



**UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL GENERAL PACHECO**

**TÉCNICO SUPERIOR EN SISTEMAS
INFORMÁTICOS**

**LABORATORIO DE COMPUTACIÓN V
TP INTEGRADOR**

NOTA: Para realizar el trabajo práctico final deberá utilizar las siguientes herramientas: GIT/ MAVEN / HIBERNATE / JSP /SPRING CORE/SPRING MVC

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. A cada cliente se le asignará un usuario particular para poder acceder a la web del banco.
- ABML de cuentas y asignación de cuenta a cliente. Una cuenta debe pertenecer a un solo tipo de cuenta. Se le podrá asignar un máximo de 4 cuentas a un cliente. Al crear la cuenta se le asignará un monto inicial de 10.000\$.
- Autorización de préstamos, asignación de monto prestado, tiempo y cuotas que deberá pagar el cliente.
- Incluir algún informe en donde se incluya un rango de fechas.

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.
- El cliente podrá pedir un préstamo de cierto dinero al banco y pagar las cuotas. El banco recibirá el pedido y autorizará o no dicho préstamo. Si se autoriza el préstamo, se asigna dinero a la cuenta.
- El cliente tendrá un menú de pago de prestamos en donde figure las cuotas que debe pagar, podrá pagar seleccionando la cuota y cuenta de donde se descontará dicho gasto.

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, no hay diferentes bancos. 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 cuenta posibles:

- Caja de ahorro en pesos
- Caja de ahorro en dólares
- Cuenta corriente
- Cuenta corriente especial en pesos
- Cuenta corriente especial en dólares

Tipos de movimientos posibles:

- Alta de cuenta, genera un movimiento de dinero positivo en la cuenta origen.
- El alta de un préstamo genera un movimiento de dinero positivo en la cuenta origen.
- El pago de un préstamo genera un movimiento de dinero negativo en la cuenta origen.
- 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 e importe.
- Un cliente tiene un DNI, nombre, apellido, sexo, nacionalidad, fecha de nacimiento, dirección, localidad, provincia y un usuario para acceder a la página.
- Una cuenta tiene un cliente asignado, fecha de creación, tipo de cuenta, un número de cuenta, CBU, un nombre y un saldo.

- Un préstamo tiene un cliente asignado, una fecha, el importe prestado, cuotas, un plazo de pago en meses, el monto que deberá pagar por mes. Cada cuota registrará la fecha en la cuál se pago la misma.

Condiciones generales del TP.

- Se pedirá trabajar con GIT (Se subirá al aula virtual un apunte para ayudarlos, cada integrante deberá tener su propio usuario y subirá el aporte de código con su respectivo usuario)
- Debe haber un login. En cada pantalla se podrá observar el usuario logueado.
- Debe haber paginación/búsquedas/filtros en los listados.
- Mensajes aclaratorios “Se agrego correctamente en la base de datos”
- Trabajar con programación en capas y utilizar interfaces.
- En los descolgables deben aparecer las descripciones y no sus identificadores.
- Manejar bajas lógicas y no físicas.
- Validación de los campos.
- Antes de rendir, se deberá cargar la base de datos con varios registros e información válida para la prueba.

Importante, acerca de la última entrega del TP integrador:

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, les pedimos 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. Veremos el GIT para verificar que todos hayan aportado al TP.

Al momento de la evaluación, se deberán tener datos precargados válidos.

FECHAS DE ENTREGAS DEL TP INTEGRADOR

PRIMERA ENTREGA

TEMA: MAVEN + HIBERNATE + JSP (DISEÑO DE PANTALLAS).

FECHA: Martes 09/06/2020 23:55

MODALIDAD:

Se subirá al aula virtual una carpeta comprimida que contenga el proyecto y un archivo WORD con las impresiones de pantalla de cada JSP. Explicar en el WORD que se realiza en cada pantalla.

CONDICIONES:

- Se deberá crear el proyecto con MAVEN y manejar las dependencias desde el archivo POM.
- La base de datos a utilizar será MySQL, no se aceptarán otros gestores.
- La base de datos junto con las tablas y las relaciones serán creados a partir del proyecto mediante la utilización de Hibernate.
- Crear un archivo llamado main. Dentro del mismo, se pide llenar la base de datos con mínimamente 10 registros válidos por cada tabla. Por ejemplo, si tienen que llenar la tabla clientes coloquen nombres y apellidos correctos... no nombre "fafssafs" y apellido "fdskkjfsd".
- Las pantallas deberán realizarse con JSP. Se evaluará que controles van a utilizar, por lo que es sumamente importante que se visualicen TODOS los controles. Ejemplo, si se va a editar sobre una grilla... dibujar una grilla y colocar en su interior el botón eliminar y modificar.... O si van a agregar sobre la grilla, dibujar los textbox respectivos. No se pide entregar funcionalidad.

SEGUNDA ENTREGA: FUNCIONALIDAD ADMINISTRADOR

TEMA: MAVEN/HIBERNATE/SPRING CORE Y SPRING MVC.

FECHA: Viernes 19/06/2020 23:55

MODALIDAD:

Se subirá al aula virtual una carpeta comprimida que contenga el proyecto y un archivo WORD que explique cada pantalla y su funcionalidad.

CONDICIONES:

- Dada la entrega número uno, se agregará la siguiente funcionalidad del administrador: ABML de cliente, ABML de cuentas y asignación de cuenta a cliente.
- Todas las instancias deben ser manejadas mediante SPRING CORE. Utilizar SPRING MVC para las peticiones (No servlets).
- La pantalla principal debe llamarse Index.jsp (Ejecutare el código desde esta pantalla para probar así la funcionalidad)

TERCERA ENTREGA: FUNCIONALIDAD COMPLETA

TEMA: MAVEN/HIBERNATE/SPRING CORE Y SPRING MVC

FECHA: Martes 30/06/2020.

MODALIDAD:

Se sube al aula virtual y se defiende virtualmente mediante clase ZOOM, cada grupo rendirá en un horario previamente definido.

CONDICIONES:

- Se deberá entregar la funcionalidad completa.
- Todas las instancias deberán ser creadas a través de la utilización Beans (Spring Core). Utilizar Spring MVC.
- Programar en capas y utilizar interfaces.