

Utilizaremos el archivo **cliente.php** del campus.

1. Crear una clase Persona, una clase PYME y una clase Multinacional. Las 3 clases deben heredar de la clase Cliente.
2. Mover los atributos y métodos correspondientes (que sólo correspondan a personas físicas), de la clase Cliente a la clase Persona.
3. Agregar a las clases PYME y Multinacional los atributos y métodos que correspondan. (ej: CUIT, Razón social, etc). Agregar constructor y métodos getters y setters.
4. Realizar los cambios necesarios de modo tal que la clase Cliente pase a ser una clase abstracta.
5. Crear una clase abstracta Cuenta con atributos: CBU, Balance, Fecha de último movimiento. Agregar setters y getters para obtener los datos de la cuenta. Agregar un constructor que permite inicializar el CBU.
6. Agregar un método abstracto para **debitar** cierto monto que reciba como parámetro el monto y desde donde se está haciendo la transacción (cajero Banelco, cajero Link, caja). Agregar otro método (no abstracto) que permita **acreditar** cierto monto (y programar su comportamiento). (Tener en cuenta que cada método debe, además, modificar la fecha de último movimiento).
7. Crear 4 clases llamadas Classic, Gold, Platinum y Black. Todas “hijas” de la clase Cuenta.
8. Implementar los métodos debitar y acreditar en cada clase, con las siguientes reglas de negocio:
 - a. Debitar:
 - i. Classic: Si la transacción es desde un cajero Banelco, debe debitar un 5% mas. De Link un 10% mas. Desde caja no se agrega importe extra.
 - ii. Gold: Desde Banelco y Caja no se agrega importe extra. Desde Link un 5% mas.
 - iii. Platinum: 5% desde cualquier medio, a menos que la cuenta tenga un balance de 5.000 o más.
 - iv. Black: No se cobra importe extra.
 - b. Acreditar: Utiliza la cuenta que realiza su padre, pero debe modificar el importe a acreditar según el tipo de cuenta.
 - i. El banco se queda con un 5% si es Classic.
 - ii. Se queda con un 3% si es Gold, salvo que se esté acreditando 25.000 o más.
 - iii. Si es Platinum el banco no retiene nada.
 - iv. 0% para Black o 4% para montos superiores a 1.000.000.
9. Hacer los cambios necesarios en el Cliente, para que pueda tener una cuenta asociada. (Concepto de composición). Agregar la asignación de la cuenta en el constructor.
10. Una vez por mes, el banco debe cobrarle a cada cliente por los servicios prestados. Cada Cliente deberá tener un método que permita cobrar los servicios:
 - a. Si el cliente tiene cuenta Classic, el banco cobra un monto fijo de \$100.
 - b. Si el cliente tiene cuenta Gold, el banco cobra \$10 por cada letra del apellido del cliente o \$5 por cada letra de la razón social.

- c. Si el cliente tiene una cuenta Platinum el banco cobra un 10% del total del balance de la cuenta.
 - d. Si el cliente tiene una cuenta Black, el banco cobra \$500, más \$100 multiplicado por el número de día de la semana en la que se hizo la última transacción.
`$date = '2016/09/28';`
`$weekday = date('l', strtotime($date));`
11. Generar una interfaz “Liquidable” que declare un método **liquidarHaber**. Tanto PYME como Multinacional deben implementar “Liquidable”. El método **liquidarHaber** recibe una Persona y un monto en donde se le paga a la Persona dicho monto de la cuenta de la empresa.
- a. Dentro de PYME este método cobra un 1% de cada liquidación que desaparecen de la cuenta de la PYME.
 - b. Dentro de Multinacional este método cobra un valor fijo de \$500 que desaparecen de la cuenta de la Multinacional.
12. Crear una interfaz “Imprimible” que declare un método **mostrar**. Este método deberá retornar lo que corresponda, según la clase que lo implemente:
- a. Para Persona, el nombre y apellido.
 - b. Para PYME, la razón social
 - c. Para Multinacional, la razón social.
 - d. Para las cuentas, el balance.
13. Crear un archivo “banco.php”, que instancie un Banco y lo llene con Clientes (que ya deben tener cuentas a la hora de ser creados). Luego de eso, realizar las operaciones disponibles como prueba, imprimiendo para cada una sobre qué Cliente se hizo y como queda el balance de su Cuenta luego de la operación.
14. Teniendo los archivos de la Cuenta hechos con el docente:
- a. Crear un nuevo directorio llamado “cuentas” donde se situarán todos nuestros archivos de operaciones de las Cuentas(Classic, Gold, Platinum, Black)
 - b. Crear un nuevo directorio llamado “Clientes” donde se situarán todos nuestros archivos de operaciones de los Clientes(Cliente, Multinacional, Persona, Pyme)
 - c. Crear un nuevo directorio llamado “interfaz” donde se situarán todos nuestros archivos que sean interfaces.
 - d. **Utilizando namespaces**, corregir el código para que siga funcionando normalmente.

