

**SISTEMA CORPORATIVO  
UNIVERSIDAD TECNOLÓGICA DE SANTIAGO  
UTESA**

**Recinto Santo Domingo Oriental  
Facultad de Arquitectura e Ingeniería  
Carrera de Ingeniería en Informática**



**SISTEMA DE GESTIÓN Y ADMINISTRACIÓN DE PRÉSTAMOS PARA  
PRESTAMISTAS Y COOPERATIVAS. PRESTAFACIL.**

Proyecto De Grado Para Optar Por El Título De:

**Ingeniero en Informática**

Presentado por:

**Damian Agustin Guillen Duarte  
Yeuri Dismar Moreta Martinez  
Rafael Emilio Mateo Cuevas**

Asesor

**Elvys Cruz M.A**

Santo Domingo, Republica Dominicana  
Marzo 2020

**SISTEMA CORPORATIVO  
UNIVERSIDAD TECNOLÓGICA DE SANTIAGO  
UTESA**

**Recinto Santo Domingo Oriental  
Facultad de Arquitectura e Ingeniería  
Carrera de Ingeniería en Informática**

**SISTEMA DE GESTIÓN Y ADMINISTRACIÓN DE PRÉSTAMOS  
PARA PRESTAMISTAS Y COOPERATIVAS. PRESTAFACIL.**

Proyecto De Grado Para Optar Por El Título De:

**Ingeniero en Informática**

Presentado por:

**Damian Agustin Guillen Duarte 1-14-6957**

**Yeuri Dismar Moreta Martinez 2-15-6503**

**Rafael Emilio Mateo Cuevas 1-14-6920**

Asesor

**Elvys Cruz M.A**

Santo Domingo, Republica Dominicana  
Marzo 2020

## Contenido

DEDICATORIAS .....	i ii iii
AGRADECIMIENTOS .....	v vi vii
RESUMEN .....	viii
INTRODUCCIÓN .....	ix.

### **CAPÍTULO I: ASPECTOS GENERALES DEL PROYECTO .....**

1.1. Objetivos.....	1
1.1.1. Objetivo General.....	1
1.1.2. Objetivos Específicos.....	1
1.2. Antecedentes.....	2
1.3. Importancia.....	5
1.4. Justificación.....	5
1.5. Delimitación.....	6
1.6. Planteamiento del problema.....	7
1.7. Formulación del problema.....	7

### **CAPÍTULO II: ANÁLISIS DEL SISTEMA .....**

2.1. Investigación de campo.....	9
2.1.1. Análisis de requisitos.....	11
2.1.2. Requisitos técnicos.....	12
2.2. Funciones del sistema.....	13
2.3. Metodología de desarrollo.....	14
2.4. Casos de Uso.....	14
2.4.1. Diagramas de Casos de Uso.....	15
2.5. Diccionarios de datos .....	22

### **CAPÍTULO III: DISEÑO Y DESARROLLO DEL SISTEMA .....**

3.1. Diseño del sistema.....	27
3.1.1. Arquitectura del sistema.....	27
3.2. Herramientas y lenguajes de desarrollo.....	27
3.2.1. PHP.....	27
3.2.2. C#.....	28

3.2.3. JavaScript.....	29
3.2.4. MySQL.....	30
3.2.5. CSS.....	31
3.2.6. Bootstrap.....	32
3.2.7. Xampp.....	33
3.2.8. Postman.....	34
3.3. Pantallas del sistema .....	35
3.3.1. Dashboard.....	35
3.3.2. Listado de Clientes.....	35
3.3.3. Listado de Préstamos.....	36
3.4. Módulos del sistema.....	37
3.5. Roles de usuarios.....	38
 CONCLUSIONES.....	 40
RECOMENDACIONES.....	42
BIBLIOGRAFÍA.....	43
ANEXOS.....	45

## ÍNDICE DE FIGURAS

Figura 1. Logo Finsoftek Inc.....	2
Figura 2. Logo Prestan2.....	3
Figura 3. Logo Kpresta.....	4
Figura 4. Logo PrestamistApp.....	4
Figura 5. Logo PHP.....	27
Figura 6. Logo C#.....	28
Figura 7. Logo JavaScript.....	29
Figura 8. Logo MySQL.....	30
Figura 9. Logo Bootstrap.....	32
Figura 10. Logo Xampp.....	33
Figura 11. Logo Postman.....	34
Figura 12. Imagen del Dashboard.....	35
Figura 13. Imagen del listado de clientes.....	35
Figura 14. Imagen del listado de préstamos.....	36
Figura 15. Conexión a la base de datos.....	45
Figura 16. Código de las rutas de los módulos.....	45

## ÍNDICE TABLAS

Tabla 1. Funciones del sistema.....	13
Tabla 2. Gestión de clientes (agregar).....	15
Tabla 3. Gestionar préstamos (agregar).....	16
Tabla 4. Solicitudes de préstamos (solicitar).....	17
Tabla 5. Cobranza de préstamos (registrar).....	18
Tabla 6. Gestionar Usuarios (modificar).....	19
Tabla 7. Realizar cuadre caja (cerrar caja).....	20
Tabla 8. Gestionar Garantías.....	21
Tabla 9. Expedientes.....	22
Tabla 10. Préstamos.....	23
Tabla 11. Cuotas.....	25
Tabla 12. Historial de pagos.....	26
Tabla 13. Usuarios.....	27
Tabla 14. Garantías.....	28
Tabla 15. Historial de usuarios.....	29

## **DEDICATORIAS**

•

*A Dios.*

Por haberme dado fuerza y salud para cumplir todas mis metas.

*A mis Padres Agustin e Ysabel.*

Por brindarme todos los recursos necesarios y estar a mi lado apoyándome y aconsejándome siempre.

*A mi Hermana Genesis Mabel.*

Por ayudarme y ser un modelo a seguir para mi.

Todo este trabajo y esfuerzo han sido posibles, gracias a ellos.

***Damian***



*A Dios.*

Por haberme acompañado y guiado a lo largo de mi carrera, por darme fuerza en los momentos de debilidad.

*A mi Padres.*

Por el apoyo que recibí de ellos durante el transcurso de mi carrera.

*A mis Hermanos.*

Por ser parte importante de mi vida, gracias por estar ahí, cada momento que lo necesite.

***Yeuri***

Dedico este proyecto de grado a mi madre Yesenia Cuevas Díaz quien es la persona más importante en mi vida. Eres una mujer que simplemente me llenas de orgullo, siempre has estado pendiente de mi bienestar y mi educación, gracias a ti he podido culminar mi camino universitario muchas gracias.

***Rafael***

## **AGRADECIMIENTOS.**

## **AGRADECIMIENTO**

Agradezco a Dios por darme la fortaleza y la oportunidad de terminar una etapa importante de mi vida.

A mis padres y a mi familia por el apoyo incondicional y porque siempre estuvieron presentes en todo momento para ayudarme en mi carrera profesional.

Al maestro Amadis Suárez, por guiarme y darme buenas oportunidades para crecer como profesional.

También agradezco a Elvis Cruz y a todos mis compañeros que me acompañaron en todo el transcurso de la universidad.

***Damian***

## **AGRADECIMIENTO**

Agradezco a Dios por la fortaleza y sabiduría que me dio para vencer los obstáculos.

A mis padres que me han dado la existencia y en ella la capacidad por superarme y desear lo mejor en cada paso por este camino difícil y arduo de la vida a construir y forjar las persona que ahora soy.

A mi maestro por su gran apoyo ofrecido en este trabajo, por habernos transmitidos los conocimientos obtenidos y habernos llevado paso a paso en el aprendizaje.

A mis compañeros de trabajo, por ser mis compañeros y porque siempre estaban dispuestos a hacer lo imposible para hacer un buen trabajo.

También agradezco a el señor Elvis Cruz, Felipe Davit y a un gran amigo Ronel Cruz, porque estas personas ayudaron a que este sueño se hiciera realidad.

Muchas gracias a todos.

***Yeuri***

## **AGRADECIMIENTO**

Le agradezco a Dios por haberme permitido recorrer el camino para llegar a esta meta y por permitir poder llegar hasta aquí y presentar este proyecto de grado con el que termino mi camino universitario.

Agradezco a mis compañeros de grupo quienes fueron promotores para poder alcanzar la meta de culminar mi carrera universitaria les agradezco por los consejos que me proporcionaron gracias a estos pude continuar adelante y culminar el largo camino que me ha permitido llegar hasta el final.

***Rafael***

## **RESUMEN**

En el presente documento se detalla el procedimiento de desarrollo del sistema de gestión y administración de préstamos para prestamistas y cooperativas “PrestarFacil”. Está dividido en 3 capítulos.

En el primer capítulo desarrollamos los aspectos generales del proyecto como objetivos, importancia, justificación, antecedentes, planteamiento del problema y delimitación.

En el segundo capítulo se realiza el análisis de sistema explicando los requerimientos, funciones del sistema, metodologías y estructura de los datos.

En el tercer capítulo se presenta el diseño del sistema que incluye las pantallas, los roles de usuarios y los requisitos necesarios para la implementación del software.

Finalmente, en el último capítulo se encuentran las conclusiones y recomendaciones obtenidas en el desarrollo del proyecto finalizado.

## INTRODUCCIÓN

Los beneficios de contar con un sistema de administración de préstamos, se vuelve sumamente importante para un prestamista o cooperativa. La automatización de las actividades, el acceso a la información y el control de negocio. Contar con este tipo de servicios ofrece tantas ventajas, que consideramos que todos los prestamistas y cooperativas deberían tenerlo.

Haremos todo lo posible para que nuestro sistema de préstamos cumpla las necesidades actuales de nuestros clientes. Entendemos que esta decisión de inversión implica un desembolso de fondos para la empresa, por este motivo mantendremos metodologías que cumplan con los plazos de entregas pactados y tengan una excelente calidad. El sistema se realizará para 6 roles claramente definidos y los cuales serán los usuarios finales:

- Administrador.
- Cobrador.
- Contador.
- Secretario.
- Cajero.
- Soporte técnico.

A continuación, pretendemos brindar una breve descripción de nuestro plan de negocio.



**CAPÍTULO I**  
**ASPECTOS GENERALES DEL PROYECTO.**

## **1. OBJETIVOS.**

### **1.1.1. OBJETIVO GENERAL.**

Ofrecer un sistema de préstamo competitivo, eficiente y de calidad para satisfacer las necesidades y demandas de los prestamistas y cooperativas.

### **1.1.2. OBJETIVOS ESPECÍFICOS.**

- Investigar la demanda insatisfecha que tienen los clientes con respecto a los sistemas de préstamos existentes, a fin de brindar soluciones a esta.
- Proporcionar un servicio al cliente eficiente y de calidad, que marque una diferencia con nuestra competencia.
- Actualizar el sistema constantemente con las tecnologías del momento.
- Formar un equipo altamente capacitado que desarrolle el sistema sin problemas y en el plazo estimado.

## **1.2. ANTECEDENTES.**

### **1.2.1. SISTEMA DE PRÉSTAMOS FINSOFTEK.**



*Figura 1. Logo FinSofTek Inc. Fuente: <http://programaparaprestamos.com>*

Es un sistema de préstamos desarrollado por la empresa Finsoftec Inc, que es una empresa dedicada al desarrollo y ventas de software para el manejo de préstamos y otros instrumentos financieros.

Tienen como objetivo de proporcionar el poder de manejo de préstamos bancario a todo individuo o negocio que maneje financiamientos o préstamos.

### 1.2.2. SISTEMA DE GESTIÓN DE PRÉSTAMOS, HIPOTECAS Y FINANCIAMIENTOS.



*Figura 2. Logo Prestan2.* Fuente: <https://web.prestan2.com>

Prestan2 es un sistema desarrollado por Grupo MediaSoft Srl. Le permite gestionar de manera rápida, fácil y segura sus financiamientos, préstamos personales, hipotecarios, entre otros, ofreciendo una serie de herramientas especializadas para cada área de su empresa.

Tienen como objetivo proporcionar un sistema de préstamos moderno y eficiente para prestamistas y cooperativas.

### 1.2.3. KPRESTA



*Figura 3. Logo Kpresta. Fuente: <https://www.kpresta.com>*

Kpresta fue desarrollado por un equipo de programadores encabezado por el ingeniero Alberto Martínez en Santiago de los caballeros, República dominicana. Busca la manera de hacer el cobro de préstamos más fácil y sencillo para los prestamistas. Incluye todas las herramientas necesarias para operar y administrar su operación como el procesamiento de préstamos, calculadoras de préstamos, manejo de ingresos, reportes, manejo de referencias comerciales, información laboral y referencias personales de clientes.

### 1.2.4. PRESTAMISTAPP



*Figura 4. Logo PrestamistApp. Fuente: <https://prestamistapp.com/>*

Es una aplicación móvil que busca poder facilitar la cobranza de préstamos desde un móvil o pc, y ayudar a muchos prestamistas que buscan como llevar su negocio de manera segura y rápida. Incluye amortización, reporte de ingresos, historial de pagos, cuentas por cobrar, recibo de pagos, pagaré notarial y más módulos.

### **1.3. IMPORTANCIA.**

El internet y la tecnología ha cambiado radicalmente la forma de hacer negocios, los servicios se ofrecen en línea y los procesos son más eficientes.

Los prestamistas necesitan aprovechar todas las ventajas que ofrecen estas nuevas tecnologías y un software de administración y gestión de préstamos puede satisfacer esta necesidad y generar valor al mercado.

Es muy importante poder contar con este sistema porque podemos realizar préstamos de manera precisa y sencilla. El uso de este software garantiza la seguridad de la información y evita que pueda ser robada o alterada por personas ajenas a la empresa.

### **1.4. JUSTIFICACIÓN.**

En el mercado de los préstamos es imprescindible gestionar y sostener un control sobre todo lo que pasa en el negocio para evitar fraudes, errores, pérdidas de dinero, y de esta manera mantener un rendimiento aceptable y asegurar una continuidad de futuro, por lo cual este motivo es nuestro principal motor para desarrollar este proyecto.

Este se basa en la creación de un sistema de préstamos multiplataformas con almacenamiento en la nube, que tendrá diferentes tipos de usuarios como el administrador, cobrador y el secretario, y permitirá mantener informado en tiempo real al prestamista sobre los clientes, préstamos, cobros, balance de capital, estados financieros y reportes.

### **1.5. DELIMITACIÓN DEL PROYECTO**

El software a construir tiene como objetivo principal, apoyar en la gestión de prestamistas y cooperativas de diferentes provincias de república dominicana, que deben compartir datos a nivel local. Se desea automatizar, fundamentalmente, la gestión de préstamos y de esta manera integral, desde el momento en que una persona requiera un préstamo, pasando por la oferta de dicha vivienda a los posibles beneficiarios, la gestión de las visitas y la posterior firma del contrato de arrendamiento de efectivo. Se contempla la posibilidad de utilizar correo electrónico o mensajes SMS generados automáticamente, para agilizar la comunicación con los clientes.

También se desea implantar un sitio web en el que se pueda consultar información sobre los préstamos disponibles.

## **1.6. PLANTEAMIENTO DEL PROBLEMA**

En la actualidad, muchos prestamistas y cooperativas necesitan un sistema que les permita mantener un control para así evitar fraudes, descuadres, pérdida de tiempo, y para brindar un mejor servicio a sus clientes.

Obtener un servicio que satisfaga esta expectativa de los prestamistas es actualmente un problema.

## **1.7. FORMULACIÓN DEL PROBLEMA**

¿Es factible desarrollar un sistema que cumpla con las nuevas exigencias que tienen los prestamistas y cooperativas?



## **CAPÍTULO II**

### **ANÁLISIS DEL SISTEMA.**

## **2.1. INVESTIGACIÓN DE CAMPO**

En el comienzo del proyecto vamos a comenzar con un cliente que nos proporcionará la información relevante para el desarrollo del mismo.

El nombre de este cliente es Javier Tejada. Es un prestamista privado y su negocio se llama D` ALEX PRESTA FACIL

**¿De qué manera usted registra los clientes y los préstamos de su negocio?**

Registro los clientes y los préstamos en un archivo Excel.

**¿Cómo es el proceso de cobro de su empresa?**

Notificamos a nuestros clientes por diferentes medios de comunicación para que nos realicen el pago por transferencia, y también tengo a mi disposición a un cobrador que sale a la calle a cobrar.

**¿Posee algún sistema en el cual pueda administrar sus préstamos?**

No. Uso una libreta y una plantilla que tengo en Excel.

**¿Alguna vez ha presentado inconvenientes con el cuadre de caja y la contabilidad?**

Si. En muchas ocasiones he tenido que acostarme a las 2 AM para poder hacer el cuadre del día. Eso me quita tiempo que podría usarlo para estar con mi familia.

**¿Cómo lleva el control de su negocio?**

Me mantengo en comunicación constante con mi cobrador. Reviso las informaciones que tengo en mi agenda y en el Excel, y al final del mes hago un pequeño reporte de mis ganancias.

**¿Le gustaría tener un servicio web para administrar su negocio?**

Si, me gustaría intentarlo para mejorar mi negocio.

**¿Cree que a través de este sistema pueda ser más eficiente en su negocio?**

Si. Nunca he usado un sistema de préstamos, pero estoy seguro que mejorará mucho mi negocio.

**¿Qué opiniones nos ofrece para la implementación de este sistema?**

Me gustaría que el sistema funcione en mi celular porque es el dispositivo que siempre tengo en la mano. También es importante la contabilidad, los reportes y las estadísticas.

### **2.1.1 ANÁLISIS DE REQUISITOS.**

Por las conversaciones que tuvimos con nuestro primer cliente, decidimos que los requisitos del sistema de préstamos son los siguientes:

- La aplicación debe ser multiplataforma, siendo el móvil y la PC los dispositivos preferidos por los usuarios.
- La aplicación tiene que tener los siguientes usuarios: Administrador, Cobrador, secretario, Cajero y Contador.
- La aplicación debe de ser confiable y fácil de usar, para que el usuario pueda hacer todas las actividades de su negocio y le sobre tiempo que pueda aprovechar en otros aspectos de su vida.
- La aplicación debe de tener estadísticas y reportes intuitivos para que el administrador esté al día con su negocio.
- La aplicación tiene que tener un cuadro de caja intuitivo, para evitar cualquier inconveniente y agilizar el proceso.
- El sistema debe de ofrecer un soporte técnico que esté disponible en horario de lunes a sábados, para solucionar cualquier percance que se presente.

### **2.1.2. REQUISITOS TÉCNICOS.**

Para el correcto funcionamiento del sistema, requerimos lo siguiente:

- Un servidor para alojar el sistema web y la base de datos.
- Un dominio.
- Un servidor de correo.
- Una plataforma de marketing.
- Una aplicación móvil y de escritorio.
- Una aplicación de monitoreo de red, firewalls y un antivirus.
- Un repositorio para guardar los códigos del proyecto.

## 2.2. FUNCIONES DEL SISTEMA

*Tabla 1. Funciones del sistema*

<b>FUNCIÓN</b>	<b>TIPO USUARIO</b>
Registrar y editar clientes	Secretario
Solicitar préstamos	Cajero
Registrar y editar préstamos	Administrador
Cobranza de préstamos	Cobrador
Realizar cuadre de caja	Administrador
Acceso al módulo de contabilidad, estadísticas, reportes y conciliaciones bancarias	Contador y Administrador
Acceso al módulo de garantías	Administrador
Configuración de usuarios, contratos, cartas de saldo y notificaciones	Administrador y Soporte técnico
Aplicar descuentos	Administrador
Reimprimir recibos de pagos	Administrador
Acceso al historial de usuarios	Administrador y Soporte técnico
Acceso a la agenda de cobros, rutas y a los mapas	Cobrador y Administrador
Visualizar préstamos y clientes	Secretarios, Cobrador y Administrador
Exportar datos	Contador y Administrador

## **2.3. METODOLOGÍA DE DESARROLLO**

Para desarrollar el proyecto vamos a utilizar la metodología SCRUM, que se caracteriza porque se enfoca más en el cliente y en el desarrollo constante del proyecto que en la planificación tradicional.

Los atributos que se evalúan en esta metodología [1 p. 92, 93] son los siguientes:

- Adaptarse a los entornos turbulentos.
- Satisfacción del usuario final.
- Favorable al outsourcing.
- Aumento de la productividad.
- El respeto de un nivel de calidad.
- El respeto de las fechas de entrega.
- Cumplimiento de los requisitos.

## **2.4. CASOS DE USO.**

Según IBM, “Un caso de uso es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable y pueden representarse como un elemento gráfico en un diagrama y como una especificación de caso de uso en un documento textual”

A continuación, mostramos diagramas de los casos de usos del sistema.

### 2.4.1. DIAGRAMAS DE CASOS DE USO.

*Tabla 2. Gestión de clientes (agregar).*

<b>Casos de Uso</b>	Gestionar Clientes
<b>Acción</b>	Agregar
<b>Actor</b>	Secretario
<b>Finalidad</b>	Registrar Clientes
<b>Descripción</b>	Esta acción consiste en registrar un cliente en el sistema
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1, Opción registrar cliente	2. Se visualiza un formulario para ingresar los datos
3. Se completan todos los campos del formulario	
4. Se da click a guardar	5. Se guarda la información y se redirige al listado de los clientes.



**Tabla 3. Gestionar Préstamos (agregar).**

<b>Caso de Uso</b>	Gestionar Préstamos
<b>Acción</b>	Agregar
<b>Actor</b>	Administrador
<b>Finalidad</b>	Registrar Préstamos
<b>Descripción</b>	Esta acción consiste en registrar un préstamo
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1. Opción registrar préstamo	2. Se visualiza un formulario para registrar el préstamo
3. Se elige al cliente y garante, y se completan todos los campos del formulario	
4. Se da click en calcular.	5. Se genera una tabla de amortización
6. Se confirma la tabla de amortización y se da click a guardar	7. Se redirige al préstamo

**Tabla 4. Solicitudes de préstamos(solicitar).**

<b>Caso de Uso</b>	Solicitudes de Préstamos
<b>Acción</b>	Solicitar
<b>Actor</b>	Cajero
<b>Finalidad</b>	Solicitar Préstamos
<b>Descripción</b>	Consiste en realizar las solicitudes de los préstamos que serán aprobados por el administrador
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1. Opción solicitar préstamo	2. Se visualiza el formulario para la solicitud del préstamo
3. Se completan todos los campos del formulario	
4. Se presiona el botón solicitar préstamo	5. Se realiza una solicitud del préstamo para el administrador
	6. Se envía una notificación al administrador por correo electrónico
7. El administrador queda encargado de aceptar o rechazar la solicitud	

**Tabla 5. Cobranza de préstamos(registrar).**

<b>Caso de Uso</b>	Cobranza de préstamos
<b>Acción</b>	Agregar
<b>Actor</b>	Cobrador
<b>Finalidad</b>	Cobrar cuotas de préstamos
<b>Descripción</b>	Consiste en realizar el cobro de las cuotas del préstamo
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1. Opción cobranza préstamo	2. Se visualiza el listado de los préstamos
3. El cobrador elige el préstamo a cobrar.	4. Se redirige a una página con los datos del préstamo
5. Se da click a pagar.	6. Se abre la ventana de pagos.
7. Se ingresa el monto a pagar y se da click en pagar	8. Se registra el pago en el sistema.
	9. Se imprime el recibo de pago
	10. Se redirige al préstamo para visualizar el pago realizado

**Tabla 6. Gestionar Usuarios (Modificar).**

<b>Caso de Uso</b>	Gestionar Usuario
<b>Acción</b>	Modificar
<b>Actor</b>	Administrador y Soporte técnico
<b>Finalidad</b>	Modificar Usuario
<b>Descripción</b>	Consiste en modificar los datos del usuario
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1. Opción modificar usuario	2. Se visualiza el formulario para editar el usuario
3. Se ingresan los nuevos datos del formulario	
4. Se da click a guardar	5. Se redirige al listado de usuarios

**Tabla 7. Realizar cuadro de caja (cerrar caja).**

<b>Caso de Uso</b>	Cuadre de Caja
<b>Acción</b>	Cerrar caja
<b>Actor</b>	Administrador
<b>Finalidad</b>	Cerrar caja
<b>Descripción</b>	Consiste en cerrar la caja con los pagos del día
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1. Opción cerrar caja	
2. Se da click al botón cerrar caja	3. Se abre una ventana para elegir al cobrador del cuadro de caja
4. Se elige al cobrador	5. Se despliega un formulario con los datos del cuadro de caja
6. Se completa todos los campos del formulario y se da click a cerrar caja	7. Se cierra la caja para el cobrador previamente elegido

**Tabla 8. Gestionar Garantías.**

<b>Caso de Uso</b>	Gestionar Garantías
<b>Acción</b>	Agregar
<b>Actor</b>	Administrador
<b>Finalidad</b>	Registrar garantía
<b>Descripción</b>	Con esta acción podemos registrar una garantía en el sistema
Secuencia de los eventos	
Acciones del usuario	Acciones del sistema
1. Opción garantías	2. Se visualiza 3 pestañas con los tipos de garantía (vehículos, apartamentos y otros)
3. Se elige el tipo de garantía y se da click a nueva garantía	4. Se visualiza el formulario para agregar la garantía
5. Se ingresan todos los campos del formulario y se da click a guardar	6. Se guarda la garantía en el sistema con estado de “almacén”

## 2.5. DICCIONARIO DE DATOS.

*Tabla 9. Expedientes:* En esta tabla se registran los clientes y garantes del sistema.

ATRIBUTO	DESCRIPCIÓN	TIPO	PK	NULL
Id	Identificador único del cliente o garante	INT	Si	Not Null
Nombre	Nombre	VARCHAR(100)		Not Null
Apellido	Apellido	VARCHAR(100)		Not Null
Teléfono	Teléfono	INT		Not Null
Cedula	Cédula	VARCHAR(50)		Not Null
Fecha_nacimiento	Fecha de nacimiento	DATETIME		Not Null
Dirección	Dirección	TEXT		Not Null
Email	Email	TEXT		Not Null
Referencias laborales	Referencias laborales	TEXT		Not Null
Comentarios	Comentarios	TEXT		Not Null

**TABLA 10. Préstamos:** Se registran los préstamos del sistema.

<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO</b>	<b>PK</b>	<b>NULL</b>
Id	Identificador único del préstamo	INT	Si	Not Null
Tipo	Tipo de préstamo	INT		Not Null
Codigo_referencia	Código de referencia manual	VARCHAR(100)		Not Null
Capital	Capital del préstamo	DOUBLE		Not Null
tipo_tasa	Tipo de tasa del préstamo	INT		Not Null
porciento_tasa	Tasa del préstamo	DOUBLE		Not Null
nro_cuotas	Número de cuotas	INT		Not Null
Frecuencia	Frecuencia	INT		Not Null
Fecha_contrato	Fecha del contrato	DATETIME		Not Null
fecha_primer_pago	Fecha primer pago	DATETIME		Not Null
Mora	Mora	DOUBLE		Not Null
dias_gracia	Días de gracia	INT		Not Null
Costo_cierre	Costo de cierre	DOUBLE		Not Null



<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO</b>	<b>PK</b>	<b>NULL</b>
Id_moneda	Moneda	INT		Not Null
Estado	Estado del préstamo	INT		Not Null
Id_usuario	Id del usuario que registró el préstamo	INT		Not Null
Id_cliente	Id del cliente	INT		Not Null
Id_garante	Id del garante	INT		Not Null
Id_garantia	Id de la garantía	INT		Not Null
Detalles_garantia	Detalles de la garantía	TEXT		Not Null
Id_contrato	Id del contrato del préstamo	INT		Not Null
Total_intereses	Total de intereses	DOUBLE		Not Null
Total	Total	DOUBLE		
Monto_cuotas	Monto de las cuotas	DOUBLE		Not Null
Fecha_creacion	Fecha de creación del préstamo	DATETIME		Not Null
Fecha_actualizacion	Fecha de actualización del préstamo	DATETIME		Not Null

**Tabla 11. Cuotas:** En esta tabla se guardan las cuotas de los préstamos.

<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO</b>	<b>PK</b>	<b>NULL</b>
Id	Identificador único de las cuotas	INT	Si	Not Null
Capital	Capital de la cuota	DOUBLE		Not Null
Intereses	Intereses de la cuota	DOUBLE		Not Null
Mora	Mora de la cuota	DOUBLE		Not Null
Otros_gastos	Otros gastos de la cuota	DOUBLE		Not Null
Fecha	Fecha de la cuota	DATETIME		Not Null
Id_prestamo	Id del préstamo a la que pertenece la cuota	INT		Not Null
Estado	Estado de la cuota	INT		Not Null

**Tabla 12. Historial de pagos:** En esta tabla se guardan los pagos de las cuotas.

<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO</b>	<b>PK</b>	<b>NULL</b>
Id	Identificador único del pago	INT	Si	Not Null
Capital_pagado	Capital pagado	DOUBLE		Not Null
Intereses_pagados	Intereses pagados	DOUBLE		Not Null
Mora_pagada	Mora pagada	DOUBLE		Not Null
Otros_gastos_pagados	Otros gastos pagados	DOUBLE		Not Null
Total_pagado	Total pagado	DOUBLE		Not Null
Fecha	Fecha del pago	DATETIME		Not Null
Estado	Estado del pago	INT		Not Null
Id_usuario	Id del usuario que hizo el pago	INT		Not Null
Id_cuota	Id de la cuota	INT		Not Null
Id_prestamo	Id del préstamo	INT		Not Null

**Tabla 13. Usuarios:** En esta tabla se guardan los usuarios que tendrán acceso al sistema.

<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO</b>	<b>PK</b>	<b>NULL</b>
Id	Identificador único del usuario	INT	Si	Not Null
Tipo	Tipo de usuario	INT		Not Null
Nombre	Nombre del usuario	VARCHAR(100)		Not Null
Usuario	Usuario	VARCHAR(100)		Not Null
Clave	Contraseña del usuario	TEXT		Not Null
Estado	Estado del usuario	INT		Not Null
Fecha_creacion	Fecha de creación del usuario	DATETIME		Not Null
Fecha_actualizacion	Fecha de actualización del usuario	DATETIME		Not Null

**Tabla 14. Garantías:** En esta tabla se guardan las garantías que se van a usar para los préstamos.

ATRIBUTO	DESCRIPCIÓN	TIPO	PK	NULL
Id	Identificador único de la garantía	INT	Si	Not Null
Tipo	Tipo de garantía	INT		Not Null
Nombre	Nombre de la garantía	VARCHAR(100)		Not Null
Json_caracteristicas	En este campo se guardan las características de las garantías	TEXT		Not Null
Descripción	Descripción de la garantía	TEXT		Not Null
Comentarios	Comentarios de la garantía	TEXT		Not Null
Fecha_creación	Fecha de creación	DATETIME		Not Null
Fecha Actualización	Fecha de actualización	DATETIME		Not Null
Estado	Estado de la garantía	INT		Not Null

**Tabla 15. Historial de Usuarios:** En esta tabla se guarda el historial de las actividades de los usuarios.

<b>ATRIBUTO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO</b>	<b>PK</b>	<b>NULL</b>
Id	Identificador único del historial de usuario	INT	Si	Not Null
Accion	Tipo de acción que hizo el usuario	INT		Not Null
Descripcion	Descripción de la actividad del usuario	TEXT		Not Null
Fecha	Fecha de cuando se hizo la actividad	DATETIME		Not Null
Id_usuario	Id del usuario	INT		Not Null
Id_modulo	Id del módulo	INT		Not Null

**CAPÍTULO III**  
**DISEÑO Y DESARROLLO DEL SISTEMA.**

### **3.1. DISEÑO DEL SISTEMA.**

#### **3.1.1. ARQUITECTURA DEL SISTEMA.**

El diseño de este sistema web para la gestión y administración de los préstamos está basado en PHP, C#, JavaScript, MySQL, CSS, Bootstrap, Xampp y Postman.

### **3.2. HERRAMIENTAS Y LENGUAJES DE DESARROLLO.**

#### **3.2.1. PHP.**

PHP es un lenguaje para programar páginas de contenido dinámico. Este es un lenguaje del lado del servidor, el servidor lee el lenguaje php, y manda los datos transformados en lenguaje HTML. El lenguaje PHP se incorpora al lenguaje normal de la página (HTML) mediante scripts. Fue creado originalmente por Rasmus Lerdorf en 1995. Se ha ido ampliando con el tiempo, actualmente va por la versión 7, y trabaja conjuntamente con otros programas como son la base de datos MySQL y el servidor Apache.



*Figura 5. Logo PHP.* Fuente: <https://www.php.net/download-logos.php>.



### 3.2.2. C#.

C# es un lenguaje de programación moderno, de propósito general y orientado a objetos desarrollado por Microsoft y aprobado por la Asociación Europea de Fabricantes de Computadoras (ECMA) y la Organización Internacional de Normalización (ISO). C# fue desarrollado por Anders Hejlsberg y su equipo durante el desarrollo de .Net Framework.

C# está diseñado para Common Language Infrastructure (CLI), que consiste en el código ejecutable y el entorno de tiempo de ejecución que permite el uso de varios lenguajes de alto nivel en diferentes plataformas y arquitecturas informáticas.



*Figura 6. Logo C#. Fuente: <https://docs.microsoft.com/es-es/dotnet/csharp>.*

### 3.2.3. JAVASCRIPT.

JavaScript es una especie de lenguaje de programación ligera, interpretado por la mayoría de los navegadores y que les proporciona a las páginas web, efectos y funciones complementarias a las consideradas como estándar HTML. Este tipo de lenguaje de programación, con frecuencia son empleados en los sitios web, para realizar acciones en el lado del cliente, estando centrado en el código fuente de la página web.

JavaScript fue creado por la compañía de software “Netscape Corporación” para que fuese colocado en su navegador 2.0 y que gracias a su simplicidad, aún continúa siendo una de las herramientas de gran utilidad, para la creación de páginas web que posean algo más que texto.

Otras de las empresas que desarrollo JavaScript fue Microsoft, para su Explorer, que por lo general es similar al Netscape.

Es importante que quede claro, que JavaScript no es del todo un lenguaje de programación, sino más bien un lenguaje de script (rutinas o guiones).



*Figura 7. Logo JavaScript.* Fuente: <https://github.com/voodooitikigod/logo.js>.

### 3.2.4. MYSQL.

MySQL es un sistema gestor de bases de datos. Pero la virtud fundamental y la clave de su éxito es que se trata de un sistema de libre distribución y de código abierto. Lo primero significa que se puede descargar libremente de Internet (por ejemplo, de la dirección ([www.mysql.com](http://www.mysql.com))); lo segundo (código abierto) significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo. Esa es también la base del funcionamiento del sistema Linux, por eso MySQL se distribuye fundamentalmente para Linux, aunque también hay versiones para Windows.



*Figura 8. Logo MYSQL.* Fuente: <https://www.mysql.com>.

### **3.2.5. CSS.**

CSS es un mecanismo complementario del lenguaje HTML que permite indicarle al navegador el estilo que debe darle a los distintos elementos al desplegar la información de un sitio web.

Mientras que los comandos HTML le indican al navegador que ciertas partes del texto son títulos, subtítulos, enlaces o párrafos, CSS es lo que le dice, entre otras cosas, qué fuente, tamaño, color o alineación debe tener un elemento de la página.

Dicho de otro modo, es más fácil y rápido enlazar tu plantilla HTML a un archivo CSS para que el navegador entienda que debe desplegar todos los Títulos H2 en azul, antes que integrar un comando de estilo independiente para cada H2 que incluyas en tus páginas.

Otra de las ventajas del CSS es que puedes usar una misma plantilla para darle formato a varios sitios web, y si necesitas hacer cambios basta con modificar el elemento en la plantilla y actualizar los sitios vinculados.

### 3.2.6. BOOTSTRAP

Este framework es uno de los más populares del mercado, habiendo sido desarrollado por el equipo de Twitter. Bootstrap ha sido creado pensando en ofrecer la mejor experiencia de usuario tanto a usuarios de PC (¡IE7 incluido!), como a Smartphone y tabletas. Utiliza un grid responsive de 12 columnas y trae integrado decenas de complementos, plugins de JavaScript, tipografía, controladores de formularios y mucho más. Además, utiliza el preprocesador de CSS LESS.



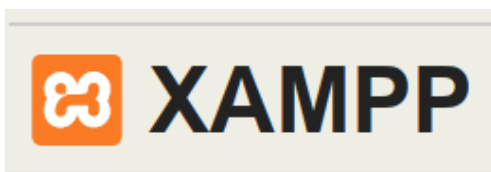
*Figura 9. Logo Bootstrap.* Fuente: <https://getbootstrap.com/docs/3.4/about>.

### 3.2.7. XAMPP.

Es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Server, etc. entre muchas cosas más.

Si alguna vez has intentado instalar Apache, sabes que no es una tarea fácil, sin embargo con XAMPP todo es diferente. Una de las ventajas de usar XAMPP es que su instalación es de lo más sencilla, basta descargarlo, extraerlo y comenzar a usarlo.



*Figura 10. Logo XAMPP.* Fuente: <https://www.apachefriends.org/es/index.html>.

### 3.2.8. POSTMAN.

Postman nace como una herramienta que principalmente nos permite crear peticiones sobre Apis de una forma muy sencilla y poder, de esta manera, probar las Apis. Todo basado en una extensión de Google Chrome. El usuario de Postman puede ser un desarrollador que esté comprobando el funcionamiento de una API para desarrollar sobre ella o un operador el cual esté realizando tareas de monitorización sobre un API.

Alrededor de la idea de testear las APIs, Postman nos ofrece un conjunto de utilidades adicionales para poder gestionar las APIs de una forma más sencilla. Es por ello que nos va a proporcionar herramientas para documentar los APIs, realizar una monitorización sobre las APIs, crear equipos sobre un API para que trabajen de forma colaborativa, convirtiendo a Postman plataforma de desarrollo de APIs que se basa por un modelo de desarrollo API First.



*Figura 11. Logo PostMan.* Fuente: <https://www.postman.com>.

### 3.3. PANTALLAS DEL SISTEMA.

#### 3.3.1. DASHBOARD.

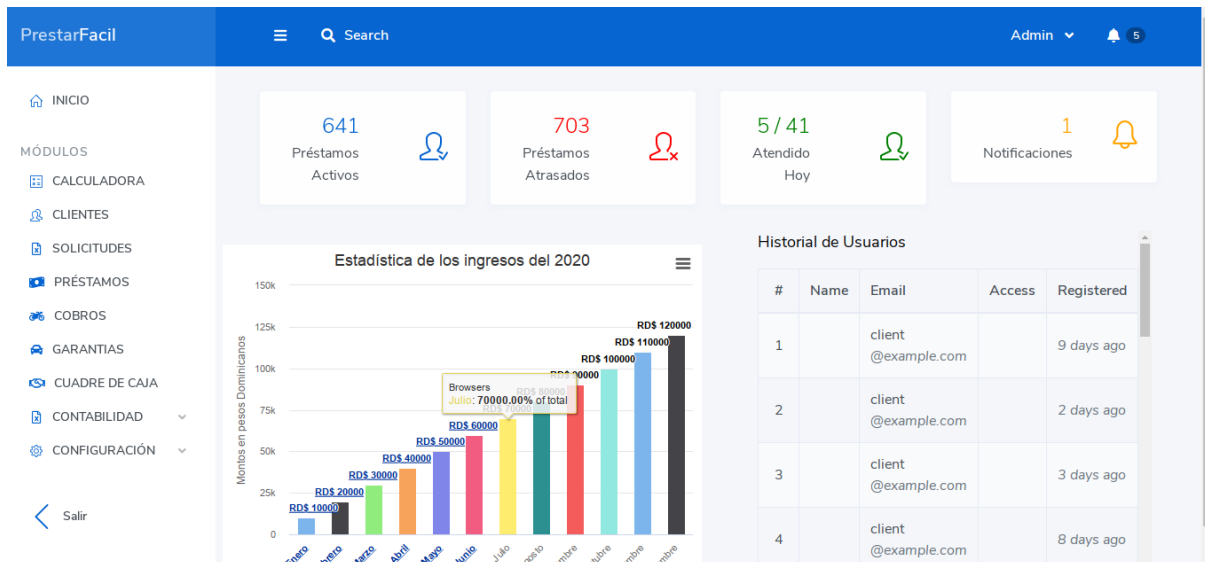


Figura 12. Imagen del Dashboard.

#### 3.3.2. LISTADO CLIENTES.

The 'LISTA / CLIENTES' page includes a search bar, 'Imprimir' and 'Excel' buttons, and a table of clients. The table has columns for actions, photo, name, last name, alias, commercial name, ID card, and phone number.

--	FOTO	NOMBRE	APELLIDO	APODO	N. COMERCIAL	CÉDULA	TELÉFONO
		Ana Josefa	Valera Ortiz		Colmado Josefa	000-0000000-0	809-000-0000
		Jose Antonio	Molina		Taller Molina	000-0000000-0	809-000-0000
		Roy	Navarrete		Rebisol S. A.	000-0000000-0	809-000-0000

Figura 13. Imagen del listado de clientes.



### 3.3.3. LISTADO PRÉSTAMOS.

PrestarFacil

☰

Search

Admin ▾

🔔

5

🏠 INICIO

MÓDULOS

📊 CALCULADORA

👤 CLIENTES

📄 SOLICITUDES

**📋 PRÉSTAMOS**

📄 COBROS

📄 GARANTIAS

📄 CUADRE DE CAJA

📄 CONTABILIDAD ▾

⚙️ CONFIGURACIÓN ▾

🔙 Salir

LISTA / PRÉSTAMOS

← Atrás

🖨 Imprimir

📄 Excel

Buscar:

--	COD. CLIENTE	COD. PRÉSTAMO	NOMBRE	CÉDULA	CELULAR	SECTOR / ZONA	CUOTAS V.	VENCIDO
	1	1	Ana Josefa	000-0000000-0	809-000-0000	ALMA ROSA	2	\$ 2,000

Figura 14. Imagen del listado de préstamos.

### **3.4. MÓDULOS DEL SISTEMA.**

El sistema estará dividido por los siguientes módulos:

- Clientes.
- Préstamos.
- Cobros.
- Garantías.
- Rutas.
- Cuadre de Caja.
- Contabilidad.
- Bancos.
- Configuraciones.

### **3.5. ROLES DE USUARIOS.**

Los usuarios que van a interactuar con el sistema son: el administrador, cobrador, contador, secretario y el soporte técnico. Estos usuarios solo tendrán acceso a su área de trabajo. A continuación, describimos los roles de cada tipo de usuario en el sistema.

#### **3.5.1. ADMINISTRADOR.**

Es el encargado de supervisar el correcto funcionamiento del negocio, Los módulos establecidos para este tipo de usuario son: reportes, estadísticas, conciliaciones bancarias, cuadros de cajas y administración de usuarios. Cuenta con todos los permisos del sistema.

#### **3.5.2. COBRADOR.**

Se encarga de realizar los cobros a los clientes, tiene acceso a los módulos de: agenda de cobros, rutas, préstamos y historial de pagos realizados. Este usuario tendrá a disposición una app móvil, por lo que puede salir a sus rutas diarias para cobrar e imprimir los recibos de pagos a los clientes.

#### **3.5.3. CONTADOR.**

Es el profesional encargado de registrar los estados económicos del negocio. Tiene acceso a los reportes, conciliaciones bancarias y estadísticas.

#### **3.5.4. SECRETARIO.**

Es la persona encargada del servicio al cliente, posee permisos para registrar clientes, solicitar préstamos, imprimir contratos, cartas de saldo y buscar estados de cuentas.

#### **3.5.5. SOPORTE TÉCNICO.**

El soporte técnico es un usuario importante en el sistema, que se encargará de resolver cualquier inconveniente que se presente, ayudar a los usuarios y en algunas ocasiones dar entrenamiento al personal. Tiene permiso a todos los módulos del sistema.

## CONCLUSIONES

La aplicación de la metodología scrum fue un pilar fundamental para el éxito del proyecto, ya que, al ajustarse a las necesidades del cliente, se logró desarrollar el sistema con mayor rapidez y eficiencia.

En el desarrollo del proyecto, decidimos dividir el sistema en frontend y backend. Esto nos permitió mitigar todos los errores que se estaban presentando y agilizar el proceso de desarrollo.

Debido a las facilidades que presentan las nuevas tecnologías, desarrollamos nuestro sistema como multiplataforma, ya que nuestro cliente solicitó una aplicación para escritorio y teléfonos móviles.

El equipo de desarrollo puso énfasis en el trabajo en equipo. Aumentamos la productividad, a dividirnos las asignaciones en las áreas que cada integrante estaba especializado.

Al finalizar el proyecto nos dimos cuenta que los lenguajes de programación que utilizamos son muy apropiados para las necesidades actuales del mercado de software y por este motivo, lo seguiremos utilizando para futuros proyectos.

Esperamos que nuestro sistema de préstamos, ayude en gran medida a los negocios de nuestros clientes, para lograr el objetivo de contribuir al mercado local e internacional.

## RECOMENDACIONES

Recomendamos utilizar la metodología SCRUM, ya que los métodos tradicionales pueden retrasar el desarrollo del proyecto.

Se recomienda utilizar el patrón de diseño MVC (modelo, vista y controlador), y dividir el proyecto en frontend y backend.

Se recomienda trabajar la base de datos en un servidor centralizado, y realizar el acceso de los datos con una api RESTful.

Es importante contar con un repositorio para el respaldo del código fuente.

Es primordial realizar respaldos semanales y mensuales a la base de datos.

El tiempo es uno de los factores más importante en el desarrollo del software, por este motivo, recomendamos crear un ambiente laboral adecuado para reducir la pérdida de tiempo.

## BIBLIOGRAFÍA

aprende-web.net, (2018). Resumen PHP. Recuperado de: <https://aprende-web.net/php/objetos/ResumenPHP.pdf>

todovisualstudionet, (2017). C# Visión General. Recuperado de: <http://todovisualstudionet.blogspot.com/2017/12/c-vision-general.html>

Conceptodefinicion.de, Redacción. (2019, Julio 26). Definición de Javascript. Recuperado en <https://conceptodefinicion.de/javascript>

Sánchez, J., (2004). MySQL guía rápida (versión Windows). Recuperado de: <http://www.cartagena99.com/recursos/programacion/apuntes/mysql.pdf>

Workana, (2018). ¿Qué es CSS?. Recuperado de: <https://www.workana.com/i/glosario/css/>

Ajgallego, (2019). Frameworks responsive. Recuperado de: [https://ajgallego.gitbook.io/bootstrap-3/capitulo\\_introduccion/capitulo\\_frameworks](https://ajgallego.gitbook.io/bootstrap-3/capitulo_introduccion/capitulo_frameworks)

Sites.google.com, (2008). ¿Qué es xampp y que necesita para ser instalado? Recuperado de: <https://sites.google.com/site/portafoliovicenciosr/poll>

Cuervo, V., (2019, febrero 14), ¿Qué es Postman? Recuperado de: <http://www.arquitectoit.com/postman/que-es-postman/>

FinSofTek Inc, (2020). Software para Prestamistas, Financieras y Cooperativas. Recuperado de: <http://programaparaprestamos.com/>

Prestan2, (2020). Sistema de gestión de préstamos, hipotecas y financiamientos. Recuperado de: <https://web.prestan2.com/>



Kpresta, (2020). Sistema de préstamos y cobros.

Recuperado de: <https://www.kpresta.com>

Prestamistapp, (2020). Sistema de administración de préstamos.

Recuperado de: <https://prestamistapp.com>

## ANEXOS

A continuación, mostramos los fragmentos de códigos más importantes del proyecto.

```
'connections' => [
    'sqlite' => [
        'driver' => 'sqlite',
        'database' => env('DB_DATABASE', database_path('database.sqlite')),
        'prefix' => '',
        'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
    ],
    'mysql' => [
        'driver' => 'mysql',
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'prefix_indexes' => true,
        'strict' => true,
        'engine' => null,
        'options' => extension_loaded('pdo_mysql') ? array_filter([
            PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
        ]) : [],
    ],
],
```

*Figura 15. Conexión a la base de datos*

```
1  <?php
2
3  /*
4  |-----
5  | Web Routes
6  |-----
7  |
8  | Here is where you can register web routes for your application. These
9  | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13 |
14 | Route::get('/', 'Dashboard_controller@index');
15 | Route::get('/dashboard', 'Dashboard_controller@index');
16 | Route::get('/estadisticas/pagos', 'Estadisticas_controller@pago');
17 | Route::get('/clientes', 'Clientes_controller@index');
18 | Route::get('/prestamos', 'Prestamos_controller@index');
```

*Figura 16. Código de las rutas de los módulos.*