SWD-Basic. Asignación final.

En el contexto de la seguridad informática, es crítico contar con mecanismos efectivos para proteger la información privada y sensitiva (Ver. GDPR, ISO 27001, entre otros.). Definir una buena contraseña para acceder a recursos privados es uno de los primeros pasos en ese objetivo. Es una práctica sencilla pero potencialmente catastrófica si no se tienen en cuenta las recomendaciones básicas para definir una contraseña segura que no sea vulnerada fácilmente.

Ejemplo de contraseñas no seguras, son aquellas que pueden ser detectadas por algoritmos de fuerza bruta o por comparación contra directorios de contraseñas habituales: password, 1234, 12345678, qwerty, etc. Utilizar este tipo de contraseñas deja en alto riesgo los activos de información que deben permanecer seguros.

Usted ha sido designado para construir una herramienta en Java para facilitar la generación de contraseñas seguras. Esta herramienta se utilizará por línea de comando.

Generar una contraseña debe tener en cuenta: la longitud final de la contraseña, si debe contener solo letras (mayúsculas, minúsculas o ambas), solo números o una mezcla de letras y números. Debe ser una opción para el usuario excluir los llamados caracteres ambiguos: Dado que contraseñas con i, I, L, l, 1 (Letra i minúscula y mayúscula, L mayúscula y minúscula, número 1) son fácilmente confusas, una buena cantidad de usuarios preferirá evitar su uso. Algo similar ocurre con o, O, 0 (Letra o minúscula, mayúscula y cero).

Incluir símbolos contribuye a la seguridad de la contraseña: @#$%& entre otros. Sin embargo, se debe dejar opcional al usuario excluir determinados caracteres no deseados: Por ejemplo: <,>, {, }, [, ] .. entre otros.

Por último, el usuario podrá decidir cuantas contraseñas generar en una sola invocación. **Cada contraseña generada debe ser diferente** (utilizar generación aleatoria). Será opcional copiar la contraseña generada directamente al ClipBoard y así dejarla disponible para pegar (Ctrl + V) en donde el usuario desee (Ver más adelante como hacerlo).

Ejemplos de uso:

> java PwdGen –Size:16 –U –L –N –S –A –E:[]{}@ –T:3 -C

-Size:16 Contraseña de 16 caracteres

-U Incluir letras en mayúscula (Uppercase)

-L Incluir letras en minúscula (Lowercase)

-N Incluir números (Numbers)

-S Incluir símbolos (Symbols). Entiéndase caracteres como: $%&()=# entre otros.

-A Excluye los caracteres ambiguos: i, I, L, l, 1, o, O, 0

-E:[]{}@ Excluir los símbolos suministrados. En este caso la contraseña generada no contendrá los caracteres []{}@

-T:3 Generar 3 contraseñas (Total)

-C Copia la contraseña generada en el ClipBoard para solo hacer CTRL + V y utilizarla. (Ver más adelante como hacerlo). Si el usuario genera múltiples contraseñas, solo la última se copiará al Clipboard.

> java PwdGen –?

Desplegará un texto describiendo las opciones. Pueden intentar con los comandos de Windows, en una consola de comando para ver un ejemplo. Si las opciones ingresadas no son válidas se mostrará este mensaje descriptivo.

> dir /?

**Nota**. Todas las opciones son por definición “opcionales”. Por defecto, la herramienta PassGen generará una contraseña de 16 caracteres (letras mayúsculas, minúsculas, números), sin símbolos, sin exclusiones y sin copiar al clipboard.

**Ejemplos.**

> java PwdGen –Size:8 –S –Copy

Genera una contraseña de 8 caracteres que contiene solo símbolos y disponible en el Clipboard.

> java PwdGen –U –N –A

Genera una contraseña de 16 caracteres (valor por defecto) que contiene solo letras en mayúsculas y números y excluyendo los caracteres ambiguos (i, I, l, L, 1, o, O, 0).

> java PwdGen –T:5 –A –Size:10

Genera 5 contraseñas de 10 caracteres que contiene letras en mayúsculas, minúsculas y números (valores por defecto) y excluyendo los caracteres ambiguos (i, I, l, L, 1, o, O, 0).

> java PwdGen –S –E:=()$% -N –Copy

Genera una contraseña de 16 caracteres (valor por defecto) que contiene símbolos y números disponible en el Clipboard, excluyendo los caracteres =()$%

> java PwdGen –E:abc123DEF

Genera una contraseña de 16 caracteres (valor por defecto) que contiene letras en mayúsculas, minúsculas y números (valores por defecto) excluyendo los caracteres abc123DEF

**Nota 2.** Investigar cómo pasar argumentos desde Eclipse a un programa en java. Estos argumentos son los que llegan en el arreglo String[] args del método main().

**Nota 3.** Suministrar el código fuente y las pruebas unitarias (JUnit 4). Recomendación: Utilizar la metodología TDD.

**Nota 4**. Estudiar el siguiente código de ejemplo, para copiar un String en el Clipboard.

**import** java.awt.Toolkit;

**import** java.awt.datatransfer.Clipboard;

**import** java.awt.datatransfer.StringSelection;

**public** **class** TestClipBoard {

**public** **static** **void** main(String[] args) {

String h = "Hola Mundo";

StringSelection selection = **new** StringSelection(h);

Clipboard clipboard = Toolkit.*getDefaultToolkit*().getSystemClipboard();

clipboard.setContents(selection, selection);

}

}