



En este tutorial se muestra cómo programar un cajero automático con lenguaje de programación gráfico en el software LabVIEW.

Se utilizó una imagen de un cajero físico para que el programa sea lo más real posible.

El panel frontal quedó de la siguiente forma:

(Clic en las imágenes para ampliar).



Imagen #1: Panel frontal del cajero automático.

En la pantalla se encuentran dos indicadores de texto para todos los mensajes que el programa muestre.

Hay cuatro indicadores numéricos para mostrar el NIP que se escriba en el teclado en caso de que este mismo sea correcto.

Se utilizaron botones para las teclas numéricas y los demás controles, incluyendo el que dice tarjeta, que simula la inserción de la tarjeta cuando se oprime.

Se utilizó un led cuadrado en la parte donde se inserta la tarjeta, para indicar que el cajero está listo para ser utilizado, como lo hacen algunos en la realidad.

Hablando de la programación, se explicará el código por partes ya que es algo extenso. Para empezar, todo el programa funciona con un ciclo while, que hace que el programa se esté ejecutando indefinidamente hasta que el usuario presione el botón salir.





Imagen #2: Casos para el botón "Tarjeta".

Primero, se tiene al botón que simula la inserción de una tarjeta, conectado a una estructura case, como se puede observar en la imagen #2. A su vez, cuando el caso sea verdadero, se tienen otras estructuras anidadas.



Imagen #3: Caso falso para el control de la tarjeta.

Entonces, cuando el botón de tarjeta no se ha oprimido, se estará ejecutando un ciclo FOR (Imagen #3), éste multiplica un número aleatorio por el número de iteraciones por ciclo, para que, cuando el resultado esté en un rango entre 0 y 5, se encienda el led de la tarjeta, en caso contrario se apaga, esto mediante constantes True y False (Verdadero y Falso, ó 0 y 1).

Al mismo tiempo, se podrá ver un mensaje de bienvenida en el indicador de texto de la cabecera.



Imagen #4: Código para la solicitud de código NIP.

Entonces, cuando se oprima el botón de la tarjeta, el programa estará esperando un código NIP de cuatro dígitos, para lo cual se tiene programado que sean las combinaciones de los números 2,3,4 y 5, (2345, 3245, 4532, 4235, 5234, etc). Esto se hizo mediante una compuerta AND de cuatro entradas. Entonces, si se presionan cuatro números que estén comprendidos entre el 2 y el 5, se toma como caso verdadero, dentro del caso, el programa muestra el NIP ingresado utilizando constantes numéricas para los indicadores.

A su vez, los botones del teclado numérico que se presionaron, permanecerán presionados (en modo true) hasta que los indicadores numéricos del NIP marquen el número correcto,



entonces mediante otra estructura case anidada, se “apagan” los botones, quedando nuevamente en modo false, asimismo, el texto del encabezado se borra y se muestra un mensaje para continuar en la pantalla principal.

Lo anterior se muestra en la imagen #5:



Imagen #5: Ingreso correcto de NIP en el programa.

De igual forma, se utiliza una estructura case para el botón corregir, el cual pone a todos los botones del teclado numérico en false. (Ver imagen #6).



Imagen #6: Programación del botón corregir.

Para que lo anterior pudiera funcionar, se cambió el modo de funcionamiento de los botones, que por default se encuentran en false, al presionarlos se ponen en true pero vuelven a false inmediatamente después de soltar el botón, como una pulsación. Entonces, se configuraron para que al presionarlos, se mantengan en true hasta que se vuelvan a presionar, aún después de soltar el botón.



Imagen #7: Nodos de propiedad para los botones de retiro.

Como se observa en la imagen #7, se utilizaron nodos de propiedad de los botones para retiro de las diferentes cantidades de dinero (\$500, \$1,000, \$3,000 y \$5,000). Estas permanecen invisibles gracias al nodo de propiedad para visibilidad, el cual permanece falso



(invisible) mediante una constante booleana.



Imagen #8: Programación del botón "Ver saldo".

Al presionar el botón ver saldo, se muestra en el indicador de texto, el saldo que la cuenta tiene (Imagen #8), de igual forma, dentro del case de ese botón, se anida otro case para el botón cancelar, el cual en caso de ser verdadero, borra los indicadores del NIP, los indicadores de la pantalla y pone en false el led de la tarjeta. Esto simula que el cajero saca la tarjeta y reinicia el proceso.



Imagen #9: Resultado de presionar el botón "ver saldo".

Lo que normalmente prosigue después de este punto, es retirar el dinero, entonces, con otra estructura case para el botón de retiro de dinero, se cambia el estado de los nodos de propiedad de las cantidades a true, o sea, visibles. Y también, se muestra en la pantalla un texto para indicar que seleccione una cantidad a retirar.



Imagen #10: Estructura case para el botón de retiro de dinero.

Entonces, lo que el usuario verá al presionar ese botón, es lo siguiente:





Imagen #11: Visibilidad de los botones para retiro de dinero.



Imagen #12: Programación de los botones de cantidades.

Finalmente, al presionar algún botón de cantidad, se pone en modo false el botón de retiro de dinero, mediante una variable local del mismo, este botón permanecía encendido (true) para que las cantidades puedan verse.

De igual forma, se muestra un mensaje con el nuevo saldo en la cuenta.

Lo anterior se hace para las demás cantidades.



Imagen #13: Configuración del botón de salida.

Para salir del programa, o mejor dicho, retirar la tarjeta, se presiona el botón de salir, el cual apaga el led de la tarjeta (simulación de expulsión), borra los indicadores del NIP, borra las pantallas y termina la ejecución del programa.

Se puede ver un vídeo demostrativo aquí abajo o puedes descargar el programa dando clic [AQUÍ](#).