

Requisitos Funcionales:

Registro de Clientes:

El sistema debe permitir el registro de información de clientes, incluyendo nombre, documento, dirección y correo electrónico.

Registro de Préstamos:

El sistema debe permitir el registro de préstamos asociados a un cliente.

Los préstamos deben contener información como la fecha de inicio, la frecuencia de pago, la cantidad de cuotas, la tasa de interés y el valor del préstamo.

Registro de Abonos:

Los abonos deben registrarse con la información del cliente, el monto del abono y la fecha en que se realizó.

Al registrar un abono, el sistema debe actualizar automáticamente la deuda del cliente y la fecha de la próxima cuota para cada préstamo asociado.

Cálculo de Deuda y Pagos:

El sistema debe calcular y mostrar la deuda restante para cada préstamo, considerando los abonos realizados.

Debe llevar un registro de los pagos realizados y actualizar el estado del préstamo a "Pagado" cuando la deuda sea cero o negativa.

Actualización Automática de Fechas de Cuotas:

El sistema debe tener la capacidad de calcular y actualizar automáticamente la fecha de la próxima cuota basándose en la frecuencia de pago establecida.

Consulta de Clientes y Préstamos:

Debe ser posible consultar la información de clientes y préstamos registrados en el sistema.

Se deben proporcionar vistas claras que muestren la información relevante de cada cliente y préstamo.

Requisitos No Funcionales:

Seguridad:

El sistema debe garantizar la seguridad de los datos del cliente y del sistema en general.

El acceso a funciones críticas, como la actualización de datos financieros, debe estar restringido a usuarios autorizados.

Usabilidad:

La interfaz de usuario debe ser intuitiva y fácil de usar.

Se deben proporcionar mensajes claros y notificaciones para informar al usuario sobre el resultado de sus acciones, como el registro de abonos.

Escalabilidad:

El sistema debe ser escalable para manejar un crecimiento en el número de clientes y préstamos registrados.

Debe poder gestionar eficientemente grandes volúmenes de datos sin comprometer el rendimiento.

Disponibilidad:

El sistema debe estar disponible de manera constante, minimizando los tiempos de inactividad y asegurando un acceso continuo para los usuarios autorizados.

Mantenibilidad:

El código del sistema debe seguir buenas prácticas de programación y ser fácilmente mantenible.

Se debe proporcionar documentación clara sobre el funcionamiento del sistema y cómo realizar tareas de mantenimiento.

Interoperabilidad:

El sistema debe ser compatible con tecnologías estándar y capaz de integrarse con otros sistemas si es necesario.