Universidad Nacional de Rosario

Experimento Empírico

Especificación formal

Autor: Stizza, Federico

Director: Cristiá, Maximiliano

Departamento de Ciencias de la Computación

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Av. Pellegrini 250, Rosario, Santa Fe, Argentina

20 de Noviembre de 2022

1. Modalidad y plazo de entrega

En caso de utilizar *Replit* como entorno de desarrollo la solución deberá ser entregada por ese medio.

Caso contrario, por favor comprimir la solución en un archivo con su nombre, por ejemplo **Stizza-Federico.zip** y envíelo a mi dirección de correo electrónico: *federico.stizza@gmail.com*.

El plazo de entrega es de 1 semana desde que recibió este documento. Se estima que como máximo necesitará cuatro horas para resolver el problema.

Le solicito que contabilice el tiempo neto que utilizó para resolver el problema ya que en la encuesta de cierre habrá una pregunta relacionada.

2. Introducción

Deberá implementar **seis** operaciones básicas de un cajero automático.

Junto con este documento se le entregarán dos archivos que le servirán de base para empezar la solución.

Dichos archivos varían según el lenguaje que haya seleccionado y son:

1. Estado.cs (C#), estado.js (JavaScript) y estado.py (Python).
2. Programa.cs (C#), index.js (JavaScript) y main.py (Python).

Los archivos del primer punto contienen una clase llamada **Estado** que deberán completar de acuerdo a lo solicitado en la especificación. La misma provee una interfaz para guardar y cargar el estado en un archivo.

Los del segundo punto corresponden al punto de entrada del sistema, el cuál reúne todas las operaciones que debe implementar y se ejecuta de la siguiente manera:

C#	dotnet run –project Solucion.csproj operacion arg1 arg2 ...
JavaScript	node index.js operacion arg1 arg2 ...
Python	python3 main.py operacion arg1 arg2 ...

3. Consideraciones generales

- Le solicito que contabilice el tiempo neto que utilizó para resolver el problema.
- Las operaciones se deben poder ejecutar secuencialmente y deben persistir los cambios de estado según corresponda.
- No es necesario que realice una traducción textual de la especificación. Este documento debe servirle para entender cuál es el funcionamiento esperado de cada operación.
- Si le resulta útil para realizar pruebas puede modificar manualmente el estado que se encuentra en el archivo **estado.json**.
- Puede incluir en su solución librerías de terceros o paquetes que crea necesario para resolver el requerimiento.

- El programa que corresponde al punto de entrada de la solución no debe ser modificado, debe mantener los nombres de las operaciones y el orden de los argumentos según se indica en la siguiente tabla.
- Aquellas operaciones que *retornan* algo deben imprimirlo en pantalla utilizando la sálida estándar.

Operación	**Invocación**
Extracción	extraccion dni clave monto
Cambio de Clave	clave dni actual nueva
Consulta de Saldo	saldo dni clave
Alta de Usuario	alta dni_administrador clave_administrador dni clave nombre sueldo saldo
Carga de Cajero	carga dni_administrador clave_administrador monto
Consulta de movimientos	movimientos dni_administrador clave_administrador dni_consulta desde hasta

4. Especificación formal

4.1. Tipos abstractos

abstract type DNI

abstract type $NOMBRE$

abstract type $CLAVE$

abstract type $FECHAHORA$

4.2. Tipos enumerados

$$\textbf{enum}\ OPERACION = extraccion \mid clave$$

$$\textbf{enum}\ RESULTADO = ok \mid usuarioInexistente \mid usuarioYaExistente \mid claveIncorrecta \mid saldoCajeroInsuficiente \mid saldoInsuficiente \mid noCumplePoliticaExtraccion \mid noCumplePoliticaExtraccion2 \mid usuarioNoHabilitado \mid limiteUsuariosAlcanzado \mid cambioDeClaveBloqueado \mid noCumpleRequisitosClave1 \mid noCumpleRequisitosClave2$$

Sinónimos de tipo

$$\textbf{type}\ MOVIMIENTO = FECHAHORA \times OPERACION$$

$$\textbf{type}\ MONTO = \mathbb{N}$$

4.3. Constantes

$$\textbf{global}\ CANT\_MAX\_USUARIOS : \mathbb{N} = 300$$

$$\textbf{global}\ LONG\_MIN\_CLAVE : \mathbb{N} = 8$$

$$\textbf{global}\ administrador : DNI$$

$$\textbf{global}\ nombre\_administrador : NOMBRE$$

$$\textbf{global}\ clave\_administrador : CLAVE$$

$$\textbf{global}\ ahora : FECHAHORA$$

Donde,

- $administrador$ es el DNI del usuario administrador.
- $nombre\_administrador$ es el $NOMBRE$ del usuario administrador.
- $clave\_administrador$ es el $CLAVE$ del usuario administrador.
- $ahora$ es el instante de tiempo representado en $FECHAHORA$ en el que se está ejecutando la operación.

4.4. Funciones externas

external function $MISMO\_MES : FECHAHORA \to \mathbb{P}\, FECHAHORA$

Asigna a cada $FECHAHORA$ un conjunto de $FECHAHORA$ que corresponden al mismo mes y año del parámetro.

external function $MISMO\_DIA : FECHAHORA \to \mathbb{P}\, FECHAHORA$

Análogo al anterior, pero el conjunto de $FECHAHORA$ corresponden al mismo día, mes y año del parámetro.

external function $DIF\_FECHAS\_DIAS : (FECHAHORA \times FECHAHORA) \to \mathbb{Z}$

Asigna a cada par (f1,f2) la diferencia en días entre $f1 - f2$.

external function $LONGITUD : CLAVE \to \mathbb{N}$

Asigna a cada clave la cantidad de caracteres que la compone.

external function $CONTIENE\_LETRA\_NUM : \mathbb{P}\, CLAVE$

Es el conjunto de todas las posibles claves que son alfanuméricas.

Nota: no es necesario que se implementen funciones que cumplan estas definiciones explícitamente. Se pueden encontrar funciones que modelen lo que cada función externa busca representar.

4.5. Estado

El estado del sistema se modela de la siguiente manera:

StateVariables $::=$
$$usuarios : DNI \to NOMBRE$$
$$claves : DNI \to CLAVE$$
$$saldos : DNI \to MONTO$$
$$sueldos : DNI \to MONTO$$
$$movimientos : DNI \to (FECHAHORA \leftrightarrow OPERACION)$$
$$saldo : MONTO$$

4.5.1. Estado inicial

$$\begin{aligned}\textbf{InitialState} ::= & usuarios = \{(administrador, nombre\_administrador)\} \\ & claves = \{(administrador, clave\_administrador)\} \\ & saldos = \emptyset \\ & sueldos = \emptyset \\ & movimientos = \emptyset \\ & saldo = 0\end{aligned}$$

4.6. Operaciones

$$\begin{aligned}\textbf{op}\ & Extraccion(dni? : DNI, clave? : CLAVE, monto? : MONTO, res! : RESULTADO) \mathrel{\widehat{=}} \\ & ExtraccionOK(dni?, clave?, monto?, res!) \\ & \lor\ UsuarioInexistente(dni?, res!) \\ & \lor\ ClaveIncorrecta(dni?, clave?, res!) \\ & \lor\ NoCumplePoliticaExtraccion(dni?, res!) \\ & \lor\ NoCumplePoliticaExtraccion2(dni?, monto?, res!) \\ & \lor\ SaldoInsuficiente(dni?, monto?, res!) \\ & \lor\ SaldoCajeroInsuficiente(monto?, res!)\end{aligned}$$

Extrae el monto especificado de la cuenta del usuario autenticado.

$$\begin{aligned}\textbf{op}\ & CambioClave(dni? : DNI, clave? : CLAVE, nueva\_clave? : CLAVE, res! : RESULTADO) \mathrel{\widehat{=}} \\ & CambioClaveOK(dni?, clave?, nueva\_clave?, res!) \\ & \lor\ UsuarioInexistente(dni?, res!) \\ & \lor\ ClaveIncorrecta(dni?, clave?, res!) \\ & \lor\ CambioDeClaveBloqueado(dni?, res!) \\ & \lor\ NoCumpleRequisitosClave1(nueva\_clave?, res!) \\ & \lor\ NoCumpleRequisitosClave2(nueva\_clave?, res!)\end{aligned}$$

Asignar la nueva clave especificada al usuario autenticado.

op $ConsultaSaldo(dni? : DNI, clave? : CLAVE, saldo! : MONTO, res! : RESULTADO) \mathrel{\widehat=}$

$ConsultaSaldoOK(dni?, clave?, saldo!, res!)$

$\lor\ UsuarioInexistente(dni?, res!)$

$\lor\ ClaveIncorrecta(dni?, clave?, res!)$

Retorna el saldo de la cuenta del usuario autenticado.

op $AltaUsuario(dni\_administrador? : DNI, clave\_administrador? : CLAVE,$

$dni? : DNI, clave? : CLAVE, nombre? : NOMBRE,$

$sueldo? : MONTO, res! : RESULTADO) \mathrel{\widehat=}$

$AltaUsuarioOK(dni\_administrador?, clave\_administrador?, dni?, clave?, nombre?, sueldo?, res!)$

$\lor\ UsuarioNoHabilitado(dni\_administrador?, res!)$

$\lor\ ClaveIncorrecta(dni\_administrador?, clave\_administrador?, res!)$

$\lor\ UsuarioYaExistente(dni?, res!)$

$\lor\ LimiteUsuariosAlcanzado(res!)$

Da de alta un nuevo usuario en el cajero.

op $Carga(dni? : DNI, clave? : CLAVE, saldo? : MONTO, res! : RESULTADO) \mathrel{\widehat=}$

$CargaOK(dni?, clave?, saldo?, res!)$

$\lor\ UsuarioNoHabilitado(dni?, res!)$

$\lor\ ClaveIncorrecta(dni?, clave?, res!)$

Acredita el monto especificado al cajero.

op $ConsultaMovimientos(dni? : DNI, clave? : CLAVE, dni\_consulta? : DNI, desde? : FECHAHORA,$

$hasta? : FECHAHORA, movimientos! : \mathbb{P}\, MOVIMIENTO, res! : RESULTADO) \mathrel{\widehat=}$

$ConsultaMovimientosOK(dni?, clave?, dni\_consulta?, desde?, hasta?, movimientos!, res!)$

$\lor\ UsuarioNoHabilitado(dni?, res!)$

$\lor\ UsuarioInexistente(dni\_consulta?, res!)$

$\lor\ ClaveIncorrecta(dni?, clave?, res!)$

Retorna los movimientos realizados por el usuario en el rango de fechas especificado.

4.7. Suboperaciones

sub $ConsultaSaldoOK(dni? : DNI, clave? : CLAVE, saldo! : MONTO, res! : RESULTADO) \mathrel{\widehat=}$

$dni? \in \mathrm{dom}\, usuarios \land$

$claves(dni?) = clave? \land$

$saldo! := saldos(dni?) \land$

$res! := ok$

sub $ExtraccionOK(dni? : DNI, clave? : CLAVE, monto? : MONTO, res! : RESULTADO) \mathrel{\widehat=}$

$dni? \in \mathrm{dom}\, usuarios \land$

$claves(dni?) = clave? \land$

$\#((MISMO\_DIA(ahora)) \lhd (movimientos(dni?) \rhd \{extraccion\})) \leq 2 \land$

$monto? \leq sueldos(dni?)\ \mathsf{div}\ 2 \land$

$monto? \leq saldo \land$

$saldo := saldo - monto? \land$

$saldos := saldos \oplus (dni?, saldos(dni?) - monto?) \land$

$movimientos := movimientos \oplus (dni?, movimientos(dni?) \cup \{(ahora, extraccion)\}) \land$

$res! = ok$

sub $CambioClaveOK(dni? : DNI, clave? : CLAVE, nueva\_clave? : CLAVE, res! : RESULTADO) \mathrel{\widehat=}$

$dni? \in \mathrm{dom}\, usuarios \land$

$claves(dni?) = clave? \land$

$LONGITUD(nueva\_clave?) \geq LONG\_MIN\_CLAVE \land$

$nueva\_clave? \in CONTIENE\_LETRA\_NUM \land$

$((MISMO\_MES(ahora)) \lhd (movimientos(dni?) \rhd \{clave\})) = \emptyset \land$

$claves := claves \oplus (dni?, nueva\_clave?) \land$

$movimientos := movimientos \oplus (dni?, movimientos(dni?) \cup \{(ahora, clave)\}) \land$

$res! := ok$

sub $ConsultaMovimientosOK(dni? : DNI, clave? : CLAVE, dni\_consulta? : DNI, desde? : FECHAHORA,$
$hasta? : FECHAHORA, movimientos! : \mathbb{P}\, MOVIMIENTO, res! : RESULTADO) \mathrel{\widehat=}$
$dni? = administrador \land$
$claves(dni?) = clave? \land$
$dni\_consulta? \in \operatorname{dom} usuarios \land$
$movimientos! := \{m : MOVIMIENTO \mid m \in movimientos(dni\_consulta?) \land$
$\land\ DIF\_FECHAS\_DIAS(m.1, desde?) \geq 0 \land$
$\land\ DIF\_FECHAS\_DIAS(hasta?, m.1) \geq 0\} \land$
$res! := ok$

sub $AltaUsuarioOK(dni\_administrador? : DNI, clave\_administrador? : CLAVE, dni? : DNI, clave? : CLAVE,$
$nombre? : NOMBRE, sueldo? : MONTO, res! : RESULTADO) \mathrel{\widehat=}$
$dni\_administrador? = administrador \land$
$claves(dni\_administrador?) = clave? \land$
$dni? \notin \operatorname{dom} usuarios \land$
$\#(\operatorname{dom} usuarios) < 300 \land$
$movimientos := movimientos \cup \{(dni?, \emptyset)\} \land$
$usuarios := usuarios \cup \{(dni?, nombre?)\} \land$
$claves := claves \cup \{(dni?, clave?)\} \land$
$saldos := saldos \cup \{(dni?, sueldo?)\} \land$
$sueldos := sueldos \cup \{(dni?, sueldo?)\} \land$
$res! := ok$

sub $CargaOK(dni\_administrador? : DNI, clave\_administrador? : CLAVE, saldo? : MONTO, res! : RESULTADO) \mathrel{\widehat=}$
$dni\_administrador? = administrador \land$
$claves(dni\_administrador?) = clave\_administrador? \land$
$saldo := saldo + saldo? \land$
$res! := ok$

4.8. Errores

$$\textbf{er}\ UsuarioInexistente(dni? : DNI, res! : RESULTADO) \mathrel{\widehat{=}}$$
$$dni? \notin \mathrm{dom}\ usuarios \land$$
$$res! := usuarioInexistente$$

$$\textbf{er}\ UsuarioYaExistente(dni? : DNI, res! : RESULTADO) \mathrel{\widehat{=}}$$
$$dni? \in \mathrm{dom}\ usuarios \land$$
$$res! := usuarioYaExistente$$

$$\textbf{er}\ ClaveIncorrecta(dni? : DNI, clave? : CLAVE, res! : RESULTADO) \mathrel{\widehat{=}}$$
$$dni? \in \mathrm{dom}\ claves \land$$
$$claves(dni?) \neq clave? \land$$
$$res! := claveIncorrecta$$

$$\textbf{er}\ SaldoCajeroInsuficiente(monto? : MONTO, res! : RESULTADO) \mathrel{\widehat{=}}$$
$$monto? > saldo \land$$
$$res! := saldoCajeroInsuficiente$$

$$\textbf{er}\ SaldoInsuficiente(dni? : DNI, monto? : MONTO, res! : RESULTADO) \mathrel{\widehat{=}}$$
$$dni? \in \mathrm{dom}\ saldos \land$$
$$monto? > saldos(dni?) \land$$
$$res! := saldoInsuficiente$$

$$\textbf{er}\ NoCumplePoliticaExtraccion(dni? : DNI, res! : RESULTADO) \mathrel{\widehat{=}}$$
$$dni? \in \mathrm{dom}\ movimientos \land$$
$$\#(MISMO\_DIA(ahora) \lhd (movimientos(dni?) \rhd \{extraccion\})) > 2 \land$$
$$res! := noCumplePoliticaExtraccion$$

$$\textbf{er}\ NoCumplePoliticaExtraccion2(dni? : DNI, monto? : MONTO, res! : RESULTADO) \mathrel{\widehat=}$$
$$dni? \in \operatorname{dom} sueldos \land$$
$$monto? > sueldos(dni?)\ \mathsf{div}\ 2 \land$$
$$res! := noCumplePoliticaExtraccion2$$

$$\textbf{er}\ LimiteUsuariosAlcanzado(res! : RESULTADO) \mathrel{\widehat=}$$
$$\#(\operatorname{dom} usuarios) \geq 300 \land$$
$$res! := limiteUsuariosAlcanzado$$

$$\textbf{er}\ CambioDeClaveBloqueado(dni? : DNI, res! : RESULTADO) \mathrel{\widehat=}$$
$$dni? \in \operatorname{dom} movimientos \land$$
$$((MISMO\_MES(ahora)) \lhd (movimientos(dni?) \rhd \{clave\})) \neq \emptyset \land$$
$$res! := cambioDeClaveBloqueado$$

$$\textbf{er}\ UsuarioNoHabilitado(dni? : DNI, res! : RESULTADO) \mathrel{\widehat=}$$
$$dni? \neq administrador \land$$
$$res! := usuarioNoHabilitado$$

$$\textbf{er}\ NoCumpleRequisitosClave1(clave? : CLAVE, res! : RESULTADO) \mathrel{\widehat=}$$
$$LONGITUD(clave?) < LONG\_MIN\_CLAVE \land$$
$$res! := noCumpleRequisitosClave1$$

$$\textbf{er}\ NoCumpleRequisitosClave2(clave? : CLAVE, res! : RESULTADO) \mathrel{\widehat=}$$
$$clave? \notin CONTIENE\_LETRA\_NUM \land$$
$$res! := noCumpleRequisitosClave2$$