

# Simulación de un Cajero Automático

María Cecilia Fernández Montefiore

Universidad Nacional del Sur

**Resumen** El presente trabajo es la descripción de un programa que simula un cajero automático. Se realiza el modelado del mismo como un autómata finito, especificando cada uno de los elementos que lo definen como tal. Además se detallan las funcionalidades adicionales como la lectura y actualización de cuentas y el bloqueo y la clave de operario. Se explica la construcción de la interfaz gráfica. Da una guía de uso para los usuarios, en el que se explican los pasos a seguir. Finalmente se detallan posibles mejoras a futuro y se concluye que la realización del proyecto ha servido como un integrador de los temas de la materia. Este se realizó como un proyecto para la cátedra de Principios de Computadoras II.

## 1. Introducción

Un cajero automático es un aparato cuya finalidad es automatizar y agilizar las tareas bancarias, sin necesidad de personal del banco. Permite extraer dinero, realizar depósitos, consultar saldo de cuenta, entre otras tareas; usando una tarjeta de plástico con banda magnética.

También es conocido como “ATM”, por siglas en inglés de *Automated Teller Machine*.

Debido a sus características, esta máquina puede ser modelada como un Autómata Finito Regular.

En el presente trabajo se programó en C++ una emulación del cajero mediante el modelado de su comportamiento y una interfaz gráfica realizada con librerías de Qt.

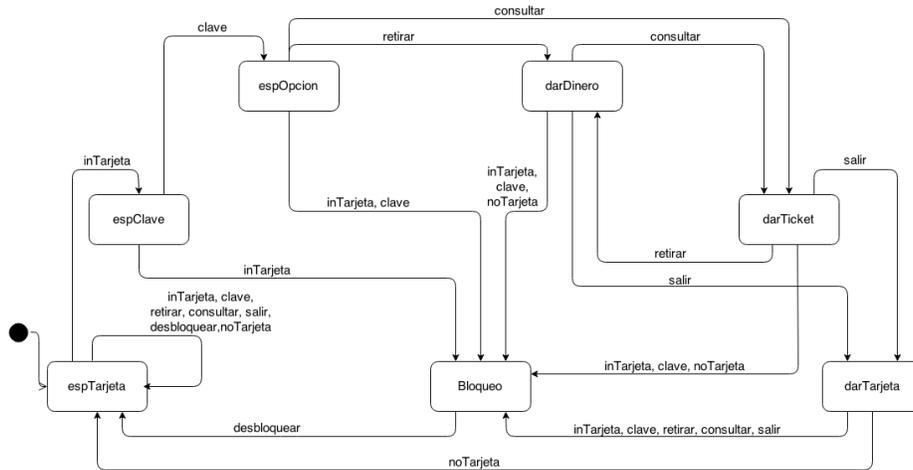
## 2. Desarrollo

La simulación del cajero automático cuenta con dos funciones: retirar dinero y consultar saldo. Ambas se realizan a partir de una base de datos de tarjetas, que se actualiza con las operaciones. Este cajero automático se programó de manera de separar las funciones de la interfaz gráfica con el autómata. Esto es una ventaja a la hora de realizar mantenimiento y actualización del programa.

### 2.1. Descripción del autómata

Un autómata finito regular se define como una máquina:

$$M = (S, \Sigma, Z, \delta, s_0, F) \quad (1)$$



**Figura 1.** Diagrama de estados

Donde  $S$  es un conjunto finito de estados,  $\Sigma$  es el alfabeto de entrada,  $Z$  es el alfabeto de salida,  $\delta$  es una función de transición,  $s_0$  es el estado inicial y  $F$  es la función de salida [Coppo Ricardo, 2014].

El cajero automático se diseñó como un autómata de siete estados. Permite siete entradas y cuenta con tres salidas.

Los estados son  $S = \{ espTarjeta, espClave, espOpcion, darDinero, darTicket, darTarjeta, bloqueo \}$ . En los primeros tres estados se espera que el usuario realice una acción, que se corresponde con alguna entrada, como insertar la tarjeta, ingresar la clave correcta o seleccionar una opción. Los siguientes tres estados son los que poseen salidas, es decir que se entrega dinero, un comprobante o se devuelve la tarjeta. El último estado es de seguridad.

Las entradas son:  $\Sigma = \{ inTarjeta, clave, retirar, consultar, salir, desbloquear, noTarjeta \}$ . La acción de insertar y retirar la tarjeta está representada por la entrada  $inTarjeta$  y  $noTarjeta$ . El ingreso de la clave se corresponde con la entrada  $clave$ , mientras que  $desbloquear$  es cuando el operario ingresa la clave de desbloqueo.

Por último, las salidas son  $Z = \{ dinero, tarjeta, comprobante \}$ .

En la figura 1 puede verse el diagrama de estados del autómata.

El estado inicial  $s_0$  es “espTarjeta”. Si se inserta una tarjeta válida en el cajero, se pasa al estado de “espClave”. Ante cualquier otra entrada, permanece en el estado inicial.

Desde “espClave” si se ingresa la clave correcta se avanza hasta “espOpcion”. Si se intenta ingresar otra tarjeta, o se quiere retirar dinero o un comprobante, el cajero queda en estado de “bloqueo”. Si se sale o se retira la tarjeta vuelve al estado inicial.

Desde “espOpcion” si la entrada es retirar o consultar se va hacia los estados “darDinero” y “darTicket” respectivamente. Si se sale o se retira la tarjeta, se vuelve al estado inicial. Ante cualquier otra entrada (excepto desbloquear) el cajero se bloquea.

	Eventos						
<b>Estados</b> ↓	inTarjeta	clave	retirar	consultar	salir	desbloquear	noTarjeta
espTarjeta	espClave	espTarjeta	espTarjeta	espTarjeta	espTarjeta	espTarjeta	espTarjeta
espClave	bloqueo	espOpcion	bloqueo	bloqueo	darTarjeta	espClave	espTarjeta
espOpcion	bloqueo	bloqueo	darDinero	darTicket	darTarjeta	espOpcion	espTarjeta
darDinero	bloqueo	bloqueo	darDinero	darTicket	darTarjeta	darDinero	bloqueo
darTicket	bloqueo	bloqueo	darDinero	darTicket	darTarjeta	darTicket	bloqueo
darTarjeta	bloqueo	bloqueo	bloqueo	bloqueo	darTarjeta	darTarjeta	espTarjeta
bloqueo	bloqueo	bloqueo	bloqueo	bloqueo	bloqueo	espTarjeta	espTarjeta

Cuadro 1. Tabla de transición

	Salidas		
<b>Estados</b> ↓	dinero	tarjeta	comprobante
espTarjeta	false	false	false
espClave	false	false	false
espOpcion	false	false	false
darDinero	<b>true</b>	false	false
darTicket	false	false	<b>true</b>
darTarjeta	false	<b>true</b>	false
bloqueo	false	false	false

Cuadro 2. Tabla de salidas

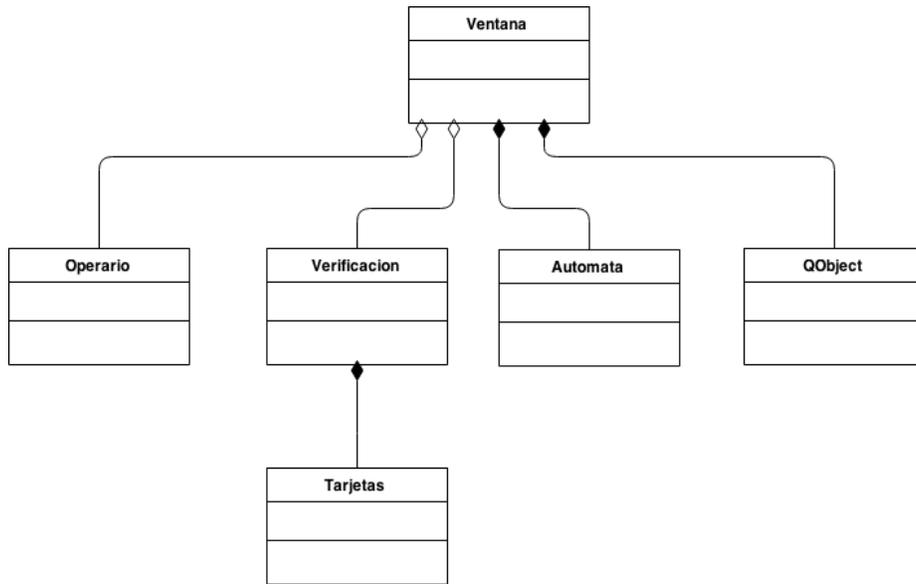
Los estados “darDinero” y “darTicket” se diferencian sólo por sus salidas, ya que las transiciones son iguales. Desde aquí se pueden ingresar las opciones consultar o retirar y el cajero va o permanece en el estado correspondiente. Si se sale se vuelve al estado inicial. Por cuestiones de seguridad ante cualquier otra entrada (excepto desbloquear) el autómata permanece en el estado de bloqueo.

En “darTarjeta”, ante entradas salir y desbloquear, se permanece en el mismo estado. Si la tarjeta se retira, se vuelve al estado inicial. En cualquier otro caso se bloquea. Por último, en el estado de “bloqueo”, si se desbloquea o se retira la tarjeta, se vuelve al estado inicial. Ante cualquier otra entrada, por seguridad, permanece bloqueado.

Esto se sintetiza en la tabla de transición  $\delta$  (Cuadro 1) y en la tabla de salidas  $F$  (Cuadro 2).

## 2.2. Lectura y actualización de cuentas

Uno de los requerimientos de un cajero automático es la capacidad de manejar información de cuentas bancarias. En este caso, eso se realiza mediante un archivo de texto que contiene el número de tarjeta, la clave numérica, el saldo de la cuenta y el último retiro realizado. Cuando se ejecuta el programa, este archivo de texto se lee y se guarda en una lista. Los cambios sobre el saldo de las cuentas se hacen sobre la lista y no sobre el archivo de texto, es recién al cerrar el programa que el archivo de texto se sobrescribe.



**Figura 2.** Asociaciones entre clases

### 2.3. Bloqueo y clave de operario

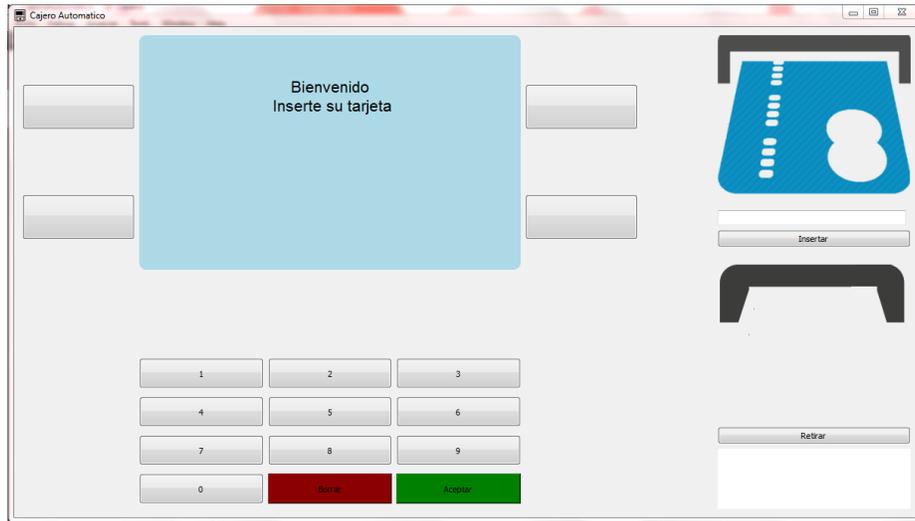
El cajero se bloquea por dos motivos. El primero es producto de la tabla de transición, explicado en la sección Automata, y se realiza con el objeto de proteger la cuenta del usuario. El segundo, es debido a un error en la lectura de archivos. El estado de desbloqueo se presenta al abrir el programa.

Para desbloquear el cajero es necesario introducir una “clave de operario”. Esta clave es numérica, y se encuentra en el archivo “operarios.txt”.

## 3. Implementación

Para la implementación se crearon cinco clases, y se utilizaron clases de Qt [Website de Qt], como QObject y las que heredan de ésta. Se utilizó Qt por resultar didáctico, tener implementados ejemplos variados, contar con varios foros de ayuda, ser gratuito, y fundamentalmente por poseer numerosas librerías para realizar interfaces gráficas.

El modelado del cajero se realiza en la clase Automata, y el de cada cuenta bancaria en Tarjetas . La representación de la red de cuentas bancarias y su verificación se hace en la clase Verificacion, que es una composición de Tarjetas. La clase Operario es la representación del empleado del banco. La interfaz gráfica se desarrolla en la clase Ventana. Ésta es una composición de Automata, QObject y otros objetos de Qt y una agregación de Verificacion y Operario.



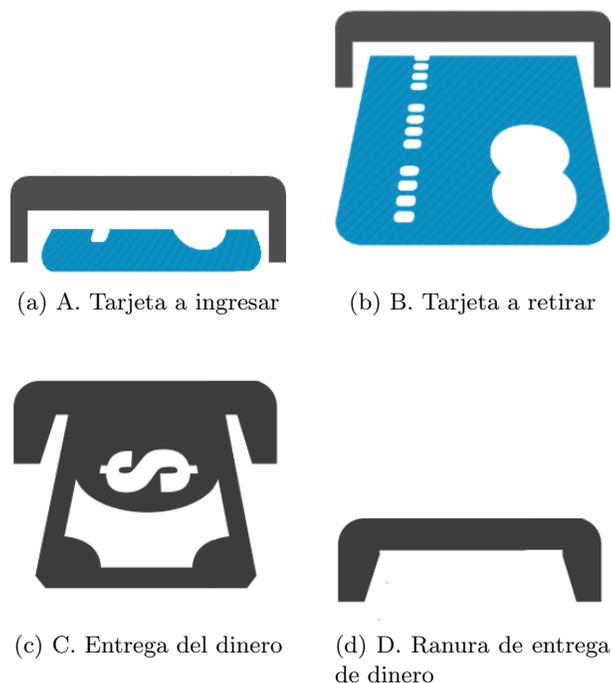
**Figura 3.** Ventana de la aplicación

#### 4. Interfaz gráfica

La interfaz gráfica se desarrolla en la clase “Ventana”. La Figura 3 es una captura de pantalla de la ventana al ejecutar el programa. La ventana propiamente dicha cuenta con una “pantalla”, en la que se muestran las opciones y se ven los datos ingresados (como la clave). A ambos lados de ésta tiene un par de botones cuya finalidad es seleccionar las opciones de la pantalla. También posee un teclado numérico con dígitos del 0 al 9, un botón de borrar y un botón de aceptar. Del lado derecho de la ventana, hay una imagen de una tarjeta que representa la ranura donde se inserta la misma en un cajero real. Esta imagen se cambia dependiendo del estado del cajero (de la figura 4 A a la 4 B). Debajo hay una línea de texto editable para ingresar el número de tarjeta y un botón Insertar/Retirar (dependiendo del estado). Cuenta con otra imagen que cambia para simular la entrega del dinero (de la figura 4 C a 4 D) y un botón Retirar. Por último, tiene un rectángulo en blanco donde se imprimen los datos de la cuenta al ser requeridos, a modo de comprobante.

Se utilizan las librerías de Qt y sus componentes gráficos.

Un aspecto a destacar es que se utilizan *Signals* y *Slots* para la comunicación entre objetos, características de esta biblioteca. Una señal es emitida cuando un evento de la interfaz gráfica es disparado. Un componente emite una señal que puede ser capturada por otro cuyo slot o zócalo ha sido configurado para recibirla.



**Figura 4.** Iconos de tarjeta y dinero

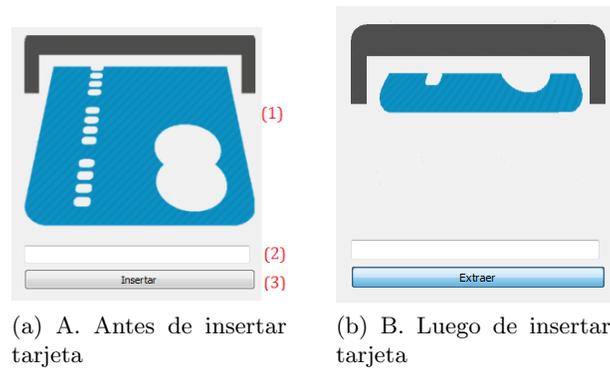
## 5. Manual de uso para el usuario

La figura 3 es la ventana al iniciar el programa. Ésta es la vista frontal del cajero automático. En la esquina superior derecha (figura 5) A se ve la imagen de una tarjeta (1), una línea de texto editable (2) y un botón con el texto Insertar (3). Se debe ingresar el número de la tarjeta en la línea de texto debajo de la tarjeta, mediante el teclado de la computadora. Si el número de tarjeta es correcto, la esquina cambia su ícono a lo que se ve en la figura 5 B.

Luego, mediante el teclado numérico (ver figura 6), se ingresa la clave correspondiente a la tarjeta. La clave se mostrará en la pantalla, y si se ingresa por equivocación un número erróneo, puede borrarse con el botón Borrar, del teclado numérico. Una vez que se ingresó la clave completa, se presiona Aceptar en el teclado numérico.

Si la clave es incorrecta, se va a pedir que se ingrese nuevamente. Si es correcta, en la pantalla (figura 7) se presentan posibles operaciones: retirar dinero (1), consultar saldo (2) o salir (3). Para seleccionar una opción se presiona el botón que se encuentra junto a la opción (en la figura 7, (a), (b) o (c)).

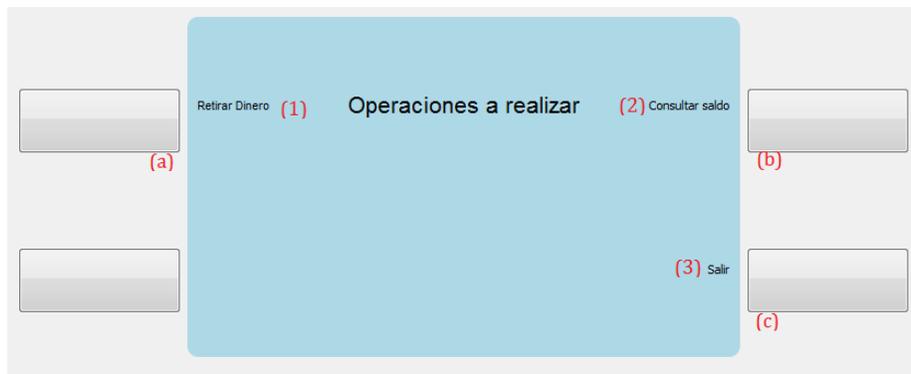
Si se selecciona la opción Consultar saldo, en la esquina inferior derecha, en el rectángulo del comprobante se imprime el saldo de la cuenta y el último retiro.



**Figura 5.** Esquina superior derecha de la pantalla



**Figura 6.** Teclado Numérico



**Figura 7.** Pantalla del cajero con las operaciones posibles

En la pantalla dice Retire su comprobante se siguen mostrando las operaciones posibles. Si se selecciona la opción Retirar Dinero, en pantalla dice “Ingrese el monto a retirar. Debe ser múltiplo de 10”. Mediante el teclado numérico se debe ingresar el monto a retirar.

Luego se presiona aceptar mediante el teclado numérico. Si no hay errores, en pantalla dice Retire el Dinero y en la derecha de la pantalla se muestra una imagen de dinero (figura 4 C). Para retirarlo y continuar

con las operaciones, se debe presionar el botón Retirar que se encuentra debajo de la imagen, y se restaura la imagen original (figura 4 D).

Pueden ocurrir dos errores. El primero, es que el monto a retirar no sea múltiplo de 10, por lo que se vuelve al menú de opciones. El segundo, que el monto a retirar sea mayor que el saldo disponible, por lo que se vuelve al menú de opciones.

Finalmente, si se presiona el botón Salir, en pantalla dice “Retire su tarjeta”. Una vez que se retiró la tarjeta, con el botón extraer, se vuelve a la pantalla inicial (figura 3).

## 6. Conclusiones

Se han cumplido los objetivos planteados en el trabajo. Sin embargo, hay mejoras que pueden realizarse, que no se han hecho por limitación de tiempo. Como se ha visto anteriormente, el cajero sólo permite retirar dinero. Una posible mejora a futuro sería implementar un mecanismo de depósito de dinero. Además, sólo se permiten operaciones sobre un tipo de cuenta y una moneda predeterminadas. Podría agregarse, por ejemplo, la opción de operar con caja de ahorro o cuenta corriente, en pesos o en dólares.

Un aspecto destacable es que en el presente trabajo se pusieron en práctica la mayoría de los temas de la materia: funciones, punteros, listas, clases, objetos, relaciones, interfaz gráfica de Qt y autómatas. Es por ello que la realización del presente proyecto resultó una actividad integradora y enriquecedora en cuanto a los contenidos de la materia.

## 7. Referencias

- COPPO, RICARDO (2014) *Apuntes de la cátedra Principios de Computadoras II, 2ª cuatrimestre 2014*. Universidad Nacional del Sur.
- Website de Qt(online), [www.qt.io](http://www.qt.io), consultado durante febrero 2015.

## 8. Bibliografía

- Code Progress (online), [www.codeprogress.com](http://www.codeprogress.com), consultado durante febrero 2015.
- Documentación de Qt(online), [www.doc.qt.io](http://www.doc.qt.io), consultado durante febrero 2015.
- Ejemplos de Qt: calculadora. Descargado desde Qt creator.
- General discussion of Qt (online), [www.comments.gmane.org](http://www.comments.gmane.org), consultado durante febrero 2015.
- Programming Examples, [www.programingexamples.net](http://www.programingexamples.net), consultado durante febrero 2015.
- Proyecto Qt (online), [www.qt-project.org](http://www.qt-project.org), consultado durante febrero 2015.
- Qt Centre (online), [www.qtcentre.org](http://www.qtcentre.org), consultado durante febrero 2015.
- Stack Overflow (online), [www.stackoverflow.com](http://www.stackoverflow.com), consultado durante febrero 2015.