

Universidad Nororiental Privada “Gran Mariscal de Ayacucho”

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas e Informática

Asignatura: **PROGRAMACION III**

Periodo: 2-2024

**PROYECTO FINAL DE LA ASIGNATURA**

**Secciones 3D1**

**VALOR: 15%**

**Título:** Portal de Pagos (no es un sistema bancario)

**Escenario:**

Programa desarrollado en c++ que administre la información de un portal que maneja pagos, depósitos, tarjetas de crédito, cuentas y clientes a diversos comercios o personas.

La información que debe manejar el sistema es la siguiente, Debe tener las siguientes clases en c++, lo que se muestra son las clases con sus atributos:

1. Usuario: Código Usuario, clave, Código Cliente
2. Cliente: Código, Nombre, fecha nacimiento, Correo Electrónico, Teléfono Celular.
3. Transacción: Código Transacción, Descripción, monto, Número de cuenta, Número Tarjeta.
4. Cuenta: Número de cuenta, Descripción, Saldo Disponible, Código Cliente, Estado.

**Métodos:**

* Al ejecutar el código en c++, se debe ingresar usuario y clave para entrar al sistema, luego se debe verificar si el usuario es un cliente o el administrador.
* Debe existir solo un usuario que no esté enlazado a un cliente (el administrador), este usuario creará los clientes y les asignará un usuario y clave a los clientes nuevos. Este usuario se debe crear automáticamente la primera vez que se inicie el programa y también se guardará en el archivo de usuarios.
* Cuando el administrador crea un cliente, debe crear el usuario y la cuenta del cliente para que este pueda acceder al sistema.
* La clase usuario tendrá el usuario y la clave para poder ingresar al sistema, esta información se deberá guardar en un archivo de texto. El código del cliente será el código del cliente al que el usuario accederá al sistema. El código de cliente que se asigne al usuario debe ser un código de un cliente que existe.
* La clase cliente manejará la información de cada cliente, los clientes los almacenarán en archivos de texto, en donde el valor de cada variable esté separada por ; (punto y coma).
* La clase cuenta, maneja toda la información de la cuenta de un cliente, esta información se debe guardar en un archivo de texto en donde el valor de cada variable esté separado por ;. (punto y coma).
* La clase transacción es para registrar las transacciones del cliente. Las transacciones se deben guardar en un archivo de texto separado por ;.(punto y coma).
* El cliente dentro de su portal, podrá ver las transacciones de sus cuentas, podrá hacer depósitos a sus cuentas, retiros de sus cuentas, depósitos entre sus cuentas y las cuentas de otros.

**Otras indicaciones:**

• Debe validar el ingreso de datos, el programa no debe parar si se ingresa datos erróneos.

• El cliente podrá modificar la clave de ingreso al sistema cuando desee.

El cliente también puede actualizar toda la información de él y de su cuenta, menos los códigos del mismo.

• Cualquier otra consideración que usted considere importante para el proyecto, debe realizarlo.

• Al ingresar clientes, debe validar que la identidad y el correo sean únicos.

• Cuando realice depósitos a otras cuentas, debe antes de confirmar, aparecer el nombre del dueño de la cuenta, si no existe, no debe realizar el depósito.

• Un cliente puede tener muchas cuentas, pero cuando ingresa con usuario y clave, solo puede ver sus cuentas y sus transacciones, al mostrar sus transacciones debe seleccionar la cuenta que desea mostrar.

• Solo el administrador puede crear clientes, cuentas los usuarios y sus claves. • El administrador puede bloquear y desbloquear usuarios.

• Si un usuario está bloqueado, debe mostrar un mensaje al respecto y no podrá entrar al sistema, solo el administrador puede desbloquear el usuario.

• Para desbloquear su usuario se debe enviar un mensaje al administrador (programe la forma de notificación)

• Debe mostrar una interfaz amigable y acorde a la aplicación o portal de pagos, en modo gráfico.

**Consideraciones a tomar en cuenta:**

1. Deben usar clases y objetos, algoritmos de ordenación y búsqueda (estos serán vistos más adelante).
2. El programa debe incluir las librerías graficas del C++, grafics.h, winbgim.h no está permitido el uso de la librería smfl.
3. No está permitido el uso de la librería vector ni algorithm
4. Se debe presentar un informe técnico impreso con portada, introducción, planteamiento del problema, objetivo general y específicos, algoritmo general, solución estratégica aplicada, diagramas de clases en UML, conclusiones, debe ser entregado el **25-11-2024 (sin prorroga)**
5. Se debe entregar el código fuente y el ejecutable.
6. Interprete el enunciado para que pueda obtener los datos involucrados en el proceso, y pueda analizar los requerimientos del programa. Recuerde que un buen análisis y un buen diseño le permitirá desarrollar con éxito su aplicación.
7. La fecha de entrega es la **que indica el Plan de Evaluación de la asignatura establecido en el Classroom** No habrá prórroga. Todos deben entregar el proyecto ese día, equipo que no entregue proyecto, no será evaluado.
8. El equipo estará conformado por 2 personas.
9. Deben ser validadas todas las entradas y salidas del programa.
10. Debe ser defendido en el laboratorio de computación no estará permitido el uso de laptop ni computadores por parte del alumno, (la defensa forma parte fundamental de la evaluación y la selección será al azar. Estudiante que no exponga solo tendrá nota del informe.)
11. Se debe usar obligatoriamente Clases y Objetos, crear el archivo .h y el .cpp
12. No se evaluarán los códigos que sean iguales, parecidos o semejantes.
13. Si la aplicación no cubre el 95% de los requerimientos, su puntación será 3/20.
14. La evaluación está conformada de la siguiente manera 7% aplicación, 10% defensa del proyecto, 3% informe proyecto. La nota de la defensa es individual.
15. Tome sus previsiones, si requiere instalar alguna librería adicional, no se le dará tiempo para realizar tal actividad. Ud. debe probar y ejecutar su aplicación antes de la defensa.
16. Cualquier información adicional que se necesite puede ser consultada luego de las horas de clases. Considerar que los requerimientos serán solicitados al profesor de la asignatura y serán atendidos por grupo de proyecto.