



**UNIVERSIDAD TECNOLOGICA NACIONAL  
FACULTAD REGIONAL GENERAL PACHECO**

**TÉCNICO UNIVERSITARIO  
EN PROGRAMACION**

**LABORATORIO DE COMPUTACIÓN IV  
TP INTEGRADOR**

**NOTA:** Para realizar el trabajo práctico final deberá utilizar JAVA. Se deberán incluir los siguientes temas: JDBC / JSP / SERVLETS / SESSION

Se pide crear un sistema de gestión para un banco en donde existirán dos tipos de usuarios: administrador del banco y clientes.

### **Usuario banco**

- ABML de clientes. Tener en cuenta que a cada cliente se le asignará un usuario y contraseña para poder acceder a la web del banco. El administrador podrá cambiar la contraseña, pero nunca se podrá modificar el usuario de ese cliente.
- ABML de cuentas y asignación de cuenta a cliente. Una cuenta debe pertenecer a un solo cliente. Se le podrá asignar un máximo de 3 cuentas a un cliente. Al crear la cuenta se le asignará un monto inicial fijo de 10.000\$.
- Autorización de préstamos. El administrador podrá autorizar o rechazar los préstamos solicitados por el cliente. Si se aprueba el préstamo entonces se le asignará al cliente el monto pedido. Tener en cuenta que se deberán generar las cuotas para que luego pague el cliente.
- Incluir informe/ reportes estadísticos para el administrador.

### **Usuario cliente**

- El cliente podrá seleccionar una cuenta y observar el historial de los movimientos realizados en esa cuenta.
- Transferencias a otros clientes: Un cliente podrá transferir dinero entre sus propias cuentas o a cuentas de otros clientes

utilizando el CBU. Se podrá realizar la transferencia mientras el usuario cuente con dinero disponible. Si no posee dinero no podrá transferir.

- Pedido de préstamo. El cliente podrá pedir un préstamo de cierto dinero al banco, seleccionar la cantidad de cuotas en las que quiere abonarlo y elegir la cuenta en donde se le depositará el pago del préstamo. El banco recibirá el pedido y autorizará o no dicho préstamo.
- El cliente tendrá un menú de pago de préstamos en donde figure las cuotas que debe pagar, podrá pagar seleccionando la cuota y cuenta de donde se descontará dicho gasto. Cada vez que se pague una cuota se guardará la fecha en la cual se pagó la misma.
- El cliente podrá visualizar su información personal. No podrá cambiar los datos, solo los visualizará.

## **Observaciones**

Asumiremos que no se generan intereses por el pago fuera de término de la cuota de un préstamo, la cuota es fija. También asumiremos que toda la funcionalidad pertenece a un mismo banco (en la tabla usuario de la base de datos deberán tener un usuario de tipo administrador de banco para que pueda acceder a la aplicación). No es necesario hacer un ABML de tipo de cuentas posibles y tipos de movimientos, los mismos deben estar cargados previamente en la base de datos. Las cuentas no pueden quedar con saldos negativos.

## **Tipos de cuentas posibles:**

- Caja de ahorro.
- Cuenta corriente.

## **Tipos de movimientos posibles:**

- Alta de cuenta. El alta de una cuenta genera un movimiento de dinero positivo en la cuenta origen.
- Alta de un préstamo. El alta de un préstamo genera un movimiento de dinero positivo en la cuenta origen.
- Pago de préstamo. El pago de un préstamo, genera un movimiento de dinero negativo en la cuenta origen.
- Transferencia. Una transferencia genera dos movimientos, un movimiento negativo en la cuenta de origen (extracción de dinero) y un movimiento positivo en la cuenta destino (depósito de dinero)

## **En cuanto a los datos, mínimamente cada tabla tiene los siguientes datos:**

- Por cada movimiento se tiene una fecha, detalle o concepto, importe y tipo de movimiento.
- Un cliente tiene un DNI, CUIL, nombre, apellido, sexo, nacionalidad, fecha de nacimiento, dirección, localidad, provincia, correo electrónico, teléfonos, un usuario y contraseña para acceder a la página.
- Una cuenta tiene un cliente asignado, fecha de creación, tipo de cuenta, un número de cuenta, CBU y un saldo.
- Un préstamo tiene un cliente asignado, una fecha, el importe que deberá pagar el cliente (con intereses), el importe pedido por el cliente, un plazo de pago en meses, el monto que deberá pagar por mes y cuotas.

## **PARA PODER APROBAR EL TP INTEGRADOR DEBE CUMPLIR LAS SIGUIENTES PAUTAS:**

- Se debe participar en el **foro del tp integrador**. Los alumnos que no participen en el foro serán considerados que no participaron de la tarea y por ende desaprobaran la misma. También se evaluará el contenido que aporto cada alumno y a partir de esto se definirá la nota que obtendrá cada uno.
- Se pedirá trabajar con **GIT**. Cada alumno deberá tener su propio usuario y subir con su usuario el código que aporta al TP.
- Debe haber un **login**.
- En cada pantalla se podrá observar el **usuario logueado**.
- **Los listados deben estar paginados.**
- **Utilizar búsquedas y filtros** en los listados. Las búsquedas hacerlas generales utilizando el comparador LIKE %contenido% para traer todos los textos que cumplen con la condición... no solo el texto que sea idéntico. También trabajar con filtros, los filtros no son iguales a las búsquedas, los filtros permiten seleccionar opciones como, por ejemplo: en el listado de movimientos que te permita seleccionar los movimientos cuyo importe total este en un rango entre 1000 a 2000, o <100, >=100, etc.
- **Mensajes aclaratorios** “Se agrego correctamente en la base de datos”
- Trabajar con **programación en tres capas, utilizar interfaces y manejar las excepciones**.

Es necesario que el código se encuentre divididos en paquetes. Las excepciones estarán en un paquete separado y deberán crear al menos dos excepciones propias.

Acerca de las entidades, las clases deben estar correctamente diagramadas. Ejemplo: la clase movimiento debe tener un objeto de tipo movimiento en su interior, no un int ID\_movimiento ni tampoco string descripción\_movimiento.

**No tiene que haber desde los JSP, llamados al DAO ni al NEGOCIO... desde los JSP se tiene que llamar al SERVLET y el SERVLET llama al negocio. Por último, le negocio llama al DAO. El SERVLET es el “controlador” de nuestras aplicaciones web.**

- En los descolgables deben aparecer también las descripciones y no solo sus identificadores.
- Manejar bajas lógicas y no físicas.
- Trabajar con PRECIOS CON DECIMALES.
- Ejemplo: validación de números, de textos, correo electrónico, , validar que si se hace click sobre el botón y no hay nada cargado entonces no se guarde ese registro en la base de datos, etc.
- Al modificar, **modificar todos los campos de la tabla**, obviamente no se modificará el campo clave. También se debe llenar la información que se le va a permitir modificar al usuario automáticamente para que éste luego cambie solo el dato que necesita. Al modificar un solo campo de la base de datos, no se deben borrar los demás campos, solo se deberá actualizar el campo que el usuario esté cambiando
- **Utilizar mensajes de confirmación.** Ejemplo: ¿Está seguro que desea eliminar ese registro?

- **Para la claves**, al dar de alta pedir la clave dos veces y usar asteriscos para no mostrarla.
- Tener cuidado con los **repetidos en la base de datos**, por ejemplo: no podría haber dos personas con el mismo DNI en la base de datos.
- Luego de realizar un cambio sobre un registro por ejemplo al agregar, **limpiar todos los controles**. Es decir, se deberá borrar la información que acaba de ingresar el usuario.
- **Incluir reportes/estadísticas con parámetros**. NO ES LO MISMO UN LISTADO QUE UN REPORTE, UN REPORTE ES INFORMACIÓN QUE SE OBTIENE A PARTIR DE LOS DATOS QUE SE ENCUENTRAN EN LA BASE DE DATOS. Ejemplo: Seleccionar fecha de inicio y fecha de fin, y obtener el monto total de egreso e ingreso en la empresa. Los parámetros en este caso sería la selección de fechas.
- Utilizar **DESCOLGABLES** en los casos que sea necesario. Ejemplo: para provincia, localidades, géneros, etc. Si se modifica dentro de una grilla y tienen campos que aplican para ser descolgables, entonces utilizar dicho control.
- En el caso de dar de **alta un usuario**, un usuario común no puede dar de alta un administrador!!.
- Cambiar el nombre de las **columnas** de las grillas y dejar uno apropiado. Usar para ello la instrucción AS del lenguaje SQL. Ejemplo: “Select p\_nombre AS nombre from Persona.
- **Antes de rendir, se deberá cargar la base de datos con varios registros** e información válida para la prueba... mínimamente 15 registros por tabla... para poder probar búsquedas y filtros.

## **ENTREGAS OBLIGATORIAS TP INTEGRADOR**

Habrá tres entregas del TP integrador. Es necesario aprobar la primera entrega, para poder presentarse a la segunda. Es necesario aprobar la segunda entrega para poder presentarse a la tercera. Cada entrega tendrá su recuperatorio. Si se desaprueba el recuperatorio se desaprueba el TP integrador y por ende la materia.

**PRIMERA ENTREGA: DOMINGO 01/11/2020 23:55HS** Se deberá subir al aula virtual una carpeta comprimida que tenga:

- El proyecto que deberá llamarse TPINT\_GRUPO\_X\_LAB4. Para el diseño de las pantallas se deberán utilizar JSP.
- Un word que contenga el DER y las impresiones de pantalla de cada JSP y en cada imagen explicar el funcionamiento que se podrá realizar sobre la pantalla.

**SEGUNDA ENTREGA: JUEVES 12/11/2020 08:00HS** Se deberá entregar la siguiente funcionalidad del administrador: Login, ABML de clientes, ABML de cuentas y asignación de cuenta a cliente.

Se deberá subir al aula virtual una carpeta comprimida que contenga el proyecto y se defenderá mediante ZOOM. Cada grupo rendirá en un horario asignado.

**TERCERA ENTREGA: JUEVES 19/11/2020 08:00hs** Se realizará de manera a distancia a través de ZOOM y se deberá entregar el TP integrador completo funcionando.

- A cada grupo se le asignará un horario.
- Es importante que a la hora de entregar el TP integrador completo, se haya probado dicho funcionamiento y se hayan cargado datos en la base de datos.
- Utilizaremos la herramienta ZOOM para realizar la defensa del TP, es OBLIGATORIO utilizar la cámara y el audio, además respetar el horario asignado. No presentarse en el horario, es equivalente a desaprobar la entrega.
- La defensa del TP integrador es individual, es decir que se presenta el grupo completo y luego se realizan preguntas individuales a cada integrante para evaluarlo de manera particular. Por ejemplo: puede suceder que, en un grupo de 3 integrantes, aprueben 2.
- Al momento de la evaluación, se deberán tener datos precargados válidos. Mínimamente 15 registros por tabla.

**RECUERDEN QUE PARA REGULARIZAR LA MATERIA ES NECESARIO:**

- APROBAR LA PRIMERA PARTE DE LA MATERIA (APLICACIONES DE CONSOLA/ESCRITORIO EN JAVA)
- APROBAR TODAS LAS ENTREGAS DEL TP INTEGRADOR Y ENTREGAR EL TP INTEGRADOR FUNCIONANDO

**FINAL DE LA MATERIA:**

- LOS ALUMNOS QUE DEBAN RENDIR EL FINAL DE LA MATERIA, DEBERÁN TRAER LOS CAMBIOS PEDIDOS POR EL DOCENTE AL MOMENTO DE ENTREGAR EL TP INTEGRADOR PARA REGULARIZAR LA CURSADA. SI AL MOMENTO DE RENDIR EL FINAL, EL ALUMNO NO TRAE LOS CAMBIOS PEDIDOS POR EL DOCENTE... ENTONCES DESAPROBARÁ DICHO FINAL.