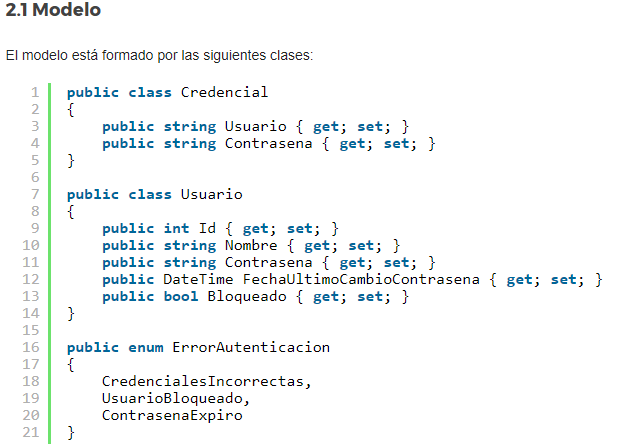
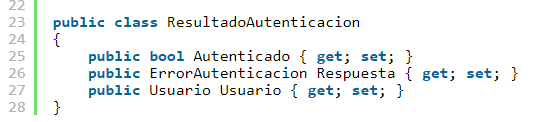
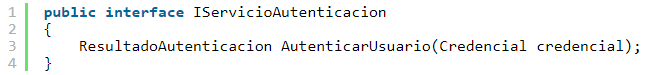
1. **Ejemplo de arquitectura cebolla**



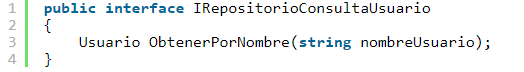


**2.2 Interfaz del servicio de autenticación**

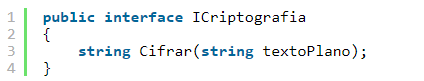
Este servicio tiene como única responsabilidad autenticar a los usuarios del sistema (recuerden el principio Single Responsability).



**2.3 Interfaz de Repositorio de usuarios**

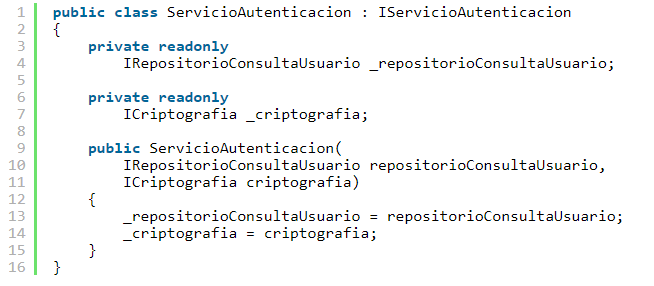


**2.4 Interfaz de Criptografía**

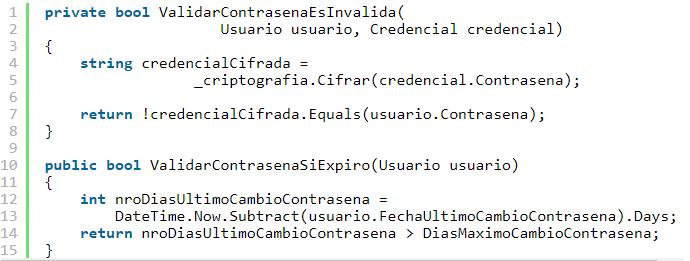


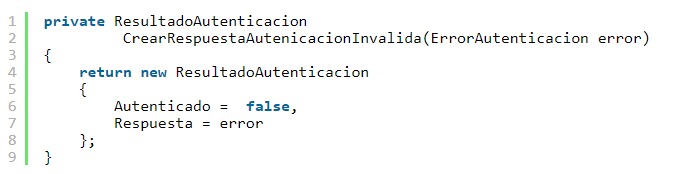
**2.5 Implementación del servicio de autenticación**

Esta clase recibe las dependencias externas usando la inyección de dependencias (dependency Injection) por constructor.

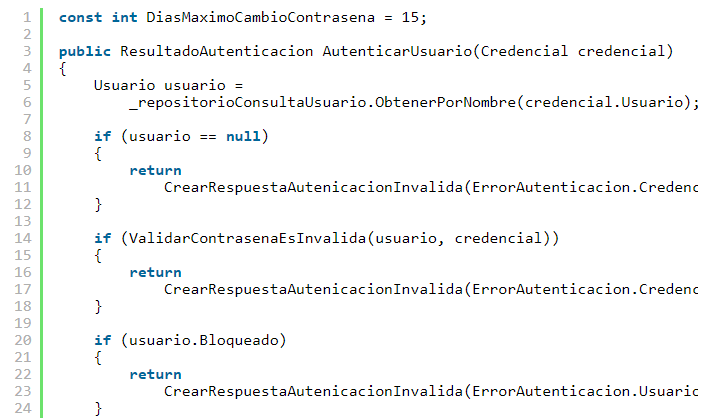


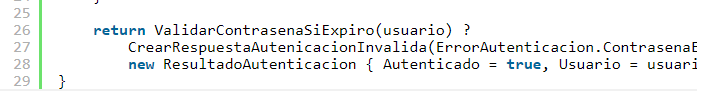
**2.5.1 Métodos de validación**



**2.5.2 Crear respuesta de validación**

**2.5.3 Método Autenticar**



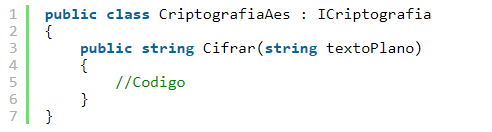


**3. Diseño de elementos de infraestructura**

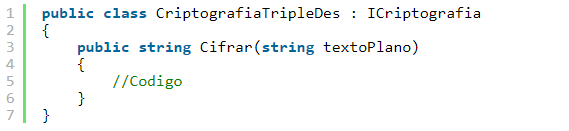
Una vez detectada las dependencias que necesita el core pasamos a definir estas implementaciones.

**3.1 ICriptografia**

El requerimiento pide que se cifren las contraseñas usando el algoritmo AES.

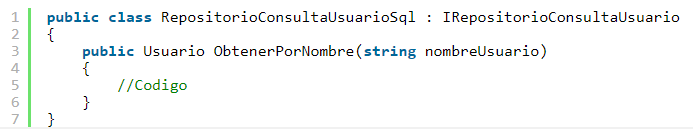


También requiere que soporte el uso del algoritmo TripleDes.



Para dar solución al tema de intercambio de algoritmos de cifrado se usa el patrón Strategy y el principio Open Closed. Aplicamos el patrón Strategy porque permite hacer cambio entre los dos algoritmos requeridos, ya que son de la misma familia. También aplicamos el principio Open Closed porque cada vez que se quiera agregar un nuevo algoritmo de cifrado no se tiene que tocar las clases existentes sino crear una nueva.

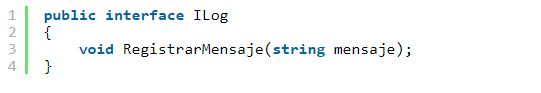
**3.2 IRepositorioConsultaUsuario**



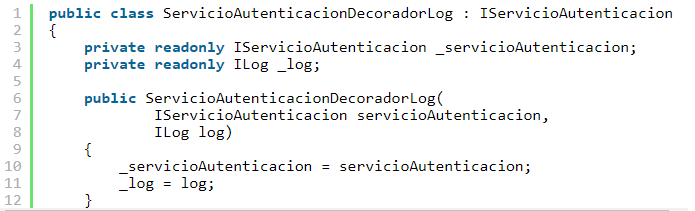
**4. Diseño Servicio autenticación decorador Log**

Para dividir las responsabilidades entre la lógica de la autenticación y el registro del log se decidió implementar el patrón Decorator. Esta nueva clase decora con la funcionalidad del registro de log a la clase ServicioAutenticacion.

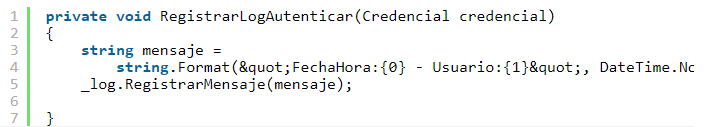
**4.1 Interfaz Log de mensajes**



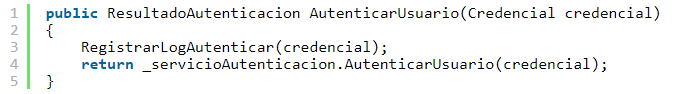
**4.2 Implementación del decorador**



**4.2.1 Método registro de mensaje**

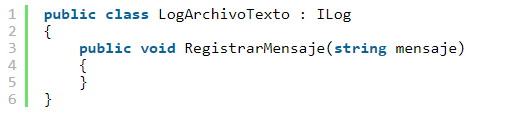


**4.2.2 Método Autenticar**

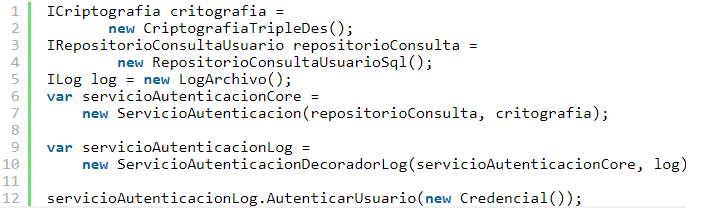


**5. Llamada desde el cliente**

**5.1 ILog con archivo de texto**

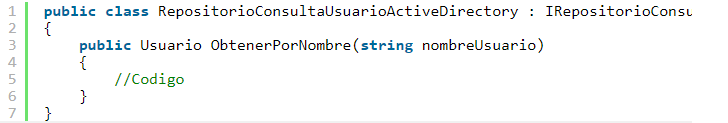


**5.2 Autenticación contra Sql**



**5.3 IRepositorioConsultaUsuario con Active directory**

Ahora se necesita buscar a los usuarios en el Active directory.



**5.4 Autenticación contra el ActiveDirectory**

Para cambiar el tipo de repositorio solo debemos pasar como referencia la clase RepositorioConsultaUsuarioActiveDirectory. Al intercambiar objetos de la misma clase base sin alterar el funcionamiento del código se aplicando el principio Liskov Substitution.

