Universidad Tecnológica Nacional UTNFra Facultad Regional Avellaneda										
Materia: PROGRAMACIÓN III										
Apellido:					**Fecha:**					
Nombre:					**Docente(2):**		**F. Lippi / P. Musella**			
División:					**Nota(2):**					
Legajo:					**Firma(2):**					
Instancia(1):	**PP**	**x**	**RPP**		**SP**		**RSP**		**FIN**	

Se debe realizar una aplicación para dar de ingreso con ***foto del usuario/cliente***.
Los datos se persistirán en archivos (ej. txt, json, csv, etc.)
Se deben respetar los nombres de los archivos y de las clases.
Se debe crear **una clase en PHP** por cada entidad y los archivos PHP solo deben llamar a métodos de las clases.

1- A- **index.php**:Recibe todas las peticiones que realiza el cliente (utilizaremos Postman), y administra a qué archivo se debe incluir.

B- **CuentaAlta.php**: (por POST) se ingresa Nombre y Apellido, Tipo Documento, Nro. Documento, Email, Tipo de Cuenta (CA – caja de ahorro o CC – cuenta corriente), Moneda ($ o U$S), Saldo Inicial (0 por defecto).
NOTA: los números de cuenta son únicos dentro del Banco, considerando para evaluarlos que la cuenta está compuesta por tipo de cuenta (CA o CC), nro. de cuenta y moneda.
Se guardan los datos en el archivo **banco.json,** tomando un id autoincremental de 6 dígitos como Nro. de Cuenta (emulado). Sí el número y tipo ya existen, se actualiza el saldo existente.
completar el alta con imagen/foto del usuario/cliente, guardando la imagen con Nro y Tipo de Cuenta (ej.: NNNNNNTT) como identificación en la carpeta: **/ImagenesDeCuentas/2023**.

2- ConsultarCuenta.php: (por POST) Se ingresa Tipo y Nro. de Cuenta, si coincide con algún registro del archivo **banco.json,** retornar la moneda/s y saldo de la cuenta/s. De lo contrario informar si no existe la combinación de nro y tipo de cuenta o, si existe el número y no el tipo para dicho número, el mensaje: “tipo de cuenta incorrecto”.

3-a- **DepositoCuenta.php**: (por POST) se recibe el Tipo de Cuenta, Nro de Cuenta y Moneda y el importe a depositar, si la cuenta existe en **banco.json,** se incrementa el

saldo existente según el importe depositado y se registra en el archivo **depósitos.json** la operación con los datos de la cuenta y el depósito (fecha, monto) e id autoincremental) .
Si la cuenta no existe, informar el error.

b- Completar el depósito con imagen del talón de depósito con el nombre: Tipo de Cuenta, Nro. de Cuenta e Id de Depósito, guardando la imagen en la carpeta **/ImagenesDeDepositos2023**.

4- ConsultaMovimientos.php: (por GET)
Datos a consultar:

a- El total depositado (monto) por tipo de cuenta y moneda en un día en particular (se envía por parámetro), si no se pasa fecha, se muestran las del día anterior.

b- El listado de depósitos para un usuario **en particular**.

c- El listado de depósitos entre dos fechas ordenado por **nombre**.

d- El listado de depósitos por **tipo de cuenta.**

e- El listado de depósitos por **moneda**.

5- ModificarCuenta.php (por PUT)
Debe recibir todos los datos propios de una cuenta (a excepción del saldo); si dicha cuenta existe (comparar por Tipo y Nro. de Cuenta) se modifica, de lo contrario informar que no existe esa cuenta.

6- RetiroCuenta.php: (por POST) se recibe el Tipo de Cuenta, Nro de Cuenta y Moneda y el importe a depositar, si la cuenta existe en **banco.json,** se decrementa el saldo existente según el importe extraído y se registra en el archivo **retiro.json** la operación con los datos de la cuenta y el depósito (fecha, monto) e id autoincremental.
Si la cuenta no existe o el saldo es inferior al monto a retirar, informar el tipo de error.

7- AjusteCuenta.php (por POST),
Se ingresa el número de extracción o depósito afectado al ajuste y el motivo del mismo. El número de extracción o depósito debe existir.
Guardar en el archivo **ajustes.json**
Actualiza en el saldo en el archivo **banco.json**

8-CuentaAlta.php:

a- La validación de cuenta existente deberá hacerse considerando los atributos: nro de cuenta y tipo de cuenta que deben ser únicos en el sistema.
b- Se debe adecuar la lógica desarrollada para permitir el ingreso por parámetro opcional de saldo, que permita definir un saldo inicial en el alta de la cuenta.
c- Se identificó una mejora funcional por la que el atributo “tipo de cuenta” contendrá la moneda siendo CA$ para caja de ahorro en pesos, CAU$S para caja de ahorro en dólares y CC$ y CCU$S para las mismas variantes de Cuenta Corriente

IMPORTANTE: verificar y adecuar todos los puntos funcionales donde estos cambios pueden impactar.

9- **BorrarCuenta.php** (por DELETE), debe recibir un número el tipo y número de cuenta y debe realizar la baja de la cuenta (soft-delete, no físicamente) y la foto relacionada a esa venta debe moverse a la carpeta **/ImagenesBackupCuentas/2023**.

10- ConsultaMovimientos.php: (por GET)
A la consulta de movimientos ya existente, incorporar:

a- El total retirado (monto) por tipo de cuenta y moneda en un día en particular (se envía por parámetro), si no se pasa fecha, se muestran las del día anterior.

b- El listado de retiros para un usuario **en particular**.

c- El listado de retiros entre dos fechas ordenado por **nombre**.

d- El listado de retiros por **tipo de cuenta.**

e- El listado de retiros por **moneda**.

f- El listado de **todas las operaciones** (depósitos y retiros) por **usuario**

Agregados y modificaciones a ejecutar en <u>2do parcial</u>:

- **A.** Se deben realizar todas las persistencias en BD utilizando PDO y sus métodos para dicho fin **(mandatorio)**

- **B.** Se debe realizar el enrutamiento de peticiones en el index utilizando Slim Framework 4 **(mandatorio)**

- **C.** Llevar el diseño de la solución al patrón MVC (Model View Controler)

Agregados y modificaciones a ejecutar durante el <u>2do parcial</u>:

- **D.** Crear la clase Usuarios en la que, al menos, deberá contar con los atributos: usuario (nombre de usuario o email), password y rol
- **E.** Se deberá validar la identidad de los usuarios y sólo podrán acceder a los recursos del sistema aquellos que estén identificados. Asimismo, sólo podrán acceder a los siguientes recursos, los roles que aquí se especifiquen:
    a. Ajustes -> rol supervisor
    b. Depósitos y Retiros -> rol cajero
    c. Consultas (de todo tipo) -> rol operador
- **F.** Se deberá generar un log de todos los accesos (no sólo login, sino cada vez que se intente consumir un recurso, independientemente de su resultado)
- **G.** Se deberá generar un log de transacciones en el que se registrarán los datos de fecha y hora, usuario y número de operación, luego de que la operación sea registrada.

Código obsoleto, copiado y pegado que no tenga utilidad (-1 punto).

Se pueden bajar templetes de internet o traer código hecho, pero en ningún caso se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.

Se deberá incluir una colección de Postman que contenga todas las peticiones para cada punto.

Las calificaciones deberían ser en función de los cambios solicitados sobre este enunciado.