TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ECATEPEC.

División De Ingeniería En Sistemas Computacionales.

Titulo:

Segunda vuelta – Módulo encriptación.

P r e s e n t a n:

Escobar Diaz Ulises Iván.

Erick Alberto Ruiz Tique.

A s e s o r:

Griselda Cortes Barrera

24 enero del 2022

Contenido

Introducción 2
Objetivo general 2
Objetivos específicos 2
Encriptación 2
¿Qué es la encriptación? 2
Formas de encriptación 3
Entorno de trabajo 3
Encriptación AES 3
CryptoJS 4
Modelos de encriptación/desencriptación en CryptoJS. 4
PASOS QUE SEGUIR para las paqueterias. 5
¿Cómo implementarlo? 6
Implementación general del módulo 7
Pruebas 11
Repositorio GitHub 11

INTRODUCCIÓN

En el presente documento se muestra la implementación del módulo asignado en el semestre, llamado “Encriptación” se verán los objetivos establecidos, una leve descripción sobre la encriptación, las metodologías utilizadas para el resguardo de la información (Algoritmo AES y Crypto.js), ventajas y desventajas, paqueterías y la implementación dentro de un entorno de trabajo en Angular, comencemos.

OBJETIVO GENERAL

Implementar un método de encriptación para el proyecto de segunda vuelta.

Objetivos específicos

Entender que es la encriptación.

Clasificar y seleccionar las herramientas que se usarán para la encriptación.

Desarrollar el entorno de trabajo para dicha implementación de encriptación

ENCRIPTACIÓN

¿Qué es la encriptación?

Si nos basamos en la definición de la Real Academia Española (RAE) esta dice que encriptar es “*cifrar (|| transcribir con una clave).”* (RAE, Real Academia Española - Encriptar, s.f.) Ahora tomemos la palabra cifrar para la RAE cifrar es “*Transcribir en guarismos, letras o símbolos, de acuerdo con una clave, un mensaje o texto cuyo contenido se quiere proteger.*” Por lo tanto, podemos decir con total seguridad que la encriptación es una forma de resguardar la información para que solo un selecto grupo de personas pueda, observarla, usarla y/o modificarla acorde a las necesidades.

Formas de encriptación

En el mundo de la encriptación se utilizan dos tipos de encriptación los cuales son:

- Encriptación simétrica: Con los algoritmos de clave simétrica, la misma clave se utiliza tanto para encriptar como para desencriptar los datos. Esto permite un cifrado rápido y eficiente; una generación y gestión de claves más sencillas, pero es fundamental que la clave única sólo esté disponible para los usuarios autorizados.
- Cifrado asimétrico: Los algoritmos de clave asimétrica utilizan dos claves matemáticamente relacionadas, una clave pública y una clave privada. La clave pública se utiliza para cifrar los datos, mientras que se requiere una clave privada correspondiente pero separada para descifrar los datos. Es útil para usuarios que cifran y otros más autorizados que requieren tener acceso a los datos.

En el caso de este proyecto usaremos el cifrado “Simétrico” ya que tendremos un algoritmo principal para la desencriptación, pero dependiendo la situación este tendrá una llave acorde a cierto tipo de situación o algoritmo.

ENTORNO DE TRABAJO

Las herramientas que vamos a utilizar son Angular, crypto.js y las metodologías usadas en la encriptación AES.

Encriptación AES

Es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) como FIPS PUB 197 de los Estados Unidos (FIPS 197) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en

un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

Hasta 2005, no se ha encontrado ningún ataque exitoso contra el AES. La Agencia de Seguridad Nacional de los Estados Unidos (NSA) revisó todos los finalistas candidatos al AES, incluyendo el Rijndael, y declaró que todos ellos eran suficientemente seguros para su empleo en información no clasificada del gobierno de los Estados Unidos. En junio de 2003, el gobierno de los Estados Unidos anunció que el AES podía ser usado para información clasificada.

CryptoJS

Para empezar esta es una paquetería para Angular la cual tiene como objetivo la encriptación siguiendo una colección de estándares de seguridad criptográfica implementada por JavaScript usando las mejores prácticas.

Modelos de encriptación/desencriptación en CryptoJS.

Nos ofrece una variedad amplia de estándares de encriptación con más de 14 formas distintas acordes a cada tipo de situación requerida por el o los equipos de trabajo comenzaremos con mostrar unos cuantos de estos y al final el que seleccionamos para el proyecto.

MD5.

Es ampliamente usado en las funciones HASH es utilizado por una amplia variedad de aplicaciones de seguridad, comúnmente son usados para checar la integridad de los archivos.

```
var hash = CryptoJS.MD5("Message");
```

SH3.

Tercera versión y ganador de una competencia para seleccionar el nuevo algoritmo de encriptación, donde un total de 64 competidores fueron evaluados.

La diferencia que tiene este con sus predecesores es la salida de longitud de bits, puede ser desde 224, 256, 384 o 512 bits. La configuración predeterminada es salida por 512 Bits.

```
var hash = CryptoJS.SHA3("Message", { outputLength: 512 });

var hash = CryptoJS.SHA3("Message", { outputLength: 384 });

var hash = CryptoJS.SHA3("Message", { outputLength: 256 });

var hash = CryptoJS.SHA3("Message", { outputLength: 224 });
```

AES.

Algoritmo seleccionado para la realización del proyecto.

```
var encrypted = CryptoJS.AES.encrypt("Message", "Secret

Passphrase");

var decrypted = CryptoJS.AES.decrypt(encrypted, "Secret

Passphrase");
```

Ventajas y deventajas.

En este apartado mostrare las consideraciones a tomar en cuenta al usar CryptoJS con el método de encriptación de AES.

Ventajas	Desventajas
Fácil implementación.	Requiere una clave para desencriptar o encriptar la información
Longitud de encriptación considerable.	
Cambio de longitud acorde a las nececidades del usuario.	

PASOS QUE SEGUIR PARA LAS PAQUETERIAS.

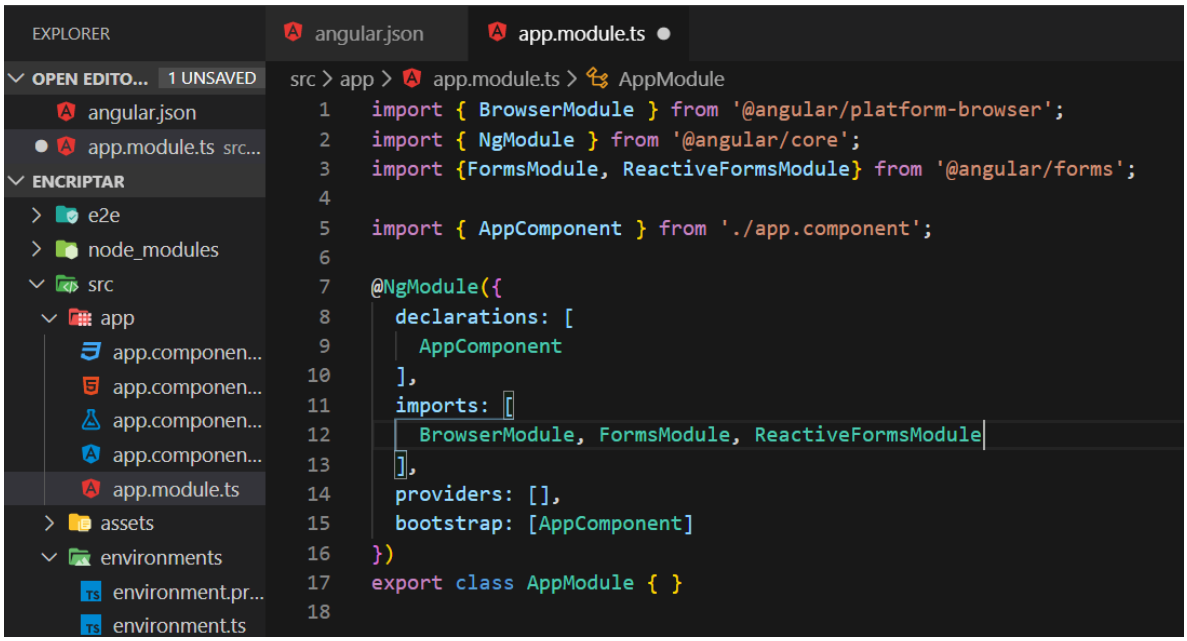
1. Creamos un nuevo proyecto o nos dirigimos a uno ya creado previamente.
2. Instalamos las dependencias a utilizar en nuestro caso usaremos CryptoJS y Bootstrap para la presentación gráfica dentro del proyecto hecho o seleccionado, abrimos un CMD y escribimos.
   ```
   npm install crypto-js –save
   npm install bootstrap –sabe
   ```
3. Ahora dentro del proyecto vamos a agregar unos módulos en “angular.json”, buscamos “styles” y ponemos
   ```
   ["src/styles.css",
   "node_modules/bootstrap/dist/css/bootstrap.min.css"]
   ```

4. donde dice “scripts y ponemos lo siguiente”
   ["node_modules/crypto-js.js",
   "node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"]
   Al final debería quedarte así:

```
"styles": [
  "src/styles.css",
  "node_modules/bootstrap/dist/css/bootstrap.min.css"],


"scripts": ["node_modules/crypto-js.js",
"node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
]
```

5. Nos dirigimos a “app.module.ts” y agregamos “import {FormsModule, ReactiveFormsModule} from ‘@angular/forms’;” y también “imports: [BrowserModule, FormsModule, ReactiveFormsModule];”



¿Cómo implementarlo?

Nos dirigimos al componente al cual queremos implementar el encriptado, nos vamos al apartado de TypeScript, llamamos a la paquetería y después ponemos los algoritmos

```
Import * as CryptoJS from ‘crypto-js’;

// Para encriptar

var encrypted = CryptoJS.AES.encrypt("Message", "Secret
Passphrase");
```

```
// Para desencriptar

var decrypted = CryptoJS.AES.decrypt(encrypted, "Secret
Passphrase");
```

IMPLEMENTACIÓN GENERAL DEL MÓDULO

Los pasos a seguir son los siguientes :

-Abrimos nuestro cmd y creamos el proyecto atraves del comando «ng new nombre-proyecto».

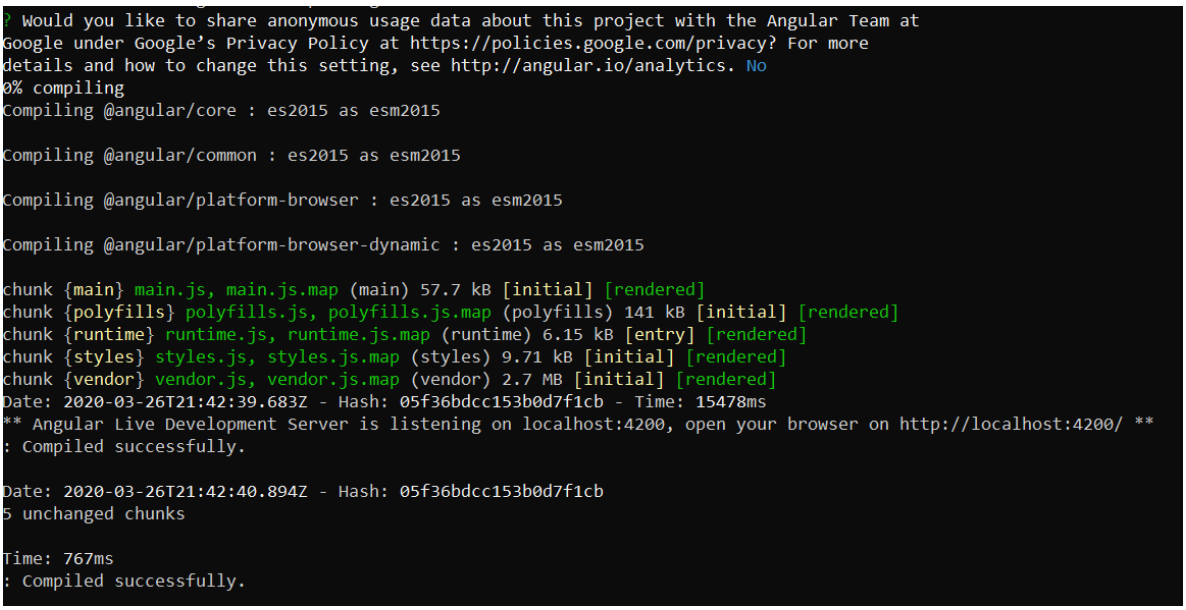
-Luego vamos instalamos las dependencias que usar nuestro proyecto :

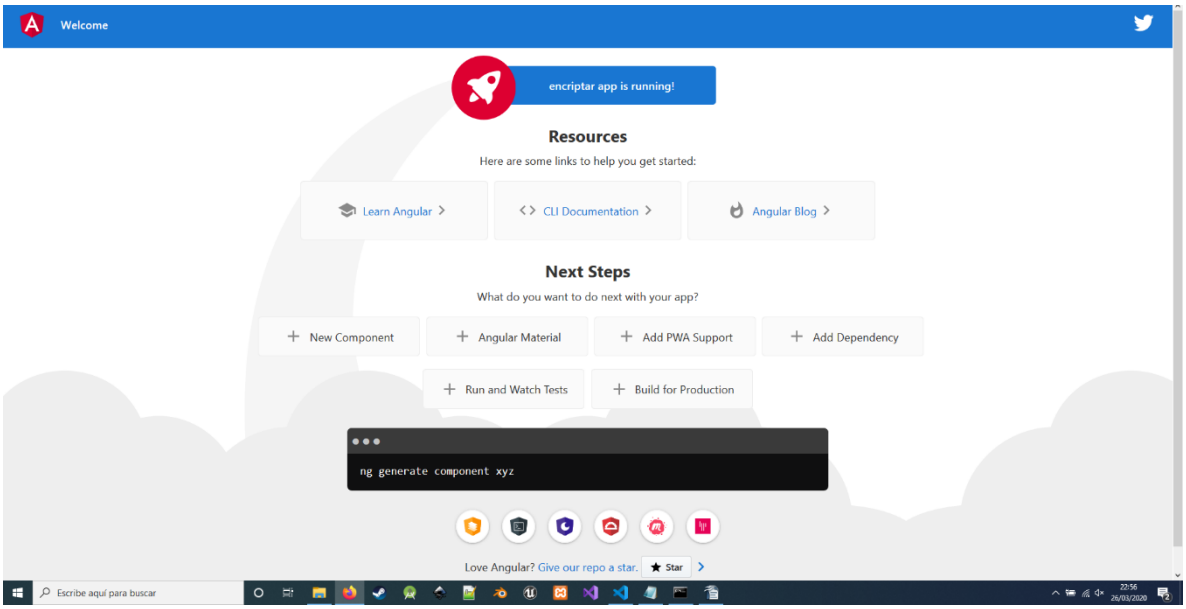
-npm install crypto-js –save

-npm install bootstrap –save

-Iniciamos visual code atraves del cmd «Code .»

-Iniciamos el servidor para ver que la aplicacion corre perfectamente «ng serve -o»





Ahora agregamos los modulos en el angular.json

```
"styles": [
  "src/styles.css",
  "node_modules/bootstrap/dist/css/bootstrap.min.css"],


"scripts": ["node_modules/crypto-js.js",
"node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
]
}
```

Con esto tenemos el proyecto configurado para usar las modulos agregados.

Ahora vamos a importar unas herramientas en el app.module.ts para poder usar formularios en nuestra app.

Hacemos un import de :

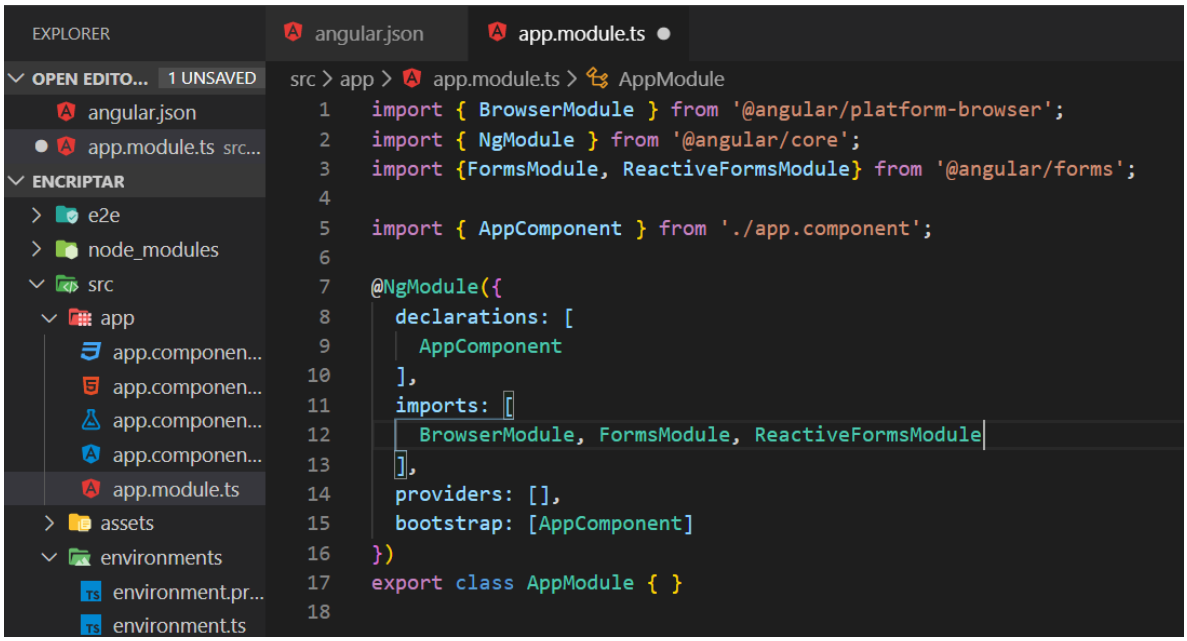
import {FormsModule, ReactiveFormsModule} from '@angular/forms';

y tambien lo agregamos en el imports de @NgModule:

imports: [

BrowserModule, FormsModule, ReactiveFormsModule

],



Creamos un componente.

Una vez creado el componente vamos a la vista del componente «cifrar.component.html»

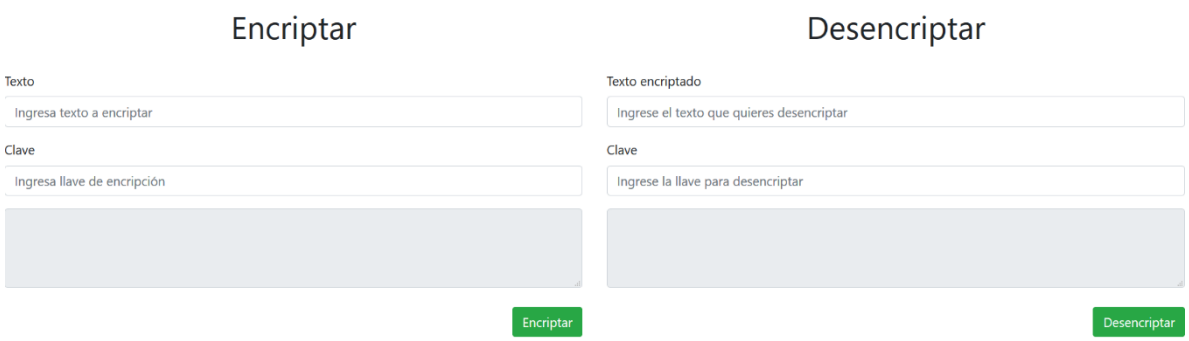
y creamos un formulario que contendra 2 partes :

La primera parte es la de cifrado que contendra las siguientes partes :

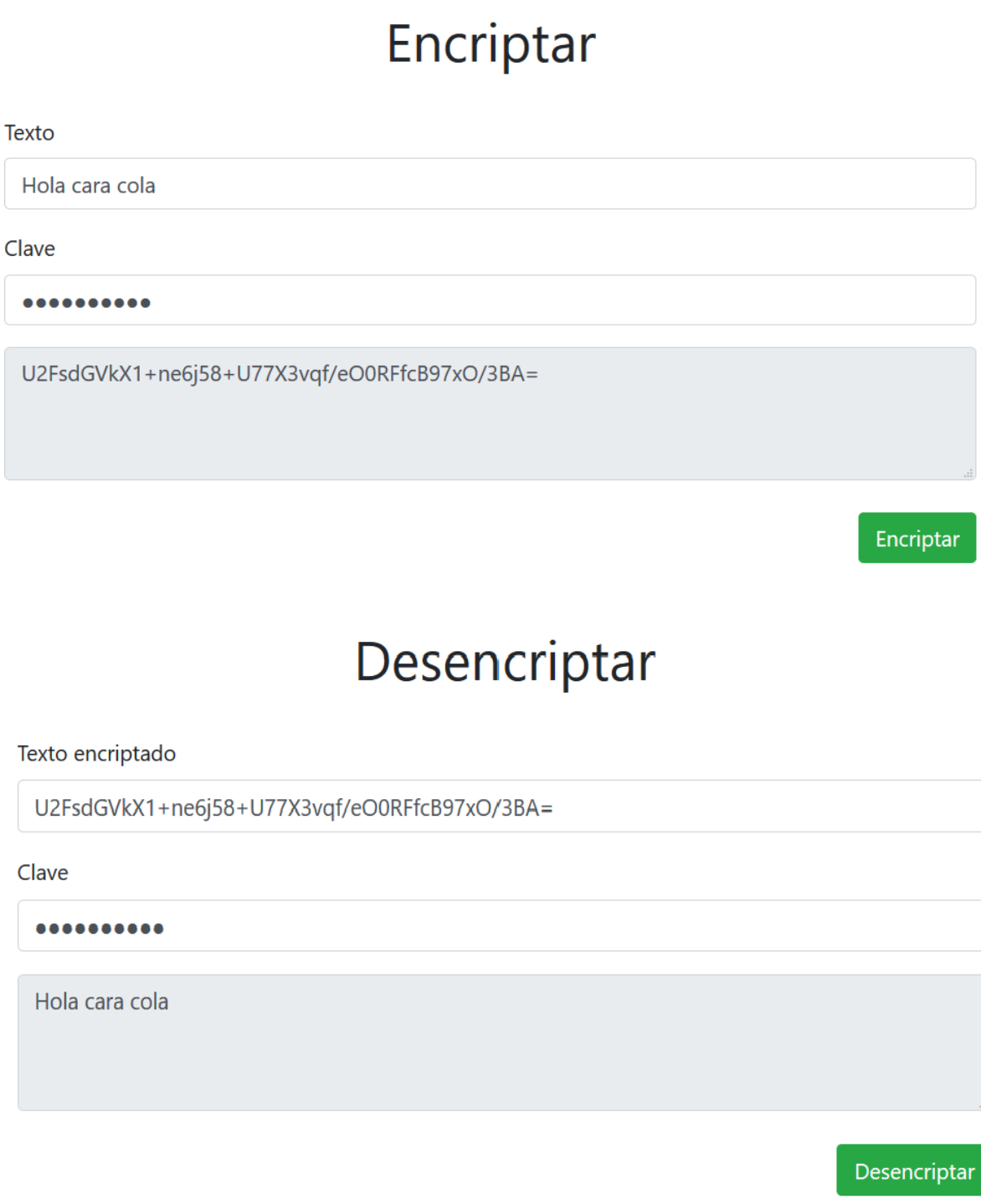
- Una caja para el texto que vamos a cifrar
- Una caja para meter la clave
- Un textarea para mostrar el texto ya cifrado
- Un boton para llamar a la funcion de convertirTexto pasandole un valor «encriptar»

La segunda parte es la de descifrado y contendra las siguientes partes

- Una caja para meter el texto que queremos descifrar
- Una caja para meter la clave usada en el cifrado y que usaremos en esta caja para descifrar el mensaje
- Un textarea que nos mostrara el mensaje descifrado
- Un boton para llamar a la funcion convertirTexto pasandole un valor «desencriptar»



Pruebas



REPOSITORIO GITHUB

https://github.com/sandyCortes/front-generate-reports