



**BASES DE DATOS**  
Segundo Cuatrimestre de 2024  
Proyecto N° 3 - Parte 1  
**Implementación de un sistema de base de datos bancario**

## 1. Descripción de la aplicación

La aplicación para este sistema se encuentra parcialmente implementada en Java siguiendo el patrón de diseño MVC. Tanto la Vista (Interfaz de Usuario) como el Controlador están totalmente implementados y su tarea consistirá solamente en implementar el Modelo para brindar las funcionalidades que se detallan a continuación.

Existen en la aplicación dos secciones principales: simulación de un **Cajero Automático (ATM)**, para uso de los clientes del banco, y **Administración de Préstamos**, para uso de los empleados del banco. Describiremos a continuación el funcionamiento de cada una de ellas.

### 1.1. Cajero automático (ATM)

Esta parte de la aplicación simulará alguna de las funciones de un cajero automático. La interfaz de usuario permite ingresar un número de tarjeta y un número de PIN de forma oculta, y se deberá controlar que el número de PIN corresponda al número de tarjeta ingresado. En caso de que el PIN sea incorrecto deberá mostrarse un mensaje adecuado y permitir ingresar el PIN nuevamente. En caso que el PIN sea correcto se mostrará un menú que debe permitir realizar las siguientes consultas:

- *Consulta de Saldo*: mostrará el saldo actual de la caja de ahorro asociada a la tarjeta ingresada.
- *Ultimos Movimientos*: mostrará un resumen en forma de tabla con las últimas 15 transacciones realizadas sobre la caja de ahorro. Dicho resumen deberá contener la siguiente información ordenada por fecha y hora en forma descendente:
  - Fecha y hora de la transacción.
  - Tipo de transacción.
  - Monto de la transacción: en caso que la transacción decremente el saldo (extracción, transferencia y débito), el monto deberá mostrarse como un número negativo (precedido por el signo "-").
  - El código de caja donde fué realizada.
  - Caja de ahorro destino, en caso de ser una transferencia.
- *Movimientos por período*: Permitirá ingresar dos fechas válidas y mostrará un resumen (con la misma información que se muestra en *Ultimos Movimientos*) con las transacciones realizadas entre las dos fechas ingresadas.

**Nota 1:** las funciones para realizar transferencias y extracciones serán implementadas en el [proyecto 3 - parte 2](#) usando transacciones.

**Nota 2:** El formato de la fecha y hora recuperado desde JAVA a través de JDBC puede variar según el formato de fecha que esté configurado en Windows. **La aplicación entregada deberá funcionar correctamente independientemente de la configuración de fecha y hora de Windows** (ver apéndice [A](#))

## 1.2. Administración de Préstamos

Esta parte de la aplicación proveerá alguna de las funciones de administración que realizan los empleados del Banco. Inicialmente deberá permitir que se ingrese el número de legajo de un empleado y un password de forma oculta, y controlará que el password corresponda al número de legajo ingresado. En caso de que el password sea incorrecto deberá mostrarse un mensaje adecuado y permitir ingresar el password nuevamente. En caso que el password sea correcto (el empleado se considera “logueado”) se mostrará un menú que permitirá realizar las siguientes tareas:

- *Creación de Préstamos*: la interfaz permitirá seleccionar un cliente mediante su tipo y número de documento y controlará que el mismo no esté pagando un préstamo actualmente. En caso que el cliente tenga un préstamo vigente se deberá mostrar un mensaje y no se podrá crear un nuevo préstamo. Si el cliente no tiene préstamos vigentes se permitirá ingresar un monto que no supere el máximo monto superior de la tabla **Tasa\_prestamo** y se permitirá seleccionar una cantidad de meses que se corresponda con un período de los existentes en la tabla **Tasa\_prestamo**. En base a esta información se podrá obtener la tasa de interes correspondiente y calcular el valor de la cuota (utilizando la fórmula mencionada en el enunciado del proyecto 1). Con toda esta información se creará un nuevo préstamo con fecha actual, asociado al cliente ingresado y al empleado que se encuentra “logueado” actualmente.

**Observación:** En esta instancia no se cargarán en la B.D. las cuotas (pagos) asociadas al préstamo.

- *Pago de Cuotas*: la interfaz permitirá seleccionar un cliente mediante su tipo y número de documento. Una vez seleccionado el cliente la aplicación deberá mostrar el número, valor y la fecha de vencimiento de todas las cuotas impagas (sin fecha de pago) del préstamo asociado a dicho cliente. Se deberá permitir seleccionar una o más cuotas y registrar el pago de las mismas.
- *Listado de clientes morosos*: mostrará un listado en forma de tabla con información de los clientes que se han atrasado en al menos 2 cuotas de su préstamo, esto es, existen al menos dos cuotas impagas cuya fecha de vencimiento es menor a la fecha actual. Dicho resumen deberá contener la siguiente información:
  - Número de cliente, tipo y número de documento, nombre y apellido del cliente.
  - Número de préstamo, monto, cantidad de meses y valor de la cuota del préstamo asociado al cliente.
  - Cantidad de cuotas atrasadas de cada préstamo.

## 2. Fechas y condiciones de entrega

Para obtener la aplicación parcialmente implementada siguiendo el patrón MVC deberán ingresar al siguiente enlace <https://classroom.github.com/a/1HqfVHdX> para asociar su usuario de GitHub (deberá crear uno si no tiene) a su nombre (de los que aparecen en el listado), aceptar la invitación a la tarea en grupo (assignment) *ProyectoBD2024* y crear el nombre de la comisión (team) (si es el primero de la comisión en ingresar) o asociarse a una comisión (team) ya creada (ver capturas apendice B).

Estos pasos crearan automáticamente un repositorio en su cuenta de GitHub con una copia del proyecto parcialmente implementado. La aplicación esta estructurada siguiendo el patrón y convenciones de nombres explicados en la clase de [patrón MVC](#). **Su tarea consistirá en** implementar solamente el Modelo para obtener una aplicación completamente funcional que acceda a la base de datos banco y brinde todas las funciones detalladas anteriormente.

En la carpeta */doc* del repositorio podrá encontrar los diagramas de eventos que muestran el comportamiento de las diferentes partes de la aplicación. El proyecto en su estado inicial compila y se ejecuta correctamente con datos estáticos ficticios introducidos en el código fuente, que permiten probar su comportamiento. Dentro del código del proyecto se encuentra indicado donde deberán incluir su código, el mismo esta marcado como un comentario que pueden buscar el siguiente patrón `* TODO` (en eclipse pueden abrir la ventana de tareas que se encuentra en el menu Windows/Show View/Tasks).

[Aquí](#) puede ver un tutorial sobre el uso de GitHub publicado por la cátedra de Tecnología de Programación:

**Observación:** No realice las actualizaciones del repositorio directamente desde GitHub ni descargue el proyecto con el botón de descarga ZIP, dado que no corresponden al flujo de trabajo a realizar durante un proyecto de desarrollo de software real.

- **Fecha límite de entrega: 22 de Octubre.** La entrega se hará directamente en el repositorio de *github-classroom* de cada comisión. Además de completar los **archivos fuentes** que implementan el Modelo, deberá incorporar al repositorio un **archivo .jar ejecutable** (con todas la librerías incluidas) en la carpeta raíz y los **archivos banco.sql y datos.sql** dentro de la carpeta */sql*. Para la corrección se tomará en cuenta el estado del repositorio en la rama *main* a la fecha de entrega.
- **Comisiones:** Los proyectos deben realizarse en comisiones de *dos alumnos* cada una. Las comisiones deberán ser *las mismas* para todas las entregas.
- **Importante:** La entrega en *tiempo y forma* de este proyecto es *condición de cursado* de la materia.

## A. Funciones de MySQL para el manejo de fechas y tiempo

MySQL provee funciones para manipular fechas y tiempo que resultan muy útiles para el desarrollo de este proyecto. En general estas funciones se pueden consultar al servidor mediante la sentencia SQL select y devuelven una tabla con una única celda que contiene el resultado (más información ver sección 12.7 del manual de MySQL). A continuación se muestran ejemplos de algunas de las funciones disponibles.

**Importante:** los ejemplos que se muestran son el resultado de invocar a las funciones desde el cliente *mysql.exe*. El formato de la fecha y hora recuperado desde JAVA a través de JDBC puede variar según el formato de fecha que esté configurado en Windows. **La aplicación entregada deberá funcionar correctamente independientemente de la configuración de fecha y hora de Windows.** Para esto puede utilizar las funciones provistas en la clase *Utils.Fechas*.

```
■ mysql> SELECT CURDATE();
```

```
+-----+
| CURDATE() |
+-----+
| 2021-09-27 |
+-----+
```

```
1 row in set (0.03 sec)
```

Devuelve la fecha actual en la forma: año-mes-día

```
■ mysql> SELECT CURTIME();
```

```
+-----+
| CURTIME() |
+-----+
| 13:09:37 |
+-----+
```

```
1 row in set (0.00 sec)
```

Devuelve la hora actual .

```
■ mysql> SELECT NOW();
```

```
+-----+
| NOW() |
+-----+
| 2021-09-27 13:09:45 |
+-----+
```

```
1 row in set (0.00 sec)
```

Devuelve la fecha y hora actual .

```
■ mysql> SELECT DAYOFWEEK('2021-09-26');
```

```
+-----+
| DAYOFWEEK('2021-09-26') |
+-----+
| 1 |
+-----+
```

```
1 row in set (0.02 sec)
```

El número devuelto representa el día de la semana correspondiente a la fecha dada. 1 = domingo, 2 = lunes, 3 = martes, ... , 7 = sábado.

■ mysql> SELECT TIMEDIFF('13:40:00', '08:05:00');

```
+-----+
| TIMEDIFF('13:40:00', '08:05:00') |
+-----+
| 05:35:00                          |
+-----+
1 row in set (0.00 sec)
```

TIMEDIFF(h1, h2) devuelve la diferencia de tiempo entre dos horas h1 y h2. h1 debe ser mayor que h2, sino devolverá un número negativo.

TIMEDIFF también se puede usar para calcular la diferencia de tiempo entre dos fechas:

mysql> SELECT TIMEDIFF('2021-09-27 01:01:01', '2021-09-26 23:59:59');

```
+-----+
| TIMEDIFF('2021-09-27 01:01:01', '2021-09-26 23:59:59') |
+-----+
| 01:01:02                                                  |
+-----+
1 row in set (0.00 sec)
```

■ mysql> SELECT TIME\_TO\_SEC('1:00:00');

```
+-----+
| TIME_TO_SEC('1:00:00') |
+-----+
|                3600    |
+-----+
1 row in set (0.00 sec)
```

Devuelve la cantidad de segundos que representa el tiempo dado.

## B. Repositorio Github del proyecto

Join the classroom:

**drg-dcic-uns-bd2024**

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

Can't find your name? [Skip to the next step](#) →

Identifiers	
FERNANDEZ, MAXIMO ALEJANDRO	>
FIGUERE, JOAQUIN	>
GARCIA, DIEGO	>
GARELLI, VALENTINO LEON	>
GARTNER, FACUNDO NICOLÁS	>
GARZA, ROCIO	>
GOMETRO, IVO VALENTIN	>
GERARDI, MATIAS	>
GIACOMODONATO, GIULIA	>
	>

Figura 1: seleccionar su nombre de la lista

drg-dcic-uns-bd2024

## Accept the group assignment — ProyectoBD2024

Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later.

---

Create a new team:

Garcia-Diz

+ Create team

Figura 2: Nombrar la comisión (team) con los apellidos de los integrantes - Create team

drg-dcic-uns-bd2024

## Accept the assignment — ProyectoBD2024

Once you accept this assignment, you will be granted access to the `proyectobd2024-drg-git` repository in the `drg-dcic-uns` organization on GitHub.

---

Accept this assignment


Figura 3: Aceptar la tarea correspondiente al proyecto para que se cree el repositorio



## You're ready to go — Garcia-Diz

You accepted the assignment, **ProyectoBD2024**.

Your team's assignment repository has been created:

 <https://github.com/drg-dcic-uns/proyectobd2024-garcia-diz>

We've configured the repository associated with this assignment.

Note: You may receive an email invitation to join [drg-dcic-uns](#) on your behalf. No further action is necessary.

Figura 4: (si no aparece presione F5) Click en el enlace para ir al repositorio



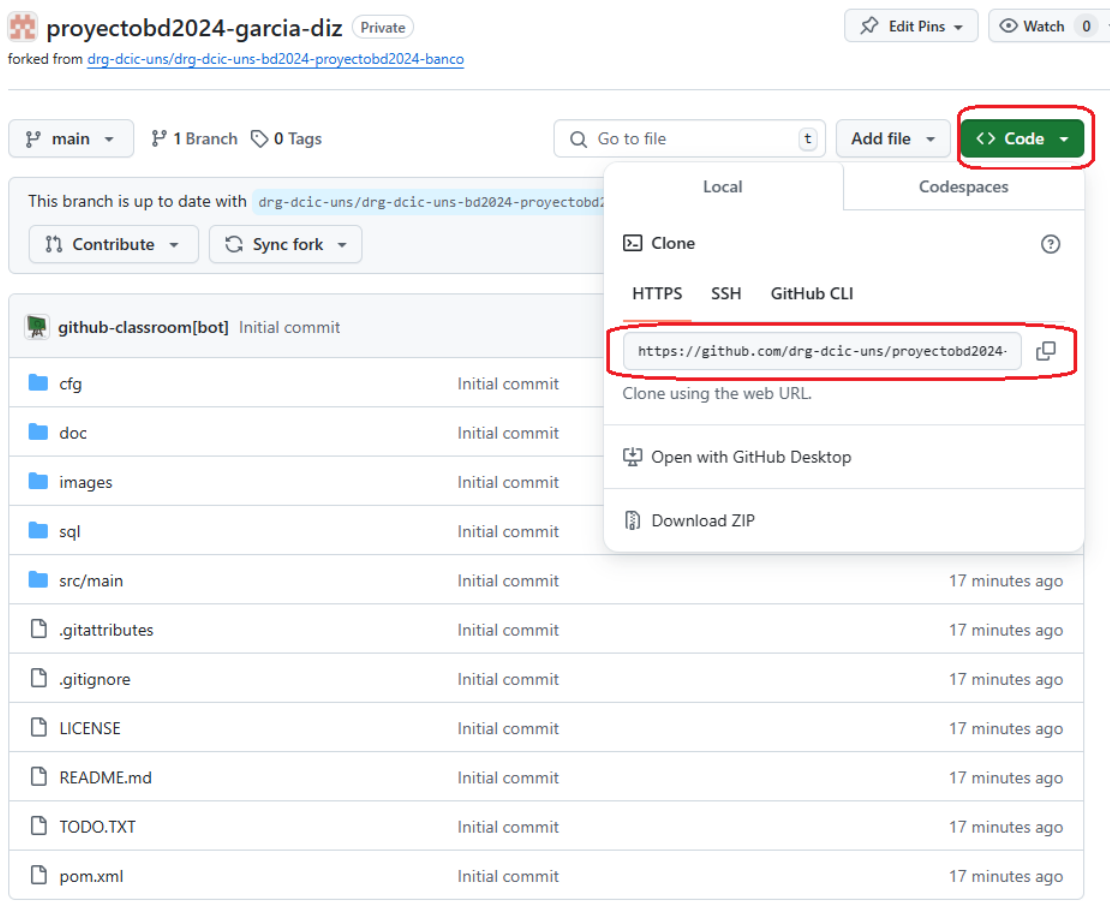


Figura 5: Repositorio creado y enlace para clonarlo en su repositorio local