

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México
Escuela de ingeniería y Ciencias

Proyecto integrador
Carlos Josue Ceballos Arciniega A01752195
Alessandra Hernández Bolaños A01745363
Profesor: Jorge Adolfo Ramírez Uresti
Grupo 302

1 de diciembre 2020

Proyecto integrador

Introducción

En estos tiempos de pandemia como ya leímos en la situación problema, el comercio en línea está creciendo de manera exponencial, ocasionando a su vez el aumento en el registro de tarjetas de crédito y débito en páginas web para realizar compras. Uno de los mayores problemas que se presenta al realizar este tipo de transacciones es la clonación de dichas tarjetas debido a que el sistema de seguridad en las páginas web es muy ineficiente, creando incertidumbre en las personas. El objetivo de este proyecto es generar una mayor confiabilidad y alcance a nuevos usuarios.

El propósito que tiene nuestro proyecto es la creación de un sistema o app que sea capaz de reducir el fraude con un sistema de seguridad mejor implementado, creando formas de autenticación que no todos son capaces de generar, ya sea pines de único uso, así como correos y mensajes de confirmación y/o cancelación. Con este sistema, se evitaría el registro de tarjetas en muchas páginas, generando el pago por ti y al mismo tiempo registrando cada movimiento y notificando en tiempo real para una mayor seguridad.

Etapas de programación en equipo:

- i. Liga de GitHub: <https://github.com/CarlosCeballos19/POO-INTEGRADOR>
- ii. Diagrama UML de la etapa de Programación en Equipo. (Se encuentra en la última hoja de este trabajo)
- iii. Descripción de la funcionalidad del proyecto (introducción) y cada una de las clases.

Clases:

- Usuario: Dentro de esta clase el usuario es capaz de crear su cuenta introduciendo su datos personales como lo son su nombre, correo electrónico, teléfono y una contraseña.
- Tarjeta: Le pide los datos básicos de una tarjeta (nombre, número de tarjeta, CVV, Fecha de vencimiento).
- Transacción: Dentro de esta clase, el usuario crea y realiza una determinada transacción de las 3 que son posibles en el programa, introduce el monto y un nip de un solo uso que es generado en otro método.

1. Describe con detalle los 3 métodos que añadiste por cada clase y cómo esto le da una funcionalidad a los objetos creados mediante dicha clase.

Métodos de la clase Usuario:

- pedirNombre (string nombre): En este método se recibe el nombre del usuario, se renombra la variable “nombre” y funciona al momento de pedir el nombre del usuario en algún otro método.
- pedirCorreo (string correo): En este método se recibe el correo del usuario, se renombra la variable “correo” y funciona al momento de pedir el correo del usuario en algún otro método.
- pedirTeléfono(string telefono): En este método se recibe teléfono del usuario, se renombra la variable “telefono” y funciona al momento de pedir el número de tarjeta del usuario en algún otro método.
- pedirContraseña(string contraseña): En este método se recibe la contraseña del usuario de la página, se renombra la variable “contraseña” y funciona al momento de pedir la contraseña del usuario en algún otro método.

Métodos de la clase Tarjeta:

- pedirNombre(string nombre): En este método se recibe el nombre de la tarjeta, se renombra la variable “nombre” y funciona al momento de pedir el nombre de tarjeta del usuario en algún otro método.
- pedirNumTarjeta(string numTarjeta): En este método se recibe el número de tarjeta, se renombra la variable “numTarjeta” y funciona al momento de pedir el número de tarjeta del usuario en algún otro método.
- pedirFechaVenc(double fechaVencimiento): En este método se recibe la fecha de vencimiento de la tarjeta del usuario, se renombra la variable “fechaVencimiento” y funciona al momento de la fecha de vencimiento del usuario en algún otro método.
- pedirCvv(int cvv): En este método se recibe el cvv de la tarjeta, se renombra la variable “numTarjeta” y funciona al momento de pedir el número de tarjeta del usuario en algún otro método.

Métodos de la clase Transacción:

- pedirNip(int nip, int np1): Este método asegura que el usuario haya introducido de manera correcta el nip que se le fue otorgado con anterioridad. Si introduce uno que

es incorrecto le aparecerá un mensaje de error y se detendrá el programa. Si colocó el mismo que se le fue otorgado el programa continúa.

- movimientoRealizado(string Realizado): Este método le permite al usuario decidir qué movimiento quiere realizar.
- pedirMonto(int monto): Este método le permite establecer el monto que el usuario desee.

iv. Al menos 3 corridas (screenshot) de las pruebas que demuestran que tu proyecto integrador funciona

Primera prueba:

```
Nombre (use guion bajo como espacio): Santander
Numero de tarjeta: 5474459999999999
Fecha de vencimiento (formato mes.ano): 07.21
CVV: 734

--- NIP DE SEGURIDAD
(un solo uso)

Mensaje de texto: Estimado cliente, este es su NIP de seguridad: 4222

--- REALICE SU TRANSACCION

Se le envio un NIP via SMS
Digite el NIP que le fue enviado: 4222
Movimientos que se pueden realizar:
(a) Deposito
(b) Pago
(c) Retiro
Que movimiento quiere realizar?: a
Ingrese el monto: 126

--- NOTIFICACION DE MOVIMIENTO:
- Estimado(a) Paola_Danae_Lopez_Perez:
Hemos detectado el siguiente movimiento: Deposito, por un monto de $126 realizado con su tarjeta con número 5474459999999999
-For favor, teclee 1 si ha sido usted e ignore el mensaje, en caso que no, teclee 0: 1

Gracias por confirmar
La notificacion ha sido enviada al siguiente correo: pdanae.09hotmail.com

--- GENERACION DE TICKET
RESUMEN DE TRANSACCION PREVIAMENTE AUTORIZADA
Ticket No. 1
Nombre: Paola_Danae_Lopez_Perez
Num. Tarjeta: 5474459999999999
Transaccion realizada: Deposito
```

Segunda prueba:

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main

***** BIENVENIDO *****

--- CREE SU CUENTA

Nombre (use guion bajo como espacio): Evangeline_Peralta
Evangeline_Peralta
Correo: KagamineBLgmail.com
Telefono: 5534274756
Contrasena: wer

--- AHORA, INGRESE LOS DATOS DE SU TARJETA

Nombre (use guion bajo como espacio): Evangeline_Peralta
Numero de tarjeta: 12345678911
Fecha de vencimiento (formato mes.ano): 12.21
CVV: 123

--- NIP DE SEGURIDAD
(un solo uso)

Mensaje de texto: Estimado cliente, este es su NIP de seguridad: 8242

--- REALICE SU TRANSACCION

Se le envio un NIP via SMS
Digite el NIP que le fue enviado: 1234

--- NIP INCORRECTO, TRANSACCION NO AUTORIZADA
exit status 1
```

Tercer prueba:

```
Nombre (use guion bajo como espacio): Carlos
Correo: cj@gmail.com
Telefono: 5687412589
Contraseña: cjc0290

--- AHORA, INGRESE LOS DATOS DE SU TARJETA

Nombre (use guion bajo como espacio): Carlos
Numero de tarjeta: 555747784
Fecha de vencimiento (formato mes.ano): 02.26
CVV: 458

--- NIP DE SEGURIDAD
(un solo uso)

Mensaje de texto: Estimado cliente, este es su NIP de seguridad: 8901

--- REALICE SU TRANSACCION

Se le envio un NIP via SMS
Digite el NIP que le fue enviado: 8901
Movimientos que se pueden realizar:
(a) Deposito
(b) Pago
(c) Retiro
Que movimiento quiere realizar?: c
Ingrese el monto: 4587

--- NOTIFICACION DE MOVIMIENTO:
- Estimado(a) Carlos:
Hemos detectado el siguiente movimiento: Retiro, por un monto de $4587 realizado con su tarjeta con número 555747784
-Por favor, teclee 1 si ha sido usted e ignore el mensaje, en caso que no, teclee 0: 0

Movimiento cancelado, notifique a su proveedor de servicio
```

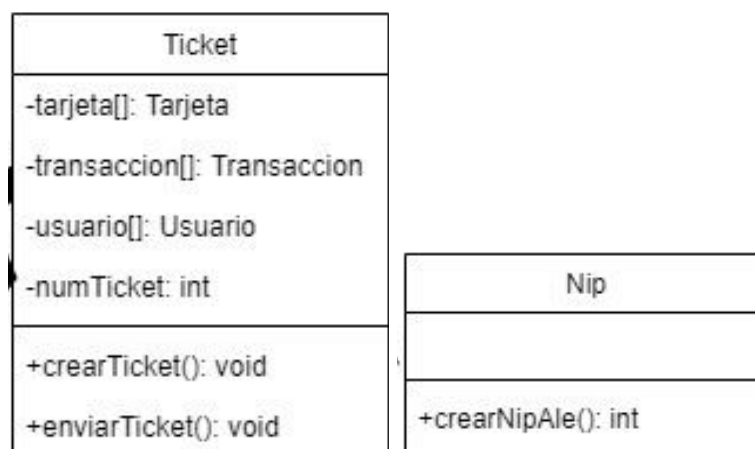
Etapas de programación individual:

Carlos Josue Ceballos Arciniega:

i. Liga a GitHub de la Etapa de Programación Individual

<https://github.com/CarlosCeballos19/POO-INTEGRADOR>

ii. Diagrama UML de la etapa de Programación Individual.



iii. Descripción de la funcionalidad del proyecto (tomando en cuenta la nueva clase que añadiste) y cada una de las clases.

- Ticket: Dentro de esta clase el sistema toma datos de las tres clases principales para crear un ticket, el cual será enviado al cliente en cuanto realice cualquier tipo de transacción, resumiendo lo más importante de dicho movimiento.
- Nip: Dentro de esta clase, únicamente se crea un NIP aleatorio que es enviado al cliente y debe confirmar con este para poder realizar su transacción sin problema alguno, es único y solo se puede usar una vez.

1. Describe con detalle los 3 métodos que añadiste por cada clase y cómo esto le da una funcionalidad a los objetos creados mediante dicha clase.

Métodos de la clase Ticket:

- crearTicket(Usuario u, Transacción t, Tarjeta tj): Este método genera el ticket tomando datos que fueron proporcionados y están dentro de la clase usuario, transacción y tarjeta, resumiendo el movimiento bancario realizado.
- enviarTicket(Usuario u): Este método genera un mensaje para notificar que el ticket será enviado a cierta dirección de correo.

Método de la clase Nip:

- crearNipAle(): Este método genera un número aleatorio de 4 dígitos que el usuario debe ingresar para poder continuar con su transacción.

iv. Al menos 3 corridas (screenshot) de las pruebas que demuestran que tu proyecto integrador funciona al añadir la nueva clase.

```
--- GENERACION DE TICKET
RESUMEN DE TRANSACCION PREVIAMENTE AUTORIZADA
Ticket No. 1
Nombre: Paola Danae Lopez Perez
Num. Tarjeta: 5474459999999999
Transaccion realizada: Deposito
```

```
--- GENERACION DE TICKET
RESUMEN DE TRANSACCION PREVIAMENTE AUTORIZADA
Ticket No. 1
Nombre: Carlos
Num. Tarjeta: 58466974
Transaccion realizada: Deposito
El ticket será enviado acjc@flao.com
```

```
--- NIP DE SEGURIDAD
(un solo uso)

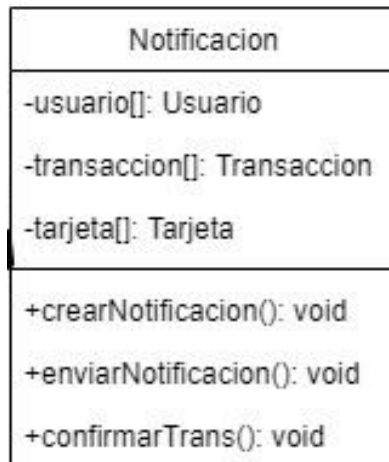
Mensaje de texto: Estimado cliente, este es su NIP de seguridad: 1256
```

Alessandra Hernández Bolaños

i. Liga a GitHub de la Etapa de Programación Individual

<https://github.com/CarlosCeballos19/POO-INTEGRADOR>

ii. Diagrama UML de la etapa de Programación Individual.



iii. Descripción de la funcionalidad del proyecto (tomando en cuenta la nueva clase que añadiste) y cada una de las clases.

1. Describe con detalle los 3 métodos que añadiste por cada clase y cómo esto le da una funcionalidad a los objetos creados mediante dicha clase.

Métodos de clase notificación: Esta clase genera una notificación al usuario cuando se realiza algún movimiento con su tarjeta.

- crearNotificacion(Usuario u, Transaccion t, Tarjeta tj): Este método crea la notificación tomando en cuenta los valores que introdujo el usuario con anterioridad. Para esto utiliza las clases: Usuario, Transaccion y Tarjeta. De estos objetos, obtenemos: el nombre, el movimiento, el monto y el número de tarjeta.
- enviarNotificacion(Usuario u); Este método crea un mensaje de confirmación que se envió la notificación generada en el otro método. Se utiliza la clase Usuario, para obtener el correo del usuario.
- confirmarTras(): Este método le permite al usuario confirmar el movimiento, tiene tres opciones, si, no y si introduce una opción inválida sale del programa.

iv. Al menos 3 corridas (screenshot) de las pruebas que demuestran que tu proyecto integrador funciona al añadir la nueva clase.

Primer corrida:

```
--- NOTIFICACION DE MOVIMIENTO:
- Estimado(a) Alessandra Hernandez:
Hemos detectado el siguiente movimiento: Deposito, por un monto de $34 realizado con su tarjeta con nHmero 341234134
-Por favor, teclee 1 si ha sido usted e ignore el mensaje, en caso que no, teclee 0: 1

Gracias por confirmar
La notificacion ha sido enviada al siguiente correo: ale@gmail.com
```

Segunda corrida:

```
--- NOTIFICACION DE MOVIMIENTO:
- Estimado(a) Pao:
Hemos detectado el siguiente movimiento: Deposito, por un monto de $21 realizado con su tarjeta con nHmero 452352523
-Por favor, teclee 1 si ha sido usted e ignore el mensaje, en caso que no, teclee 0: 0

Movimiento cancelado, notifique a su proveedor de servicio
```

Tercer corrida:

```
--- NOTIFICACION DE MOVIMIENTO:
- Estimado(a) Carlos:
Hemos detectado el siguiente movimiento: Deposito, por un monto de $45 realizado con su tarjeta con nHmero 454525252
-Por favor, teclee 1 si ha sido usted e ignore el mensaje, en caso que no, teclee 0: 6

Opci|n inv|lida, movimiento cancelado
PS C:\Ale\Laptop\Profesional IRS\Primer semestre\Programaci3n orientada a objetos\POO-INTEGRADOR>
```


d. Reflexión final individual: ¿Qué aprendiste? ¿Qué cosas se pueden mejorar? ¿Qué parte del proyecto te gusto más? ¿Qué parte del proyecto te gustó menos?

Carlos Josue Ceballos Arciniega:

En el desarrollo de este proyecto pude comprender y reafirmar algunos conceptos básicos sobre programación, así como un nuevo lenguaje de programación que es C++, el cual está más enfocado en los objetos, esto nos llevó a aprender a crear diagramas UML, establecer clases, establecer métodos, etc. De igual manera, durante su desarrollo encontramos muchas formas de completarlo y mejorarlo, creo que con más tiempo de clase podríamos encontrar muchas cosas más para mejorarlas. Lo que me gustó del proyecto fue todo el proceso de programación en conjunto con mi compañera, ya que entre risas todo fue más ameno y no se hizo tan pesado, lo que menos me gustó fue cuando se nos generaban problemas y no sabíamos por qué o los pequeños detalles que generan un error, sin embargo, fue algo que sin duda disfruté, comprobé que todos los días se aprenden cosas nuevas.

Alessandra Hernández Bolaños:

Durante este proyecto pude aprender los conceptos básicos de la programación orientada a objetos. Aprendí a crear clases a partir de un diagrama UML, incluso a que se pueden ir mejorando conforme se va avanzando en el proyecto. Gracias a este proyecto me fue posible reforzar lo visto en clase, cómo la creación e implementación de los métodos de get, set y otros métodos diferentes. Se pueden mejorar los métodos que tenemos, pero para que sea posible necesitamos de un mayor conocimiento de la programación orientada a objetos. Lo que más me gustó de este proyecto fue el hecho de que fuera colaborativo, porque mi compañero me ayudó a comprender de una mejor manera lo visto en clase e hizo que todo fuera más sencillo. Lo que menos me gustó fue todo el estrés que me generó con un problema en específico que teníamos, pero se solucionó.

Diagrama UML (completo):

