APLICACIÓN FINANCIERA: PROYECTO DE AULA ENTREGA 1

POR:

LUIS MIGUEL GIRALDO HURTADO

SANTIAGO GONZÁLEZ MARIN

DAVID RIVERA

UNIVERSIDAD DE MEDELLÍN

PREGRADO EN INGENIERÍA EN SISTEMAS

COMPONENTE: ALGORITMOS Y PROGRAMACIÓN ORIENTADA A OBJETOS

DOCTORA: GABRIEL JAIME THOMAS URIBE (DOCENTE)

MEDELLÍN (ANTIOQUIA)

25 DE SEPTIEMBRE DE 2023

**Tabla de contenidos:**

1. Portada ……………………………………………………………………………………………………… 1
2. Descripción de la empresa y problema a abordar………………………………………. 3
3. Requisitos funcionales ……………………………………………………………………………… 4
4. Modelo del mundo …………………………………………………………………………………… 7

**Préstamo inteligente (Descripción del problema):**

Una tarde escuchamos una idea de negocio interesante, donde una persona de Copacabana compro un software diseñado en los préstamos de dinero, esto a través de la base de datos de unos usuarios, con ello las personas que pagaban de manera cumplida podían figurar a mayores préstamos y recomendar a otras personas que pudiesen acceder al servicio, a su vez si estas personas pagaban de manera oportuna podían tener los mismo beneficios, esta plataforma registra los pagos y lleva el proceso de lo que debe cada persona de manera automatizada. En este sentido la persona que creo este software se lucra vendiendo su aplicación a individuos que prestan dinero y les llama la atención el servicio, pues no hay contacto directo entre prestamista y deudor, la misma aplicación trae a sus clientes y conecta a ambas partes.

En este sentido la idea de negocio nos pareció interesante, pues es un nicho que no esta tan explotado y la idea tiene potencial, lo que sentimos que podría ser mejor es el sistemas de recomendaciones, pues de la aplicación no garantiza que ese nuevo cliente sea necesariamente buen deudor, pero nosotros quisimos adaptar esta idea de negocio, bajo nuestros parámetros para hacerlo útil a algunos familiares que se dedican a la profesión de ser rentistas de capital, por ello primero perfeccionaremos los préstamos, para tener una plataforma multifuncional, pero la idea es que en un futuro se le puedan añadir más desarrollos que dejen que el usuario no solo la use para préstamos, sino para otras actividades como renta de inmuebles o venta de los mismos, pues la finalidad nuestra es brindar soluciones enfocadas en el área financiera, donde se adapten algunos servicios que prestan los bancos a sus clientes, pero esta vez que no sea una entidad financiera constituida, sino entre las mismas personas.

Inicialmente en el programa presentaremos como función principal los préstamos, la idea es tener allí funciones principalmente enfocadas en el prestamista, donde este le brinde una interfaz a su usuario para iniciar sesión a través de un nombre de usuario, el mismo le permitirá ver si tiene pagos vigentes o ya está pagado todo el crédito; o en caso de que este no este registrado, habrá un menú para la inscripción, ya luego cuando el usuario sea válido el mismo tendrá un menú donde pueda acceder a un préstamo, dependiendo del rango de precio que se ponga estará sujeto el interés, donde el mismo puede verse afectado si se es buen o mal deudor con las cuotas de pago. Para tener acceso a la base de datos de los usuarios y el estado de cuentas, el propietario tiene su propia línea de registro con una clave encriptada, este también contara con la posibilidad de definir quien tiene acceso a su crédito, y tendrá el saldo disponible al momento de prestar, pudiendo agregar dinero a su cuenta.

**Requisitos funcionales:**

|  |  |
| --- | --- |
| Nombre | R1 – perfil cliente |
| Resumen | Están almacenados los datos del cliente, pues este para acceder a un préstamo debe registrarse en nuestra base de datos, pues allí se determinarán todas las variables |
| Entradas | * Inicio de usuario existente * Registro de usuario nuevo |
| Resultado | 1. El sistema muestra un mensaje dando la bienvenida al cliente 2. Despliega menú donde despliega, nombre, ocupación, edad y por último el valor del crédito |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Registrar usuario | Información() | nuevo\_usuario |
|  |  |  |

|  |  |
| --- | --- |
| Nombre | R2 – cuota corriente |
| Resumen | Refiere a la capacidad de realizar cálculos para determinar el monto que el cliente debe pagar de manera regular y recurrente, en este caso depende de la cantidad del préstamo pedido, generalmente se calcula para realizar pagos de forma periódica, como mensual o trimestral |
| Entradas | * Monto del préstamo |
| Resultado | 1. Se muestra el monto a dar de cuota inicial para pagar |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Calcular la cuota corriente en base al monto del préstamo solicitado | cuota\_corriente | perfil |

|  |  |
| --- | --- |
| Nombre | R3 – interés |
| Resumen | Este requisito va muy ligado al anterior, pues el porcentaje del mismo dependerá en nuestro sistema según el monto del préstamo, ya que el valor mínimo son 4’000.000, si este abarca desde esta cantidad hasta 55’000.000, se considera “pequeño”, por lo cual el interés sobre cada pago será de un 5% del valor total; pero en caso de ser un monto superior a 55’000.000 se tendrá un interés menor, que es del 3% sobre el valor del préstamo |
| Entradas | * Monto del préstamo |
| Resultado | 1. Se muestra el interés a pagar, o se rechaza el crédito por ser un monto inferior |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Mirar que tipo de préstamo es | cuota\_corriente(cant\_cuotas, tipo\_prestamo, capital:int) -> int | perfil |
| Aplicar intereses al préstamo dependiendo de su tipo | cuota\_corriente(cant\_cuotas, tipo\_prestamo, capital:int) -> int | perfil |

|  |  |
| --- | --- |
| Nombre | R4 – perfil administrador **(**visual de clientes) |
| Resumen | Permitirá al administrador o al prestamista acceder a los datos de todos sus clientes, esto incluye los nombres, datos personales y los debidos prestamos, y para que esta persona sea la única quien conozca el usuario de ingreso se tendrá una clave encriptada, de este modo el propietario podrá llevar mayor control sobre las personas a quien presta su servicio |
| Entradas | * Ingresar usuario del propietario |
| Resultado | 1. Se imprime los datos de todos los prestamos vigentes 2. Se imprimen los clientes |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Mostrar al propietario los aspirantes a prestamos | notificación(lista:list) -> bool: | Propietario |
| Preguntar al propietario si desea conceder el prestamo | notificación(lista:list) -> bool: | Propietario |
| Preguntar si desea saber el curriculum de los clientes | Curriculum() | Propietario |
| Preguntar al propietario si desea recargar la cuenta en efectivo | recarga() | Propietario |

|  |  |
| --- | --- |
| Nombre | R5 – Definir si persona es apta para el préstamo |
| Resumen | Permitirá al administrador o al prestamista acceder a los datos de todos sus clientes, esto incluye los nombres, datos personales y los debidos prestamos, y el mismo pueda aceptar o denegar a una persona el préstamo |
| Entradas | * Ingresar usuario del propietario * Se pone s/n para admitir o denegar |
| Resultado | 1. Se imprime los datos de los clientes en proceso 2. Se admite o deniega el cliente |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Ingresar como propietario |  |  |
| Preguntar al propietario si desea saber el curriculum de sus clientes | Curriculum() | Propietario |
| Preguntar al propietario si desea conceder el prestamo | notificación(lista:list) -> bool: | Propietario |

|  |  |
| --- | --- |
| Nombre | R6 – Saldo del propietario |
| Resumen | Mostrará al propietario el capital disponible a prestar, permitiéndole al mismo agregar saldo a su cuenta. |
| Entradas | * Ingresar saldo a la cuenta |
| Resultado | 1. Se imprime el saldo vigente del propietario 2. Se imprime si desea agregar saldo 3. Imprime nuevo saldo |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Ingresar como propietario |  |  |
| Preguntar al propietario si desea saber el curriculum de sus clientes | Curriculum() | propietario |
| Preguntar al propietario si desea conceder el prestamo | notificación(lista:list) -> bool: | propietario |

|  |  |
| --- | --- |
| Nombre | R7 – Guardar datos en json |
| Resumen | Este requisito se encarga de guardar los datos suministrados por el nuevo usuario como el nombre, edad, profesión, valor del préstamo(cantidad), el cual se convertirá en cliente si el propietario le concede el préstamo |
| Entradas | Nombre, profesión, edad , cantidad, saldo pagado, cuotas, cuota inicial y mora |
| Resultado | Respaldo de datos del cliente |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Convertir clase a diccionario | Todict(Perifl:”perfil”, usuario) -> dict | perfil |
| Ejecutar el método todict y guardarlo en una variable | Guardar\_en\_json | perfil |
| Abrir el archivo json en modo lectura | Guardar\_en\_json | perfil |

|  |  |
| --- | --- |
| Nombre | R8 – Pago completado |
| Resumen | Esta función permitirá al cliente ver sus cuotas pendientes, o el monto a pagar pendiente, pero si están saldadas al momento de iniciar con el usuario enseñará que no hay prestamos pendientes, y con esto nuevamente podrá solicitar nuevos créditos. |
| Entradas | * Usuario intenta pagar cuota |
| Resultado | 1. Si el usuario hace todo el proceso de inicio de cesión y saldo sus cuentas se imprime que no hay créditos vigentes |

|  |  |
| --- | --- |
| Nombre | R9 – Interés mora |
| Resumen | Método que permitirá subir el interés del usuario, en caso de no cumplir la cuota estipulada |
| Entradas | * Ingreso de la cuota estipulada según el crédito y porcentaje |
| Resultado | 1. Se imprime la subida del interés 2. Se deja como nuevo interés vigente, el interés mas el porcentaje de mora |

|  |  |  |
| --- | --- | --- |
| **Pasos** | **Métodos** | **Responsable** |
| Se pregunta al cliente cual es la cantidad que va a ingresar para el pago de su cuota |  | Capital |
| Se crea una excepción en el caso de que el valor de la cuota corriente sea mayor que el capital | Cuota\_corriente() | perfil |
| Se incrementa el atributo mora de el parámetro indicado en la función, en este caso cliente1 | intereses\_mora() | perfil |

**Módelo del mundo:**

