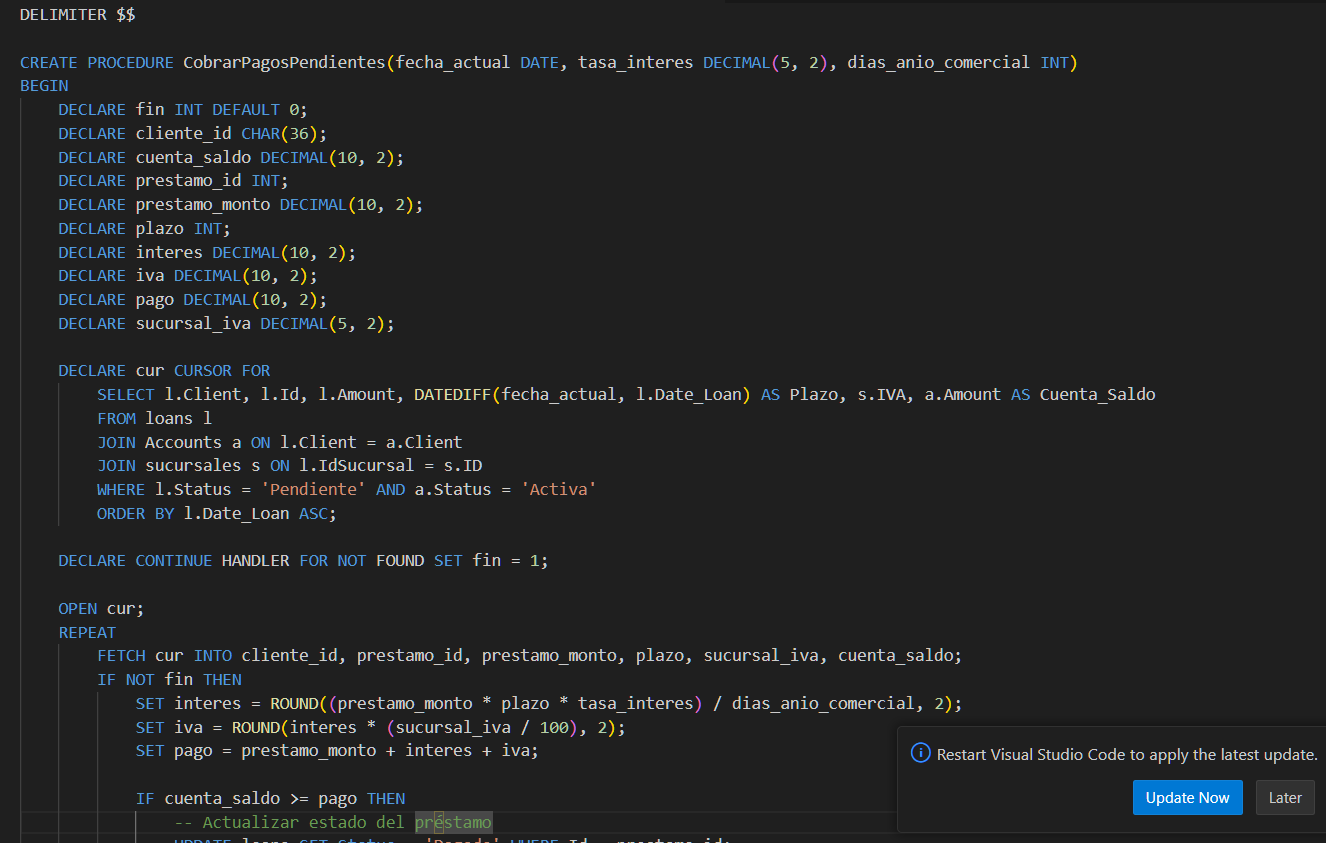
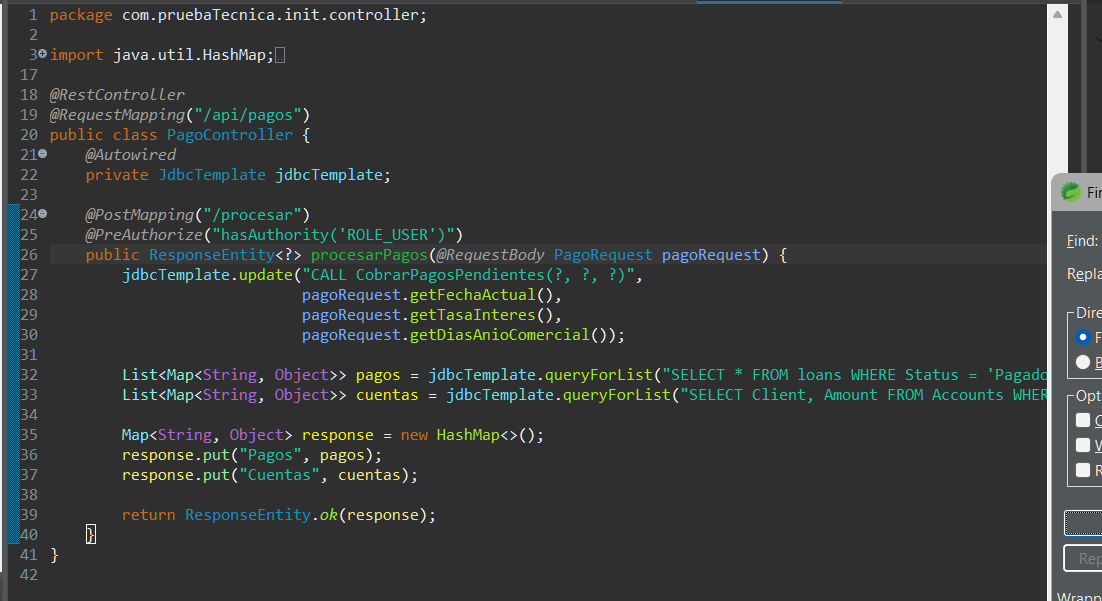
**Explicación del SP:**

Recorremos los préstamos pendientes de clientes con cuentas activas.

Para cada préstamo se calcula el interés, el IVA según la tasa de la sucursal y el monto total del pago.

Si el saldo de la cuenta es suficiente, se realiza el pago, cambiando el estado del préstamo a "Pagado" y actualizando el saldo de la cuenta. 

**Explicación del endpoint /api/pagos/procesar:**

Este endpoint recibe los parámetros de entrada y ejecuta el Stored Procedure. 

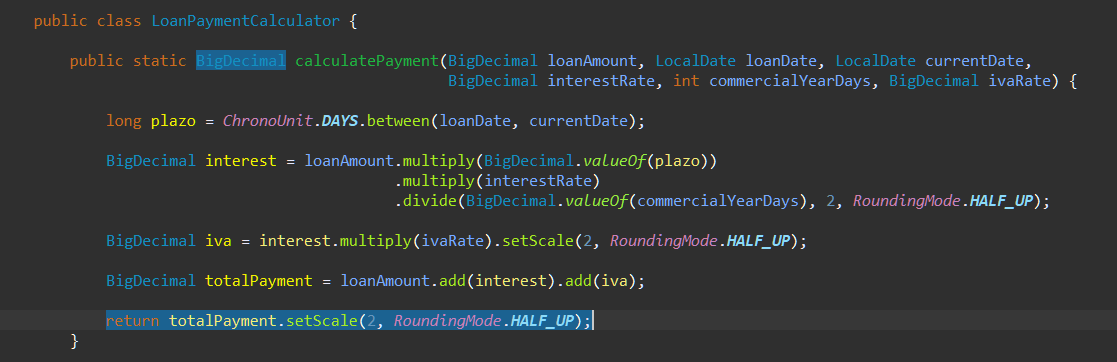
Método para calcular el monto del pago

1. Calcular el plazo (diferencia en días entre la fecha actual y la fecha del préstamo)

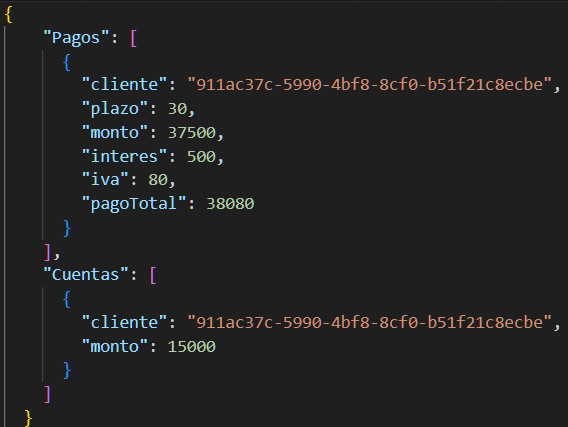
2. Calcular el interés: (loanAmount \* plazo \* tasa\_interes) / dias\_anio\_comercial

3. Calcular el IVA: Interés \* sucursal.tasa\_iva

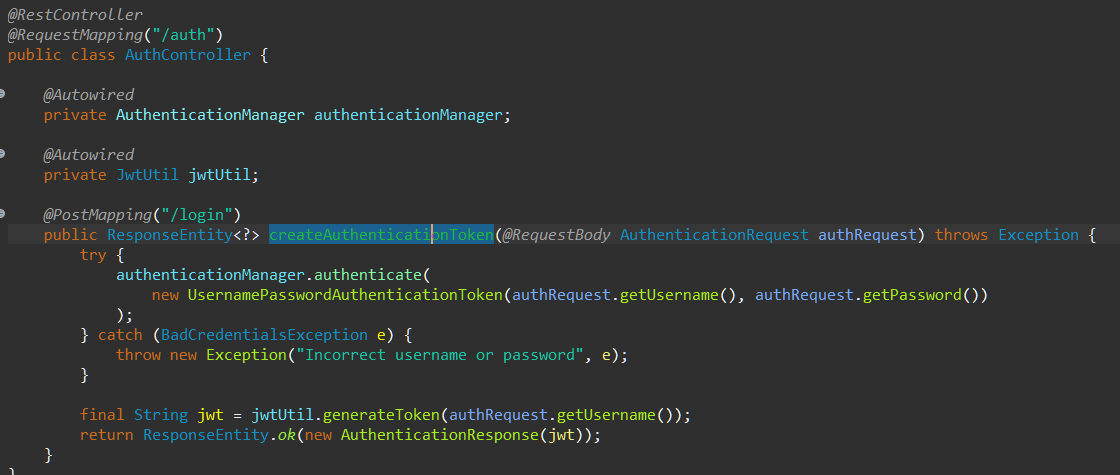
4. Calcular el pago total: loanAmount + interés + IVA

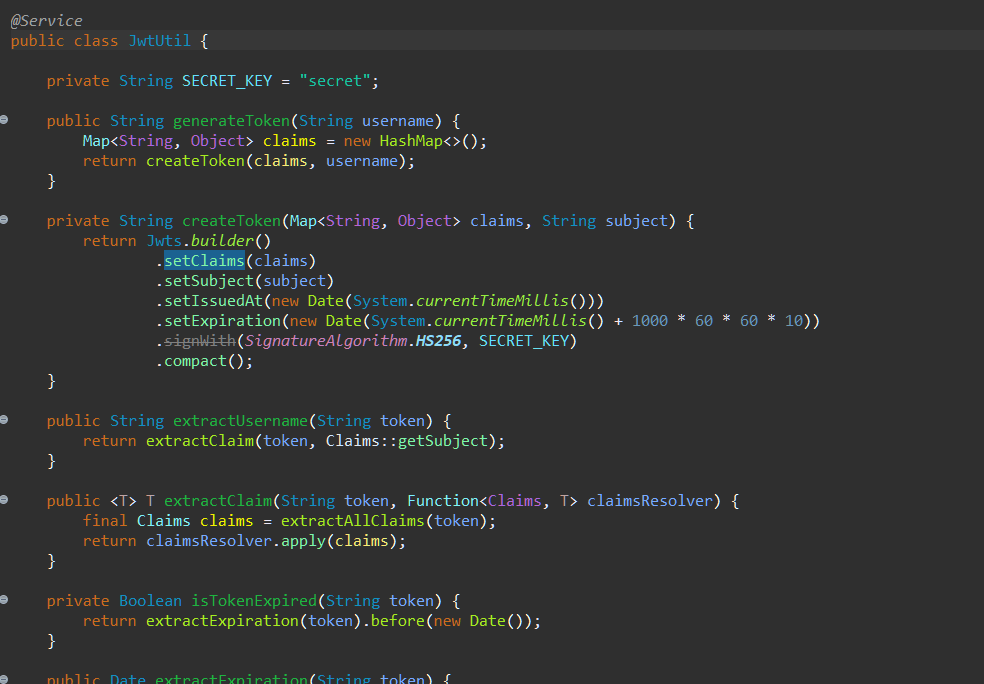


Regresa una respuesta similar a la del requerimiento.



**Explicación del endpoint /auth/login:**

Es el endpoint para autenticar usuarios mendiante JWT

Se crea un útil para la generación, validación y extracción del token JWT

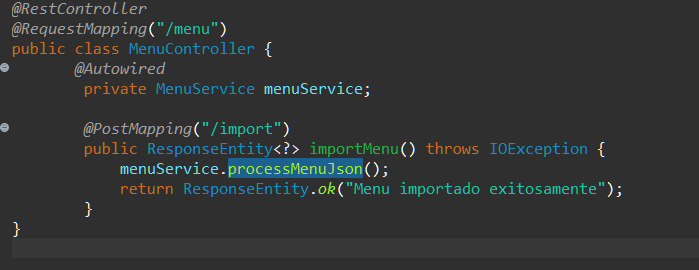
**generateToken()**: Genera un token JWT con el nombre de usuario del token y lo firma utilizando la clave secreta.

**createToken()**: Crea el token con los datos adicionales que quieras. También establece una fecha de expiración para el token.

**extractUsername()**: Extrae el nombre de usuario del token JWT.

v**alidateToken()**: Valida si el token es correcto.

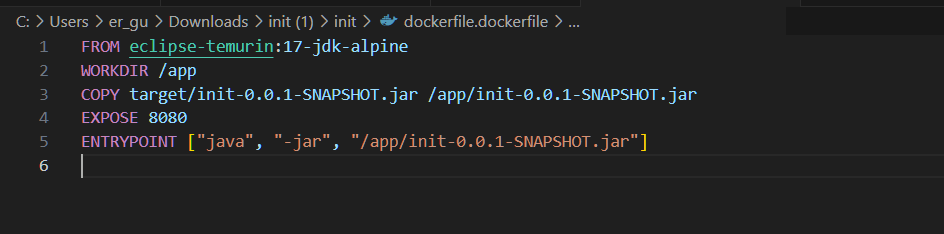
**Explicación del endpoint /menu/import:**

Realiza la importación de datos recursivamente del menú.json

Este enfoque te permite realizar cobros automáticamente con los parámetros proporcionados y también gestionar la autenticación y la inserción de datos recursivamente para el archivo menú.json

**Explicación del archivo Docker:**

Preparamos el dockerfile para que pueda empaquetarse y ejecutarse en un contenedor docker



Pasos para ejecutar el aplicativo:

Ejecutamos los siguientes comandos para crear la imagen Docker:

docker build -t myapp:latest .

Ejecutamos el contenedor Docker:

docker run -p 8080:8080 myapp:latest