

# **Proyecto de Encriptación y Desencriptación en Java**

Java Encryption and Decryption Project

**Gabriel Esteban Vásconez Barahona**

07 de febrero de 2023

## **Proyecto de Encriptación y Desencriptación en Java**

Este proyecto en Java se centra en la encriptación y desencriptación de texto utilizando el cifrado ASCII. El código está organizado en varios paquetes y clases, cada uno con un propósito específico. A continuación, se presenta una descripción detallada de las clases y su funcionalidad:

### **Paquete `com.code.asciiEncryption`**

- **AsciiEncryption:**

**Propósito:** Esta clase se utiliza para encriptar un texto en formato ASCII. Toma un texto como entrada y lo convierte en una serie de números enteros que representan los códigos ASCII de los caracteres en el texto.

#### **Métodos Principales:**

**setNewText(String text):** Toma un nuevo texto como entrada y lo convierte en una serie de números enteros ASCII.

**getText():** Obtiene el texto original.

**getCiphertextIntArray():** Obtiene el texto encriptado como un arreglo de enteros.

**getCiphertextString():** Obtiene el texto encriptado como una cadena de caracteres.

- **AsciiDecryption:**

**Propósito:** Esta clase se utiliza para descriptar un texto en formato ASCII. Puede tomar un arreglo de enteros o una cadena de caracteres como entrada y recuperar el texto original.

**Métodos Principales:**

**setNewCiphertextIntArray(int[] ciphertext):** Toma un arreglo de enteros y lo convierte en el texto original.

**setNewCiphertextString(String ciphertextString):** Toma una cadena de caracteres en formato especial y lo convierte en el texto original.

**getDecipheredText():** Obtiene el texto descriptado.

**Paquete com.code.serializedEncryption**

- **SerializedEncryption:**

**Propósito:** Esta clase implementa la serialización de un objeto de tipo SerializedEncryption. Encripta un texto utilizando un valor de clave y lo serializa en un arreglo de bytes.

**Métodos Principales:**

**encryptNewText(String text, int key):** Toma un texto y una clave como entrada, lo encripta y lo serializa en un ByteArrayOutputStream.

**getCiphertextIntArray():** Obtiene el arreglo de enteros encriptados.

- **SerializedDecryption:**

**Propósito:** Esta clase deserializa un arreglo de bytes encriptado y lo descripta utilizando una clave.

**Métodos Principales:**

**decipherNewText(ByteArrayOutputStream serializedCiphertext, int key):**

Deserializa un arreglo de bytes y lo descripta.

**getDecipheredText():** Obtiene el texto descriptado.

**Paquete com.code.tests**

- **IntArrayTextAsciiEncryptionApp:**

**Propósito:** Esta clase proporciona una interfaz de línea de comandos para encriptar y descriptar texto utilizando las clases AsciiEncryption y AsciiDecryption. Muestra el resultado de la encriptación de un texto en formato de un arreglo de enteros.

**Funcionalidad:** El usuario puede seleccionar la opción de encriptar o descriptar texto y proporcionar el texto correspondiente. El resultado se muestra en la consola.

- **SerializedEncryptionApp:**

**Propósito:** Esta clase muestra la funcionalidad de encriptación y descriptación utilizando las clases SerializedEncryption y SerializedDecryption.

**Funcionalidad:** Encripta un texto, lo serializa y luego lo deserializa y descripta.

- **StringTextAsciiEncryptionApp:**

**Propósito:** Esta clase proporciona una interfaz de línea de comandos similar a `IntArrayTextAsciiEncryptionApp`, pero muestra el resultado de la encriptación de un texto en formato de cadena de caracteres en lugar de un arreglo de enteros.

### **Conclusiones:**

Este proyecto en Java proporciona una implementación funcional de la encriptación y descriptación de texto utilizando el cifrado ASCII. Las clases y paquetes están organizados de manera clara y modular, lo que facilita la comprensión y el mantenimiento del código. Las aplicaciones de prueba proporcionan una interfaz fácil de usar para probar las funcionalidades de encriptación y descriptación.

### **Recomendaciones:**

- **Validación de Entrada:** En las aplicaciones de prueba, se pueden agregar validaciones adicionales para garantizar que las entradas del usuario sean adecuadas y para manejar errores de manera más robusta.
- **Mejoras de Interfaz:** Se pueden realizar mejoras en la interfaz de usuario de las aplicaciones de prueba para hacerlas más amigables y fáciles de usar.

El proyecto demuestra una implementación sólida de la encriptación y descriptación en Java y tiene potencial para futuras mejoras y aplicaciones.