Criptografía

Los componentes de criptografía dentro del FWK se encuentran en el siguiente namespace

Fwk.Security.Cryptography

SymetricCypherFactory Es un **e**ncriptador simétrico estático que utiliza el algoritmo RijndaelManaged

Requiere un previa configuración en el archivo .config: La configuración está basada en providers y se pueden agregar tantos encriptadores como sean necesarios. Cada uno tendrá una clave de encriptación diferente

```
<configSections>
    <section name="FwkCypherProvider"
type="Fwk.Security.Cryptography.Config.CypherProviderSection, Fwk.Bases"/>

</configSections>

<FwkCypherProvider defaultProviderName="ENCRIPTADOR_1">
    <Providers>
      <add name="ENCRIPTADOR_1" type="file" source="seed.k"/>
      <!--<add name="ENCRIPTADOR_2" type="file" source="key3.txt"  />-->
    </Providers>
  </FwkCypherProvider>
```

Lo importante aquí es que cada encriptador disponga de una semilla bien formateada source="seed.k" . Donde source="seed.k" es ruta y nombre completo del archivo de encriptación.

Para generar una nueva semilla de encriptación se puede utilizar el método **GenNewKey** de la clase SymetricCypherFactory

String seed = SymetricCypherFactory.GenNewKey ();

El resultado de seed podría quedar como el siguiente string:

KJVzHoMkFCWQCEsHaUbjPzT8kUGFRh6e2gQJC+Vtw+s=$P6ydBMk84v+lTBOd/3wtzw==

Y posteriormente almacenarla en un archivo Ej: seed.k.

Nota : Mas adelante también veremos que dentro del FWK existen herramientas generadoras de nuevas semillas

Como usarlo:

Una vez teniendo todo configurado simplemente llamar a:

Para encriptar

```
txtEncriptedText.Text = SymetricCypherFactory.Cypher().Encrypt(txtIn.Text);

txtEncriptedText.Text = SymetricCypherFactory.Cypher(PROVIDER_NAME).Encrypt(txtIn.Text);
```

Para desencriptar

```
txtDecryptedText.Text = SymetricCypherFactory.Cypher().Dencrypt(txtEncriptedText.Text);

txtDecryptedText.Text =
SymetricCypherFactory.Cypher(PROVIDER_NAME).Dencrypt(txtEncriptedText.Text);
```