1. A motivação da nossa escolha para esse código foi fazer com que os usuários não tenham risco da sua senha ser descoberta com facilidade (como por exemplo “manu123”, sendo uma “fácil” de ser descoberta), tornando assim o nosso código chamado de “***Senha Segura***”.

O nosso código gera uma senha, porém não qualquer senha, e sim uma senha segura, em que o usuário digita informações (como símbolos, números...), não necessariamente pessoais, na qual essas informações irá gerar a “***Senha Segura***”. O objetivo dela é o usuário ter a segurança de que sua senha não vai ser descoberta com facilidade, pois o código mistura e alterna as escolhas de senha de acordo com as informações que ele preencher.

Nesse código o usuário preenche os campos como digitando números, letras maiúsculas e minúsculas e símbolos. Após isso, o código gera uma senha alternando a sequência que o usuário preencheu, embaralhando as informações. Além de que se o usuário preencher algum campo errado, o programa para de rodar e mostra o motivo.

* Usamos o **def** no nosso código, onde ele em geral é usado para criar funções, que são blocos de código que fazem coisas específicas. No código da “***Senha Segura***”, ele está sendo usado para validar uma função, dando retorno de verdadeiro ou falso.
* Usamos o **return** é uma função pode ou não retornar algum valor. No código da “***Senha Segura***” ela está sendo usada para retornar se as funções são verdadeiras ou falsas.
* Usamos o **isupper**, que verifica cada caractere na string e verifica se seu código da aquela função corresponde a uma letra especifica . Se todos os caracteres são maiores, a função retorna `True`. Se pelo menos um caractere não for uma letra ou for uma letra minúscula, a função retorna `False`. No código da “***Senha Segura***”, ele está sendo usado para validar se a letra é maiúscula ou não.
* Usamos o **Islower,** que verifica cada caractere na string e verifica se seu código da aquela função corresponde a uma letra específica . Se todos os caracteres são maiores, a função retorna `True`. Se pelo menos um caractere não for uma letra ou for uma letra maiúscula, a função retorna `False`. No código da “***Senha Segura***”, ele está sendo usado para validar se a letra é minúscula ou não.
* Usamos o **`set`** que é uma estrutura de dados que representa uma coleção não ordenada e mutável de elementos únicos. No código da “***Senha Segura***”, ele está sendo usado para montar um conjunto onde ele suporta operações de conjuntos como união (`|`), interseção (`&`), diferença (`-`)... Resumindo, o uso de símbolos.
* Usamos o método **`isdigit()`,** que pertence à classe de strings. Ele retorna `True` se todos os caracteres na string são dígitos (0 a 9) e se a string não estiver vazia. Caso contrário, retorna `False`. No código da “***Senha Segura***”, ele está sendo usado para retornar se foi preenchido somente com números ou não.
* Usamos **sys.exi**, que sua função é comumente usada para encerrar um programa quando ocorre um erro fatal ou quando certas condições são atendidas para encerrar a execução. `. No código da “***Senha Segura***”, ele está sendo usado justamente para o código parar de rodar se houver erro.
* Usamos **`join`**, que é um método que pertence a strings e é usado para concatenar elementos de uma sequência (geralmente uma lista) em uma única string. Ele funciona ao inserir uma string (o "separador") entre cada par de elementos da sequência, criando assim uma string única. O método é chamado na string que servirá como separador e recebe a sequência como argumento. No código da “***Senha Segura***”, ele está sendo usado para juntar as informações que o usuário preencheu nos campos, gerando assim a senha segura e única.
* Usamos o **random.shuffle**, que embaralha uma sequência (como uma lista) aleatoriamente. No código da “***Senha Segura***”, ele está sendo usado para embaralhar as informações que o usuário preencheu, formando assim a senha.