Domenica Rosado Fernandez Salvador
Carrera de Ingeniería en Ciberseguridad
Lógica de Programación 1-ECC-1B

23 de enero 2026

Aprendizaje Autónomo 1

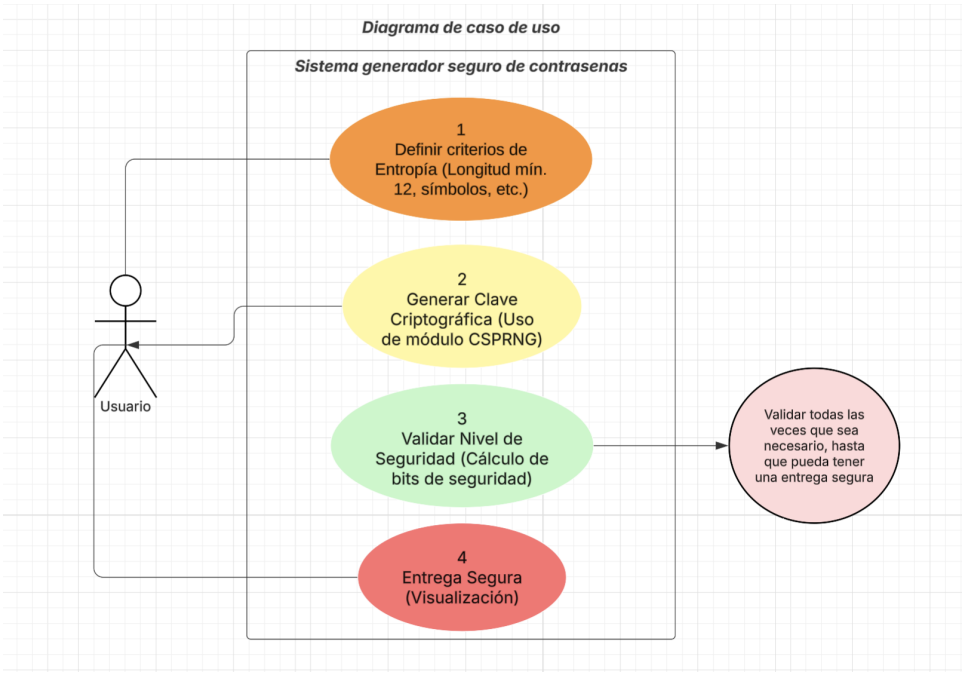
Seleccione el generador de contraseñas seguras como el software a desarrollar. En el proyecto me enfoque en la seguridad mas que nada, entonces se implemento algoritmos criptográficamente fuertes.

Análisis:

El problema que se plantea consiste en la generación de cadenas de texto aleatorias que sean seguras y difíciles de predecir. Para que el sistema funcione correctamente, es necesario cumplir con ciertos requisitos básicos de seguridad. En primer lugar, el sistema debe permitir que el usuario personalice las cadenas generadas. Esto incluye la posibilidad de elegir la longitud del texto, recomendándose un mínimo de 12 caracteres para un nivel de seguridad adecuado, así como seleccionar los tipos de caracteres que se desean utilizar, como letras mayúsculas, números y símbolos. Ejemplo: 983j#4dKQ{e2.

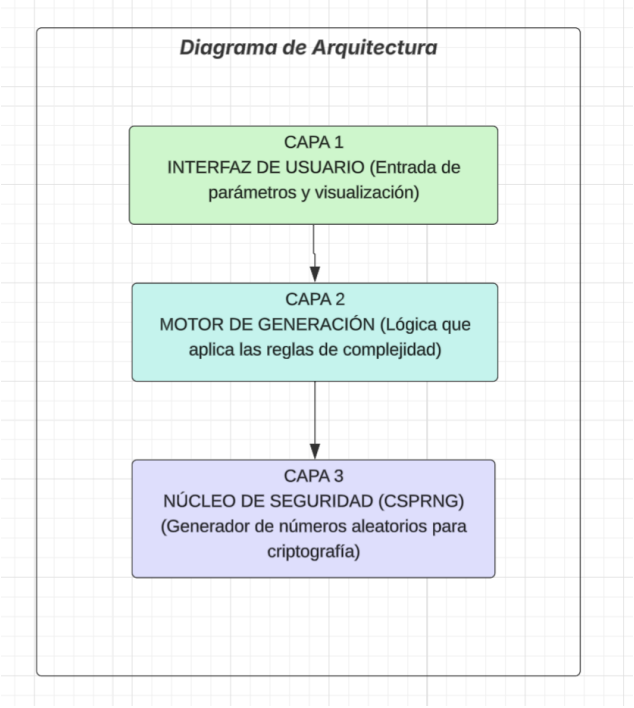
Otro aspecto importante es la aleatoriedad. El programa debe utilizar un generador de números aleatorios seguro desde el punto de vista criptográfico, con el fin de evitar patrones repetitivos que puedan facilitar el descubrimiento de las claves generadas. Y finalmente, se debe considerar la seguridad de la información. La aplicación no debe guardar ni almacenar las cadenas generadas, siguiendo el principio de no persistencia, lo que ayuda a proteger los datos del usuario y reduce posibles riesgos de seguridad.

Diagrama: Utilizando la aplicación de Lucid, y seleccione como plantilla el diagrama de caso de uso.



El diagrama muestra al **Usuario** en la izquierda, como el actor principal que dispara las funcionalidades del sistema. El flujo interactivo permite que el usuario configure los parámetros, inicie la generación y reciba el producto final (la contraseña segura), mientras el sistema realiza validaciones internas de seguridad de forma transparente.

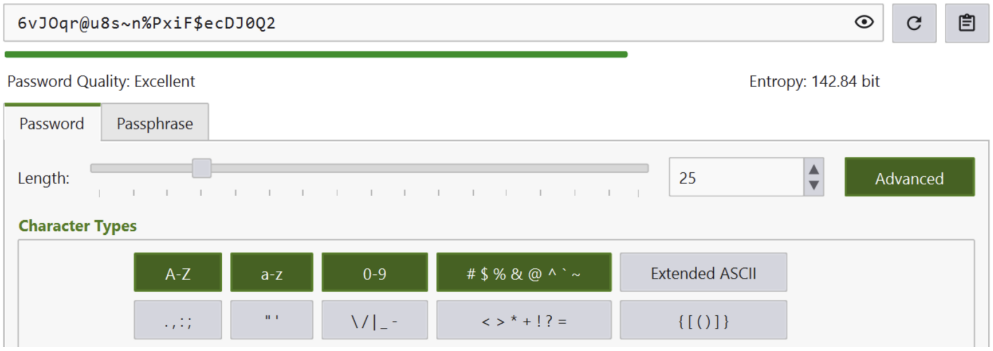
Diagrama de Arquitectura:



Capa 1, de presentación (Frontend) es la parte del sistema con la que el usuario interactúa directamente. Su función principal es permitir que el usuario ingrese los requisitos deseados, como la longitud y el tipo de caracteres, y mostrar en pantalla la clave que se genera.

Capa 2, de lógica (Business Layer) se encarga de procesar la información ingresada por el usuario. En esta capa se encuentra el algoritmo que organiza y construye la contraseña de acuerdo con las reglas de complejidad seleccionadas, asegurando que se cumplan los criterios establecidos.

Capa 3 de seguridad (CSPRNG) es el componente más importante del sistema en términos de protección. Esta capa se comunica con las librerías de seguridad del sistema operativo para obtener valores aleatorios seguros, los cuales permiten generar claves difíciles de predecir y más resistentes a ataques. Hay una aplicación que se llama Keepass que utiliza el mismo sistema.



El funcionamiento del sistema se basa en una estructura modular, en la que cada parte cumple una función específica. La interfaz no tiene acceso directo a la fuente de aleatoriedad, lo que ayuda a mantener una mejor organización y mayor seguridad. Cuando el usuario solicita una clave, la capa de lógica se encarga de comunicarse con el motor de seguridad, procesar la cadena generada y enviarla a la interfaz únicamente para su visualización, sin almacenarla. De esta manera, el software se mantiene ligero, rápido y más seguro frente a posibles intentos de predicción o ataques.