

6.1.1- Se redirecciona a la página por defecto (UI)

Si el usuario es administrador

6.2.1- Se redirecciona a la página de administración (UI)

6.2.2- Se recuperan los datos de las variables de sesión para comprobar que el usuario tenga los permisos para administrador (UI)

6.2.3- Se envía consulta de bitácora (BLL)

6.2.4- Se recuperan los datos de la bitácora de la fuente de datos (DAL)

6.2.5- Se carga el objeto bitácora (UI)

### 2.1.3. Código

1. Default.aspx: Se hace redirección a login

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

2. Login.aspx.cs: Se envian los datos del usuario:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

3. Usuario.BLL: Se verifican los datos ingresados:

Texto

Descripción generada automáticamente

4. Usuario.DAL: Se encripta clave de usuario, se controla con la base y se obtienen los datos y permisos del usuario

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Bitacora\_DAL: Se llena la bitacora con la informacion del login

Texto

Descripción generada automáticamente

5.Login.aspx: Se redirecciona luego del login exitoso

Texto

Descripción generada automáticamente

6.2.1/6.2.2: Admin.aspx: Se redirecciona a Admin.aspx y se envía petición para consulta de bitácora

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

6.2.3 Bitacora\_BLL: Se envía la consulta a la capa de acceso para obtener bitácora

Texto

Descripción generada automáticamente

6.2.4 Bitacora\_DAL: Se consulta la bitácora en la fuente de datos

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

6.2.5 Admin.aspx: Se muestran los datos en la pantalla

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

## 2.2 Copia de seguridad y restauración

### 2.2.1 Ciclo de vida

Texto

Descripción generada automáticamente

### 2.2.2. Secuencia de navegación

1- Un usuario Administrador con sesión iniciada selecciona Realizar Backup (UI) (Admin)

2- Se redirecciona a página de backup (UI)

3- Usuario selecciona la opción "backup" ingresa el nombre del archivo de backup y finaliza con la selección Realizar Backup (UI)

4- Se envían los datos de realización de backup (BL)

5- La capa de acceso a datos se encarga de realizar la copia de seguridad

6- Se indica al usuario que la copia de seguridad se realizó con éxito

7- Usuario selecciona la opción "restore", nombre del archivo de backup y finaliza con la selección Realizar Restore

8- Se envían los datos de realización de restore (BL)

9- La capa de acceso a datos se encarga de realizar la copia de seguridad

10- Se indica al usuario que el restore se realizó con éxito

### 2.2.3. Código

1- Admin.aspx: Un usuario Administrador con sesión iniciada selecciona Realizar Backup

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

2- Admin.aspx: Se redirecciona a página de backup

Texto

Descripción generada automáticamente

3- BackupRestore.aspx: Usuario selecciona la opción "backup" ingresa el nombre del archivo de backup y finaliza con la selección Realizar Backup

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

4- Usuario\_BLL: Se envían los datos de realización de backup

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

5- Usuario\_DAL: La capa de acceso a datos se encarga de realizar la copia de seguridad

Texto

Descripción generada automáticamente

6- BackupRestore.aspx: Se indica al usuario que la copia de seguridad se realizó con éxito

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto, Sitio web

Descripción generada automáticamente



7- BackupRestore.aspx: Usuario selecciona la opción "restore", nombre del archivo de backup y finaliza con la selección Realizar Restore

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

8- Usuario\_BLL: Se envían los datos de realización de restore

Texto

Descripción generada automáticamente

9- Usuario\_DAL: La capa de acceso a datos se encarga de realizar la copia de seguridad

Captura de pantalla de computadora

Descripción generada automáticamente

10- BackupRestore.aspx: Se indica al usuario que el restore se realizó con éxito

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente



## 2.3. Integridad del sistema

### 2.3.1. Ciclo de vida

Texto

Descripción generada automáticamente

### 2.3.2. Secuencia de navegación

1. Inicia en el método Page\_Load comprobando la integridad digito verificador horizontal y luego hace el del vertical. Si hay tablas con errores se guarda registro de la misma tanto para horizontal como vertical, estos datos luego se mostrarán por pantalla. (UI)

2. Se obtienen las tablas con errores de verificación DVH y las DVV. (Integridad\_BLL.cs).

3. Se recalculan los valores de los DVH (paso 6) de cada registro (paso 5) de tablas que tienen dígitos verificadores (paso 4). En caso de encontrarse un registro corrompido, se agrega a una lista de registros para luego mostrar en pantalla. (Integridad\_DAL.cs)

4. Se obtienen desde la base de datos todas las tablas que tienen dígitos verificadores. Se guardan en objetos del tipo DigitoVerificador\_BE tanto el nombre de la tabla como el DVV (se usa en la verificación del DVV, no del DVH). (Integridad\_DAL.cs)

5. Se leen todos los registros de la tabla enviada como parámetro y se guardan como una lista de objetos Registro\_BE. Cada Registro\_BE guarda el id del registro (para luego mostrar por pantalla si el registro está comprometido), un string tiene concatenados todos los campos del registro con el DVH separado por el carácter ‘;’, y el nombre de la tabla. (Integridad\_DAL.cs)

6. Se calcula el valor del DVH del registro a partir de la codificación ASCII de todos los campos del registro concatenados, multiplicando este valor por el peso que tienen en la cadena. Esta función se utiliza tanto para guardar nuevos DVH como para el recálculo en la verificación de integridad. (Integridad\_DAL.cs)

7. Se realiza la verificación de integridad de la base de datos respecto al dígito verificador vertical. (UI)

8. 8.Se obtienen las tablas con errores de digito verificador vertical. (Integridad\_BLL.cs)

9. Se recalculan los valores de los DVV (paso 11) de cada tabla que tiene dígito verificador (paso 10). En caso de encontrarse una tabla corrompida, se agrega a una lista de registros (cuyo id es DVV) para luego mostrar en pantalla. (Integridad\_DAL.cs)

10. Se obtienen desde la base de datos todas las tablas que tienen dígitos verificadores. Se guardan en objetos del tipo DigitoVerificador\_BE tanto el nombre de la tabla como el DVV (se usa en la verificación del DVV, no del DVH). (Integridad\_DAL.cs)

11. Se calcula el DVV para la tabla como la suma total de todos los DVH de cada registro de la tabla. (Integridad\_DAL.cs)

12. En caso de encontrarse errores con errores de DVH, DVV o ambos, se redirige a la página de error de integridad (UI).

13. Finalmente se imprime por pantalla una lista con las tablas comprometidas y los id de los registros que fallaron en la comprobación de DVH. En caso de error de DVV, se imprime ‘DVV’ en lugar del ID del registro (UI)

### 2.3.3. Código

**1. SiteMaster.aspx**

1.Se realiza la verificacion de integridad de la base de datos respecto al digito verificador horizontal

7.Se realiza la verificacion de integridad de la base de datos respecto al dígito verificador vertical

12.En caso de encontrarse errores con errores de DVH, DVV o ambos, se redirige a la página de error de integridad

Texto

Descripción generada automáticamente

**Integridad\_BLL.cs**

2.Se obtienen las tablas con errores de digito verificador horizontal

8.Se obtienen las tablas con errores de digito verificador vertical

Texto

Descripción generada automáticamente

**Integridad\_DAL.cs**

3.Se recalculan los valores de los DVH (paso 6) de cada registro (paso 5) de tablas que tienen dígitos verificadores (paso 4). En caso de encontrarse un registro corrompido, se agrega a una lista de registros para luego mostrar en pantalla.

4.Se obtienen desde la base de datos todas las tablas que tienen dígitos verificadores. Se guardan en objetos del tipo DigitoVerificador\_BE tanto el nombre de la tabla como el DVV (se usa en la verificación del DVV, no del DVH)

5.Se leen todos los registros de la tabla enviada como parámetro y se guardan como una lista de objetos Registro\_BE. Cada Registro\_BE guarda el id del registro (para luego mostrar por pantalla si el registro está comprometido), un string tiene concatenados todos los campos del registro con el DVH separado por el carácter ‘;’, y el nombre de la tabla.

6.Se calcula el valor del DVH del registro a partir de la codificación ASCII de todos los campos del registro concatenados, multiplicando este valor por el peso que tienen en la cadena. Esta función se utiliza tanto para guardar nuevos DVH como para el recálculo en la verificación de integridad.

9.Se recalculan los valores de los DVV (paso 11) de cada tabla que tiene dígito verificador (paso 10). En caso de encontrarse una tabla corrompida, se agrega a una lista de registros (cuyo id es DVV) para luego mostrar en pantalla.

10.Se obtienen desde la base de datos todas las tablas que tienen dígitos verificadores. Se guardan en objetos del tipo DigitoVerificador\_BE tanto el nombre de la tabla como el DVV (se usa en la verificación del DVV, no del DVH)

11.Se calcula el DVV para la tabla como la suma total de todos los DVH de cada registro de la tabla.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

# 3. Estrategia de programación

## 3.1. Programación en capas

Para la programación de la pagina web se decidió utilizar una arquitectura dividida en 4 capas:

* La capa de presentación
* La capa Business Logic Layer (BLL) o Capa de lógica de negocio
* La capa Data Access Layer (DAL) o capa de acceso a datos
* La capa Business Entity (BE) o Entidades de negocio

La capa presentación contiene toda la funcionalidad gráfica, por ejemplo, las páginas web, y la principal función es validar los ingresos del usuario y formatear las salidas de resultados del sistema. En esta capa no se realizan cálculos ni operaciones, tampoco se accede a ningún repositorio de datos. No proveer servicios a ninguna capa, y recibe servicios de las capas Business Entity y Business Logic Layer

La capa Business Entity (BE) separa el comportamiento de los objetos y construye una capa transversal. Las clases Business Entity solo contienen atributos. La capa Business Entity proveen su servicio a todas las demás capas de la arquitectura, que utilizan las entidades dentro de sus mensajes.

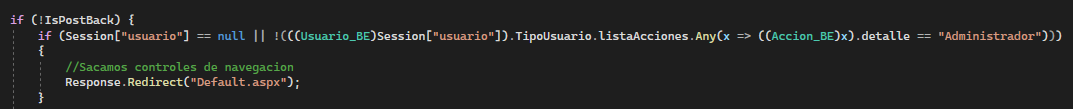
La capa Business Logic Layer queda con las clases con comportamiento, realiza todas las operaciones del negocio, no almacena ni lee datos de ningún repositorio. La capa de negocio es la responsable de realizar toda la funcionalidad propia del sistema. Brinda servicios a la capa de presentación, y utiliza los servicios de las capas Business Entity y Data Access Layer.

La capa Data Access Layer o capa de acceso a datos es la encargada de leer y grabar información en el repositorio de datos correspondientes. Esta dividida en 2 partes: La primera tiene todos los elementos necesarios para la comunicación con la base de datos, y otra que representa un mapper o un elemento encargado de traducir entidades de negocio en registros de base de datos y viceversa.

## 3.2. Control de acceso por URL

Se realiza un control dentro de la lógica de la capa de presentación para que los usuarios sin permisos o sin sesión iniciada puedan acceder paginas a las que no tienen acceso.

En el método de carga de página Page\_Load () de cada uno de los Web Forms que conforman a la aplicación, una sección de código verifica que (según lo necesite la página) el usuario tenga sesión iniciado o tenga permisos para accederla. En caso contrario, se realiza una redirección automática a la página Default.aspx del sitio.



*Ejemplo de sección de código de control de acceso por URL en Admin.aspx*

## 3.3. Encriptación

Se decidió que las claves de los usuarios sean encriptadas con el método de encriptación irreversible MD5.

Para eso, las claves de los usuarios son encriptadas utilizando el algoritmo de encriptación MD5 al momento de la registración, y son almacenadas de esta manera en la base de datos.

Al momento del inicio de sesión, la clave ingresada por el usuario es encriptada utilizando el mismo algoritmo de encriptación MD5, se obtiene la clave encriptada de la base de datos para ese usuario y ambos hashes son comparadas para verificar la identidad.

Texto

Descripción generada automáticamente

*Clave encriptada es almacenada en la base de datos*

*Texto

Descripción generada automáticamente*

*Algoritmo de generación de hash MD5 utilizado.*

Captura de pantalla de computadora

Descripción generada automáticamente

*Stored procedure que utiliza los hashes para verificar el usuario*

Texto

Descripción generada automáticamente

*Método que encripta la clave indicada en inicio de sesión y la envía al stored procedure*

## 3.4. Frecuencia de copia de seguridad

Las copias de seguridad son realizadas por un usuario con permisos de administrador de manera manual.

La periodicidad de realizar las copias de seguridad que a discreción de los administradores del sistema.

Todas las copias de seguridad y restauraciones del sistema se realizan de manera completa, es decir, que no hay división en los archivos de seguridad y la base de datos se respalda en su totalidad.

## 3.5 Tipo de conexión

Para la conexión a la base de datos, se decidió utilizar el modelo de conexión Conectado a una base de datos SQL Server.

Para esto, se resolvió que todas las consultas a base de datos (sea de lectura o escritura) sean escritas en stored procedures.

Utilizando las clases del modelo de conexión Conectado SqlCommand y SqlParammeter podemos hacer eso de los stored procedures almacenados en la base de datos. De esta manera, se protege al sitio de ataques del tipo SQL Injection.

# 4. Árbol de navegación

Diagrama

Descripción generada automáticamente