

PROYECTO DE LENGUAJES DE PROGRAMACIÓN

Prof. Dennis Romero L.

Cajero automático virtual, anti-espía.

La idea del proyecto es desarrollar un sistema que permita resolver el problema del robo de contraseñas numéricas en los cajeros automáticos.

1) Análisis del problema

Como se sabe, cada usuario del cajero automático conoce una clave de 4 dígitos, el cual permite acceso a la información de la cuenta bancaria y a realizar transacciones como: retiros de dinero, consultas, transferencias, entre otros. Sin embargo, el proceso actual de digitar la clave en el cajero automático presenta vulnerabilidades que atentan contra la privacidad del usuario. Para esto, los bancos han adoptado formas de dificultar el "espionaje" de claves realizada por terceras personas, mediante la colocación de barreras que cubren el teclado o techos para evitar ver el teclado y los botones que son presionados. Esto es incómodo y no soluciona los problemas de espionaje de claves por parte de terceros.

Si una tercera persona malintencionada logra ver la secuencia numérica que el usuario digitó en el teclado del cajero automático, existe la posibilidad de fraude, debido a que la tarjeta es vulnerable a clonación o robo. En este sentido, el sólo hecho de que alguna persona mal intencionada conozca la clave numérica de un tarjetahabiente, expone a este último a riesgos como secuestro, robos, asaltos y pérdidas económicas.

2) Análisis de la solución

La idea de solución es evitar que el usuario digite directamente su clave numérica en el cajero automático. En lugar de esto, la persona ingresará una "máscara alfabética", generada a partir de su clave numérica.

La combinación de letras (una o dos letras representando cada número) será generada de forma aleatoria a partir de la clave del usuario. Es decir, la clave numérica del usuario representa la "semilla (seed)" para el método que generará aleatoriamente la combinación de letras y permitirá convertir la clave del usuario en su respectiva secuencia de letras.

Ejemplo:

Nombre del usuario: Amanda Perez

Clave numérica: 9875

Secuencia aleatoria generada a partir de la clave: Re-Sa-V

Al momento en que el usuario se acerca al cajero e ingresa su tarjeta, el sistema reconocerá el **ID** del usuario y le pedirá ingresar su secuencia alfabética, donde las letras

que el usuario conoce se encuentran mezcladas y alineadas junto con otras letras que no corresponden con la combinación de letras que el usuario conoce. De esta forma, todas las combinaciones de letras (las que el usuario conoce y las que no) estarán distribuidas en la pantalla, también de forma aleatoria. Es decir, en cada utilización del cajero automático, el usuario tendrá secuencias diferentes de combinaciones de letras, pero siempre dentro de esas secuencias, estarán incluidas también las combinaciones de letras que el usuario conoce.

Ejemplo:

Bienvenida Amanda Perez, por favor ingrese su secuencia de letras:

BOTON1	We Re Li K A	Ke Pe Ta Le N	BOTON4
BOTON2	E Pi Ka Lo In	Ga Fi E A Ni	BOTON5
BOTON3	Pa Yu V Yi O	Ta Po O Ra Sa	BOTON6

Por medio de los seis botones del cajero, el usuario escogerá la línea en donde se encuentra cada una de sus combinaciones de letras:

Ejemplo:

BOTON1 - BOTON6 - BOTON3

(Re-Sa-V)

De esta forma, aunque el espía pueda ver claramente la pantalla del cajero y pueda ver qué botones el usuario ha presionado, no podrá saber cuál fue la combinación alfabética que el usuario conoce.

3) Detalles del proyecto:

- El objetivo es desarrollar un cajero anti-espía virtual. Pueden realizarlo con la ayuda de interfaz gráfica o no. Los lenguajes a utilizar serán Python y Objective-C, en cualquier sistema operativo.
- Pueden hacer uso de cualquier función o librería que genere número aleatorios.
- No es obligatoria la utilización de bases de datos ni arquitectura cliente-servidor, pero hay libertad de hacerlo siempre y cuando no sea un motivo para retrasos en el proyecto, sin embargo, se valorará la creatividad en el desarrollo por lo que pueden invertir también en algo que de mejores resultados.
- Condiciones necesarias:
 - Permitir la creación de usuarios
 - Nombre
 - # de cuenta
 - Monto en la cuenta
 - Otras informaciones a discreción

- Asignar una clave numérica al cliente (no es obligación que sea guardada encriptada, pero si usan algún algoritmo de encriptación será también valorado.)
- Permitir el ingreso del usuario por medio de su ID de tarjeta
- Mostrar en pantalla las combinaciones conocidas y no conocidas de sílabas/letras generadas y organizadas de forma aleatoria, como muestra el literal anterior.
- Validar que la secuencia ingresada sea la correcta
- En caso de que el ingreso fue satisfactorio, mostrar la información del usuario y/o permitir realizar una transacción
- Salida del sistema cuando se supera un tiempo de no-interacción y salida manual por medio de “cerrar sesión/Salir”

4) Uso de repositorio Git en BitBucket:

Se asume que leyeron el libro proGit y realizaron las configuraciones iniciales, tales como:

```
git config --global user.name "YOUR NAME"
git config --global user.email "YOUR EMAIL ADDRESS"
```

Pasos:

- 1) En la cuenta de Bitbucket de uno de los integrantes del grupo, creen dos repositorios privados, uno llamado **LDP_GrupoX_ObjC** y el otro **LDP_GrupoX_Py**, especificando los lenguajes correspondientes (siendo X el número de cada grupo). Habiliten la opción de Wiki e Issue tracking.
- 2) Después de cada repositorio creado, hacer (como lo muestra el link de bitbucket “I’m starting from scratch”, luego después de la creación de un repositorio):

```
mkdir /ruta/de/su/proyecto
cd /ruta/de/su/proyecto
git init
git remote add origin git@bitbucket.org:usuarioN/grupox.git
(cambiar los datos de usuario según su cuenta)
```

- 3) El primer archivo a colocar dentro de la carpeta será un archivo de *Planner* (disponible para Linux y Windows) con el cual detallarán todas las actividades asociadas a su proyecto, el tiempo que les tomará realizar cada actividad y el orden en el que cada actividad será realizada. *Planner* les ayudará a visualizar por medio de un Diagrama de Gantt, cuál será la fecha de finalización del proyecto, la cual debe corresponder con la fecha de entrega del mismo.
- 4) Una vez colocado el archivo dentro de la carpeta, deberán realizar su primer *commit* en git:

```
git commit -a -m "Plan de actividades"
```

y posteriormente subirlo al repositorio con:

```
git push -u origin master
```

- 5) Finalmente, invitar a su compañero de grupo a cada uno de los repositorios y colocarle permisos de escritura, invitarme también por medio del usuario dromerol con permisos de admin.

Sugerencias:

- Verifiquen constantemente su progreso.
- Incluyan sus actividades de desarrollo en el plan de actividades, de esta forma el proyecto avanzará de acuerdo con cada actividad cumplida.
- Realicen *commits* cuando hayan realizado avances que consideren importantes y súbanlos al repositorio. Recuerden que una de las ventajas de *Git* es que les permite revisar *commits* anteriores y eso ayuda mucho en caso de problemas. Además evita tener que crear carpetas con versiones diferentes.