
# Máster en Business Intelligence y Big Data

# Online

# 2018-2019

**Grupo 1**

Isabel Ramírez Paulino

Kelvin Javier Quezada Anazagatys

Nicacio Gómez Fernández

Starlin Francisco Gil Cruz

José Manuel Aquino Cepeda

# Contenido

# 1. Definición del Problema

**El Banco Agrícola de la República Dominicana** fue creado mediante la Ley No. 908 del 1 de junio de 1945, con el nombre original de Banco Agrícola e Hipotecario de la Republica dominicana. El Banco llena una sentida necesidad de la sociedad dominicana, relacionada con el financiamiento de las actividades productivas en la agricultura, la industria y los negocios en general.

El Banco Agrícola de la República Dominicana cuenta con reglamentos y normas para el otorgamiento de sus operaciones crediticias, las cuales están contenidas en un manual que regulan la concesión del crédito, en el que se consideran las fuentes de recursos (propios, del público y procedentes de convenios contractuales), así como la naturaleza de los proyectos a ser financiados.

Con vistas al nuevo orden mundial, esta institución ha hecho el compromiso de ponerse a tono con las normas que rigen el sector financiero, en el aspecto local, regional y universal. En consecuencia, con este documento, el Banco Agrícola da un paso de avance con miras a ser competitivo, considerando que estamos a las puertas de un mundo globalizado.

Al día de hoy, la cartera de crédito de esa entidad asciende a los 93 mil millones de pesos. La tasa de interés se sitúa entre 6 y 8%, lo que ha reducido los niveles de usura en los campos del país.

Es de las pocas naciones que suplen más del 85% de lo que consumen sus ciudadanos. En la actualidad, el 90% de lo que consumen los turistas en los hoteles es de producción nacional.

El sector agropecuario en República Dominicana tiene una considerable importancia social y económica, el área dedicada a la producción agropecuaria es de 2,6 millones de hectáreas y 242.956 dominicanos/as se dedican a este sector.

La agricultura y la ganadería representan un 8% del PIB, un 14% de la fuerza laboral y aporta alrededor de un cuarto de las exportaciones.

## 1.1 Morosidad (Definición)

Se denomina morosidad a aquella práctica en la que un deudor, persona física o jurídica, no cumple con el pago al vencimiento de una obligación.

De manera general, la condición de moroso se adquiere una vez que una obligación no es afrontada al vencimiento por parte de una persona u organización.

La morosidad hace referencia al incumplimiento de las obligaciones de pago. En el caso de los créditos concedidos por las entidades financieras, normalmente se expresa como cociente entre el importe de los créditos morosos y el total de préstamos concedidos. Así, la tasa de morosidad se define como:

Tasa de morosidad = Créditos impagados/Total de créditos.

La morosidad tiene una destacada incidencia sobre la cuenta de resultados de la entidad financiera, debido a las provisiones para insolvencias que ésta debe ir dotando para hacer frente a los posibles impagos que se vayan confirmando. Además, la entrada de un crédito en situación de morosidad implica la parada del devengo de los intereses en la cuenta de resultados.

## 1.2 Panorama General de las Posibles Causas de Morosidad

Las causas más comunes que puede ocasionar la morosidad o falta de pago a tiempo de las deudas contraídas son diversas y en cualquier caso estas pueden ser más de una a continuación enumeramos estas razones:

1. **Gestión errónea de la cobranza.**

Puede existir dentro de la institución personal poco calificado para el buen, correcto y eficiente procedimiento de cobro. Falta de control y seguimiento de las deudas existentes en la entidad. La desorganización del personal y el sistema provoca que los deudores incidan en el cumplimiento de las obligaciones contraídas. En vez de utilizar acciones para evitar que ocurra el impago, se espera a que ocurra para luego tomar carta en el asunto.

2. **Desdén del Deudor.**

Muchos deudores lamentablemente son indiferentes con las responsabilidades adquiridas, por lo mismo al saber que existen opciones de renegociaciones disponibles en la entidad financiera, pueden hacerse se la vista gorda y no pagar a tiempo solo por decisión propia que podríamos decir que es resultado de la falta de educación y hasta de respeto por el contrato establecido.

3. **Falta de tecnología apropiada.**

No contar con óptimas herramientas informáticas para la gestión de cobros, es una de las causas más vistas en el sector de las finanzas ya que son muchas las opiniones contrarias a favor de la adquisición de tecnología avanzada y a la vanguardia para el buen manejo de la cobranza y la previsión a tiempo de los posibles riesgos.

4. **La naturaleza.**

En el caso de la producción agrícola encontramos inconvenientes como cambios climáticos, inundaciones, huracanes, sequia, plagas y un gran número de factores meteorológicos propios de la tierra, el suelo y el clima, que afectan la eficiente productividad de las áreas financiadas y por lo mismo podrían incurrir en impagos. Este sería el punto principal de análisis al momento de implementar el modelo de scoring sugerido para optimizar la predicción de la morosidad.

## 1.3 Delimitación del Problema

El acceso a financiación es clave para promover la expansión y la competitividad de la agricultura en general. Un factor que ha tenido efectos negativos en el sector ha sido el incremento de los precios de los fertilizantes e insumos necesarios para la producción. A esto se suma que el 64% de los agricultores cultivan no más de cinco hectáreas. A esto se suma las amenazas y la vulnerabilidad del sector a los efectos climatológicos como huracanes, ciclones tropicales y sequías, y otras adversidades como las plagas y enfermedades, estos son causa del incremento de la morosidad o incumplimiento del pago del compromiso contraído.

Como se enumeró anteriormente, el punto 4 es el característico a las necesidades de los clientes (productores agrícolas y banco agrícola) más sobresaliente para la implementación del modelo de scoring.

Además de tomar como punto principal todas las características descriptas anteriormente (fenómenos naturales, plagas, etc.), no se deberá dejar de lado la posibilidad de que la morosidad sea producida deliberadamente por el propio cliente al incurrir en este hábito sin motivos de fuerza mayor.

En este punto se centra la implementación del sistema de scoring que se propone crear específicamente para esta institución, tomando en cuenta factores propios y característicos de este sector (agropecuaria) clientela número 1 del BANCO AGRICOLA DOMINICANO, un sistema proactivo capaz de anticiparse y reaccionar ante los incumplimientos y demoras de los clientes, analizando de forma anticipada el comportamiento del deudor tanto en la banca como el destino de inversión del préstamo de donde se supone se obtendrán las ganancias tanto para el cliente como para el propio banco.

## 1.4 El Banco Agrícola y la Morosidad

En la actualidad el banco agrícola dominicano no cuenta con herramientas para la predicción a tiempo y eficaz para detectar estas situaciones que suponen un inconveniente latente en esta institución y en cualquier otra institución financiera.

Las medidas tomadas en la entidad al momento de enfrentar la morosidad son reactivas al tener que esperar que la misma se presente para proceder a actuar. Lo ideal es implementar soluciones innovadoras y proactivas como la que ofrece el sistema (BAGRI SCORING), para de esta forma estar siempre preparados para brindar apoyo a los clientes y mantener dentro de la empresa un nivel alto de seguridad y confianza al estar situados un paso delante de los acontecimientos.

La morosidad afecta tanto a la institución, como al cliente y es un suceso que influye negativamente en la banca nacional de la republica dominicana.

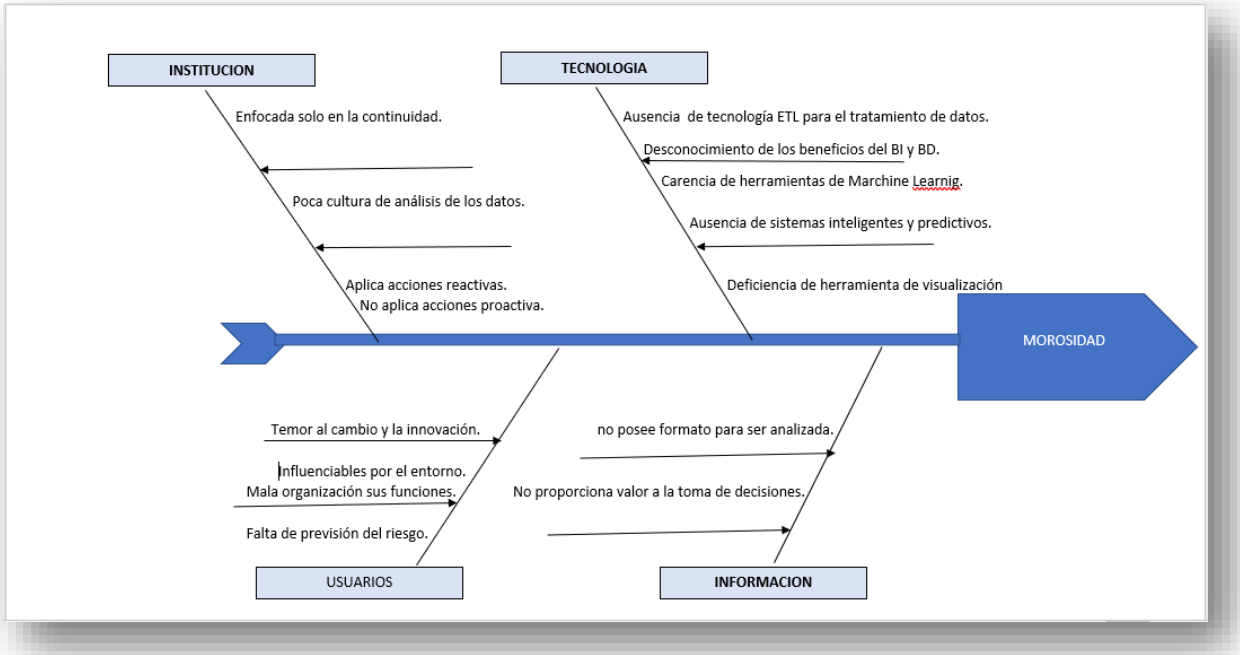



*Ilustración 1*

# 2. Validación de Hipótesis

Hipótesis: estudiando las políticas de los procedimientos para asignación de crédito de la entidad bancaria, hemos visto la necesidad de reforzar dichos procedimientos para reducir la asignación de préstamos a clientes "malos" y consecuentemente disminuir las tasas de pérdidas per cápita para la institución Banco Agrícola Dominicano.

Se ha identificado una deficiencia en todo el proceso desde la depuración de clientes para asignación de préstamos, hasta la identificación de impagos o créditos morosos dentro del Banco Agrícola Dominicano. Las informaciones sobre morosidad son obtenidas luego de que ya están incurriendo en este hecho, donde lo que corresponde es tomar medidas reactivas, no medidas de prevención, lo que se traduce en pérdidas considerables para la institución.

En vista de lo anteriormente expuesto proponemos la implementación de un sistema especializado para la prevención y disminución de la asignación de préstamos a personas que no tengan la capacidad de pagar las cuotas establecidas y en vista de encaminar a la empresa en el desarrollo tecnológico, estaremos utilizando fundamentos y buenas prácticas del BI, BD y Machine Learning para enfrentar de forma proactiva estos acontecimientos teniendo manejo y control para prevenir e influir en la selección de los mejores clientes para la institución.

## 2.1 Identificación de Hipótesis a Validar

**Hipótesis de cliente**

- El departamento de tecnología no cuenta con sistemas BI, BD.
- El departamento de tecnología no cuenta con personal BI, DB.
- El departamento de riesgo no cuenta con herramientas de scoring automatizadas.
- La correcta asignación de créditos disminuye el aumento de la morosidad.

**Hipótesis de Problema**

El Banco Agrícola Dominicano al igual que otros bancos se ve afectado con los atrasos o falta de cumplimiento de la clientela con respecto a los compromisos adquiridos. Esta es una problemática que llega a ser detectada solo cuando ya está ocurriendo, es decir, no se prevé la probabilidad de incumplimiento de los préstamos otorgados, lo ideal sería poder identificar las posibles moras a originarse para dependiendo de las características de las mismas aplicar planes correctivos que serían de gran ayuda tanto para la entidad como para el cliente que ha sido beneficiado con el crédito.

Realizar una depuración efectiva del cliente, para tratar de obtener la mayor probabilidad de "buenos", se hace una necesidad cada vez más urgente, ya que este paso representa el aseguramiento del retorno del crédito concedido.

**Hipótesis de producto/solución**

Para la solución a dicho problema estaremos implementado un sistema basado en Machine Learning y tecnologías de Big Data para detectar los posibles atrasos en los pagos de los créditos, agilizar los procesos de asignación de créditos, aumentar la efectividad en una medida considerable al momento de recuperar los créditos otorgados por la institución sin que se vea afectada la tasa interna de retorno para el banco.

## 2.2 Proceso de Validación (entrevistas y encuestas)

Para determinar si la hipótesis del por qué la morosidad se presenta en una entidad financiera, se creó el siguiente formulario de forma digital el mismo constaba de las siguientes interrogantes:

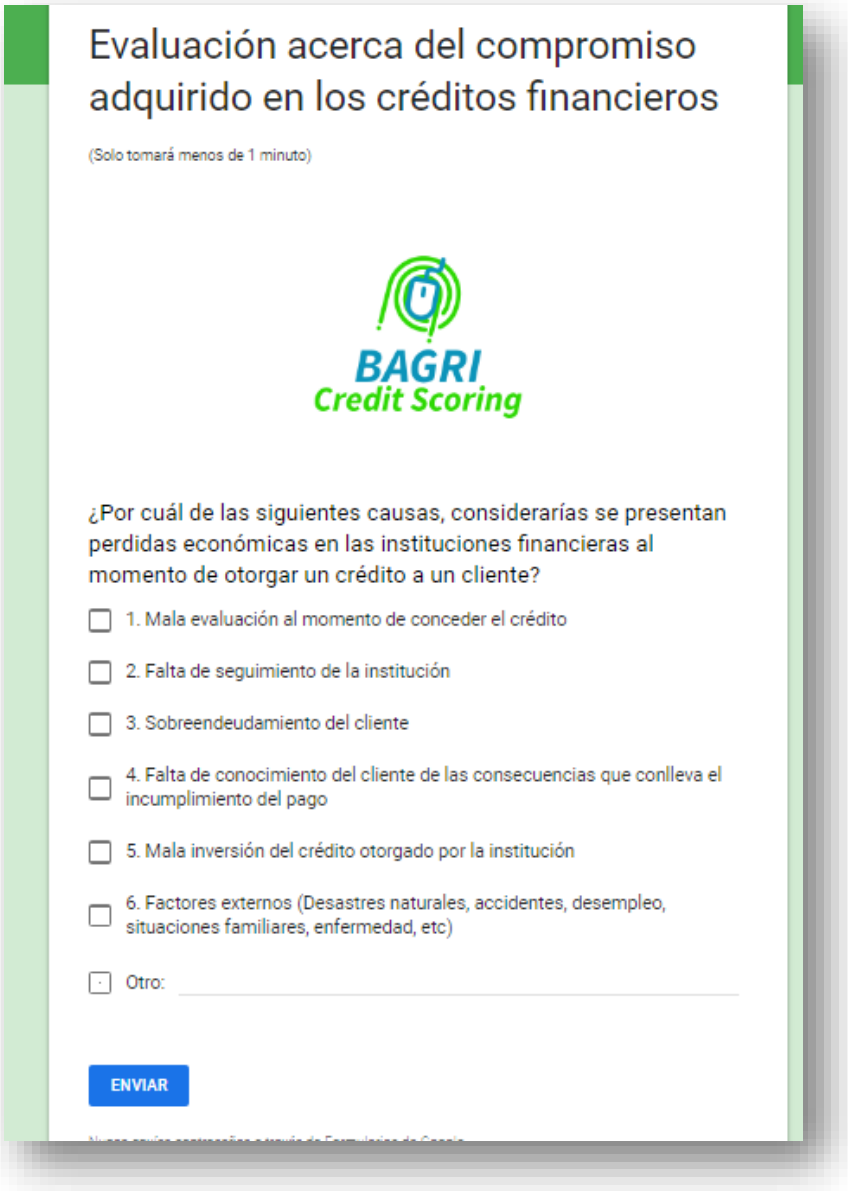



*Ilustración 2*

**Resultados:** los resultados obtenidos a través de la encuesta digital elevo dos afirmaciones que se aprecian a continuación y que reflejan y confirman la importancia de tomar acciones preventivas y de seguimiento aplicando la inteligencia de negocios desde el primer paso que es el otorgamiento del préstamo, además de que obviamente el incremento de las deudas pendientes de un individuo si no cuenta con el respaldo monetario correspondiente lo hará incumplir en sus responsabilidades.

- Sobreendeudamiento del cliente con 49 %.
- Mala evaluación al momento de conceder un préstamo 47.9 %.

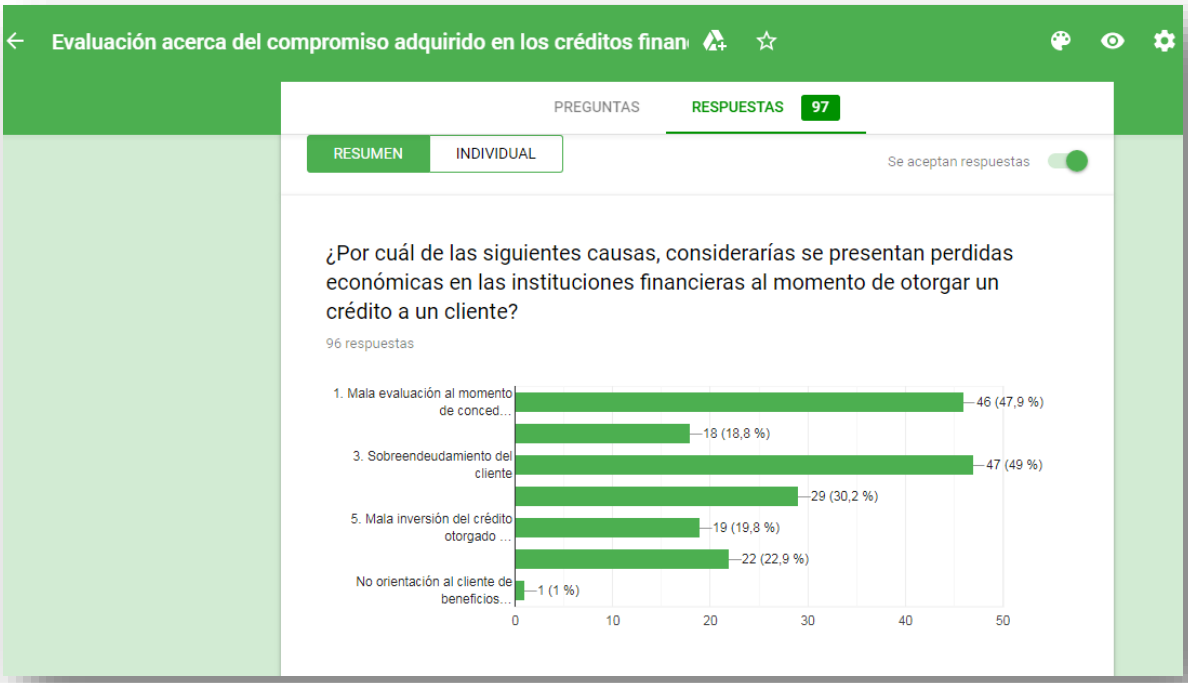



*Ilustración 3*

### *Entrevistas realizadas en el Banco Agrícola Dominicano.*

- Entrevista realizada al señor Jorge Aquino Furcal, MSC. Encargado Sección de Implementación y Desarrollo de Sistemas Banco Agrícola de la República Dominicana:

***Considera de utilidad la implementación de un sistema de scoring crediticio.***

*"Entiendo que un scoring crediticio agregaría agilidad al proceso de análisis de crédito, de un potencial prestatario (Solicitante de un crédito), además de que reduciría los márgenes de error y la posibilidad de colocar el dinero en manos con muy altos riesgos, la implementación de una aplicación de scoring crediticio sería de mucha utilidad para cualquier institución financiera al momento de analizar la posibilidad del otorgamiento de un crédito."*

## 2.2. Análisis Preliminar de Datos

Conociendo los datos con los que se realizara el proyecto:

- **¿De qué datos disponemos?**

Datos históricos correspondientes a un rango de 2 años 2017-2018

- **¿Qué datos a los que tenemos acceso no se están recogiendo?**

No se están recogiendo de forma obligatoria muchos datos que pueden ser significativos para la evaluación del cliente, la empresa ha comenzado a capturarlos, pero se encuentra a expensas de los usuarios del sistema si los ingresan o no, por esta razón existe gran deficiencia de las siguientes informaciones:

- Edad del cliente.
- Ingresos monetarios del cliente.
- Lugar donde labora el cliente.
- Tiempo en la empresa que labora el cliente.

**¿Qué datos pueden generarse a partir de nuestros productos y operaciones?**

La elaboración del sistema de scoring desde sus inicios con la sola investigación arroja necesidades no cubiertas de la empresa en lo que se refiere a la captura de datos.

A través de este sistema y la utilización del mismo se irán generando interrogantes a desglosar y por lo mismo se harán cada vez más claras las necesidades requeridas.

**• ¿Qué datos podríamos obtener de otros que nos serían de utilidad?**

Información personal del cliente a la que no disponemos con claridad y obligatoriedad.

**• ¿Qué datos que tienen otros podríamos usar en una iniciativa conjunta?**

Este proyecto de scoring es estrictamente concebido con datos internos de la entidad financiera por lo que en la elaboración del modelo que se pretende implementar no haría falta buscar informaciones externas, de implementarse otros modelos sí podrían requerirse datos externos dependiendo de lo que se busque identificar.

**• ¿Cómo podríamos estructurar y analizar nuestros datos para generar mayor valor?**

Sometiendo a los datos a una rigurosa fase de ETL y aplicándoles un análisis exploratorio.

Tomando siempre en cuenta que se pretende resolver y como.

**• ¿Estos datos son valiosos internamente para nosotros, o para nuestros clientes actuales, o para nuevos clientes potenciales, o para otras industrias?**

Estos datos son de gran valor para la empresa (cliente), para los clientes de la entidad financiera, para el sector en general y para la elaboración del proyecto en sí pues depende de estos.

## 2.3. Análisis Entorno

Factores externos que influyen en el desarrollo del proyecto:

**• Factores político-jurídico:**

Actualmente el ambiente político-jurídico de república dominicana está encausado en la búsqueda de la implementación de las buenas prácticas aplicadas con metodologías agiles, modernas y actuales por lo que por esta parte el proyecto estaría influenciado de manera positiva.

**• Factores culturales:**

En cuanto a lo cultural y como esto afecta al desarrollo del proyecto, lo más difícil podría ser que, aunque el departamento de tecnología quien es el colaborador directo esta consiente y en completo apoyo al proyecto, también intervienen otros departamentos que podrían mostrase escépticos a lo que significa implementar machine learning en la institución. Personas con

desconocimiento de las tecnologías actuales y trabados en procesos arcaicos que dependen meramente del usuario que interviene con el cliente y utiliza la tecnología meramente como herramienta en vez de comprender que esta es un aliado y un gestor de procesos automatizados e inteligentes. En fin, la mentalidad cerrada de miembros de la empresa puede causar trabas al proyecto BAGRI CREDIT SCORING.

• **Factores económicos:**

La economía es un factor que influye en cualquier proyecto y la tecnología no es la excepción, ya que indudablemente cualquier desarrollo o implementación de un plan de tecnología en una empresa implica un gasto muchas veces muy alto y no solo en hardware y software sino también en personal calificado.

• **Factores socio-demográficos:**

El mercado tecnológico es un área en constante crecimiento y evolución en todo el mundo, además de que interviene en todas las otras áreas del hacer (medicina, finanzas, educación, ventas, etc.). El desarrollo de un proyecto de Big Data y Machine Learning significan la inclusión de la entidad bancaria en sector de utilización de prácticas modernas y actuales lo que significaría un avance digital.

• **Factores tecnológicos:**

La entidad donde se desarrollará el proyecto no cuenta con la tecnología adecuada para implementar Big Data y Machine Learning, siendo estas últimas ciencias relativamente nuevas en la republica dominicana.

• **Factores medio ambientales:**

Identificar la disponibilidad de recursos naturales y de infraestructura, conocer el impacto ambiental, conocer las posibilidades, incentivos y restricciones derivadas de las reglamentaciones que regulan la conservación del medio ambiente y que pueden afectar a la actividad de la empresa.

## 2.4. Análisis Competitivo

Entender cuáles son los competidores y qué propuesta de valor ofrecen, es importante para identificar y diferenciar nuestra propuesta de valor frente a lo que actualmente existe en el mercado o identificar que otras soluciones de Big Data se han implantado en las empresas competidoras de nuestro cliente.

Nuestro cliente (Banco Agrícola Dominicano) nunca ha implementado ninguna solución de Big Data en la empresa, por lo que la implementación del proyecto (Bagri Credit Scoring) significaría la primera relación de la institución con este tipo de tecnologías.

Nuestra propuesta de valor es la mejora de un proceso que un inicio es la predicción de la morosidad, pero será un sistema escalable que podrá adaptarse a las necesidades requeridas por el banco para responder interrogantes que hoy por hoy el sistema que utiliza la entidad lo puede hacer.

El sector dominicano en cuanto a este tipo de tecnologías cuenta con diversos gestores de Business Intelligence y Big Data que proporcionan a las entidades bancarias de forma separa a la institución, es decir, que contratan los servicios de los mismos (por ejemplo, para lo que es la depuración del cliente) y esto se podría dar por la resistencia de las empresas en cambiar sus estructuras para adaptarlas a las áreas tecnológicas antes mencionadas.

**Propuesta de Valor**

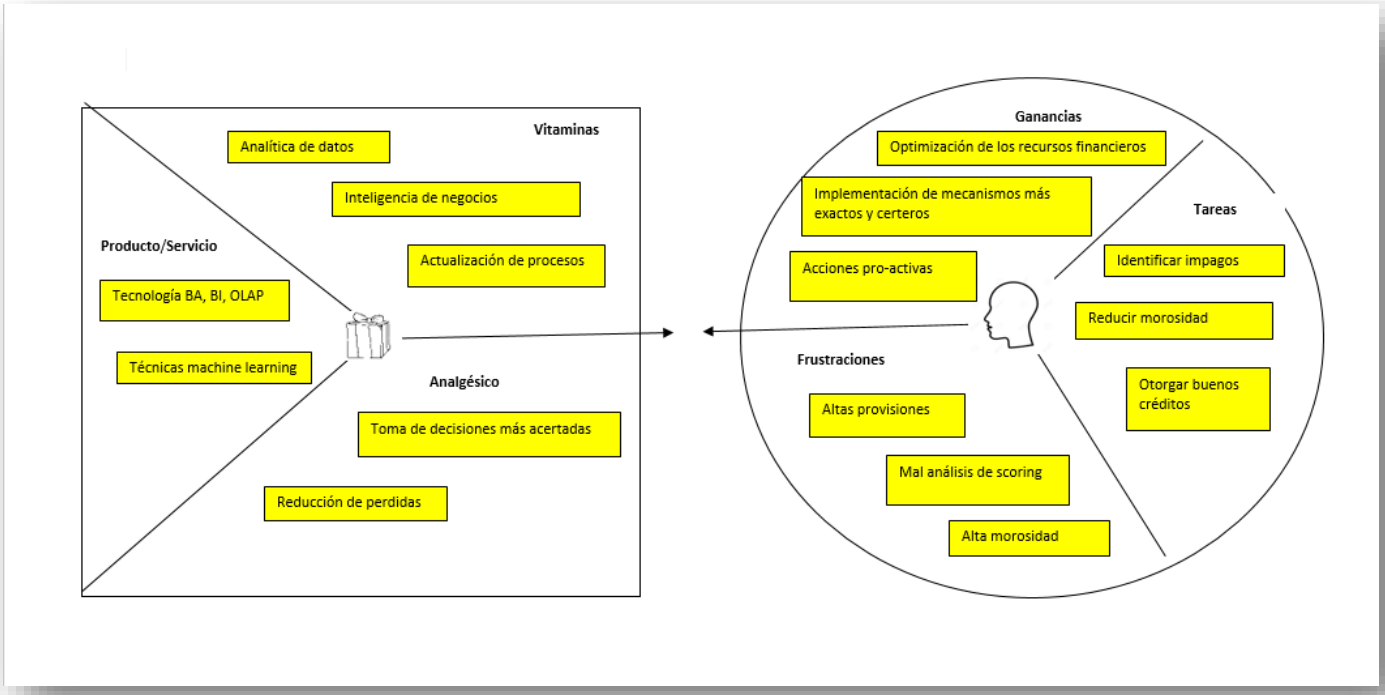



*Ilustración 4*

## 3. Análisis y Diagnóstico

Bagri Credit Scoring es una solución de Big Data y Machine Learning que se desarrolla para el Banco Agrícola Dominicano buscando optimizar y mejorar el proceso de control, monitoreo y seguimiento de los préstamos concedidos por el mismo.

La probabilidad de impagos es una situación que solo se refleja después de que ocurre o si el personal correspondiente se percata de esto, la implementación de BCS significara un paso adelante de estos acontecimientos, buscando prever e identificar los créditos malos ya concedidos por el banco.

# *DAFO*

**INTERNO**

**Fortalezas:**

Modelo basado en técnicas Machine Learning y Biga Data que busca la innovación en los procesos.

El tema de la morosidad y la determinación de los impagos es de suma importancia para el buen desenvolvimiento de la institución para la cual se desarrolla la solución.

**Debilidades:**

Falta de datos relevantes que no son recogidos en la institución de forma obligatoria.

Trabajar con categorías específicas y limitadas por ser un proyecto a la medida para una entidad bancaria determinada.

**EXTERNO**

**Oportunidades:**

Implementación de un sistema que ayuda a aplicar mejores prácticas para prevenir inconvenientes operativos.

Aplicación de nuevas tecnologías antes no utilizadas en la entidad financiera.

Escalabilidad del proyecto ***Bagri Credit Scoring*** adaptándose a las necesidades de los diferentes departamentos de la empresa.

**Amenazas:**

Incertidumbre en la institución a implementar y aplicar nuevas tecnologías.

Temor en la institución de cambiar la forma tradicional de realizar un proceso.

***Ilustración 5***

En el mercado dominicano estas tecnologías (Big Data, Machine Learning) se puede decir que son emergentes y no se utilizan en la mayoría de las entidades financieras, sin embargo, lo que si podemos encontrar es la oferta de empresas particulares que ofrecen estos servicios a los bancos.

Bagri credit scoring es una solución a la medida para el banco agrícola dominicano que pretende maximizar los resultados en la detención de los niveles de casos de impagos dentro de la cartera de préstamo y también constatar cómo esta se encuentra frente a la situación del banco.

## 3.1 Definición Modelo de Negocio

Bagri Credit Scoring es la optimización de proceso e implementado para ser escalable a toda área de la empresa

1. **Segmentos de clientes:** El banco agrícola dominicano es nuestro cliente directo y especifico, departamento de riesgo y tecnología, pudiendo ampliarse y moldearse a cada departamento existente.

2. **Propuesta de valor:** Implementar una solución analítica predictiva al proceso de depuración del cliente, asegurando la obtención de decisiones más certeras y utilizando tecnologías innovadoras, lo que supone mejora y maximización en las labores y en las herramientas.
3. **Canales:** La comunicación con nuestro cliente (Banco Agrícola Dominicano), será directa con los departamentos involucrados.

4. **Relación con el cliente:** Trato directo y personalizado debido a trabajar con personal interno de la empresa.

5. **Fuente de ingresos:** El proceso de implementación, desarrollo y continuidad el proyecto implica la captación de ingresos provenientes del cliente donde se aplique la solución.

6. **Recursos clave:** Las herramientas (plataforma, lenguaje, técnicas) que deben utilizar los desarrolladores del proyecto para gestionar sus labores.

7. **Actividades clave:** El trabajo de profesionales en las áreas de aplicación del proyecto Determinar datos a utilizar.

8. **Socios clave:** La naturaleza del proyecto hace que el Banco Agrícola Dominicano sea el socio numero 1 ya que la solución está realizada a la medida con miras de optimización. Además, los proveedores de plataforma, software y servicios para la ejecución del proyecto.

9. **Estructura de costes:** Como todo proyecto, la implementación del mismo en lo que se refiere a herramientas y los profesionales que utilizaran las mismas, supone una inversión.

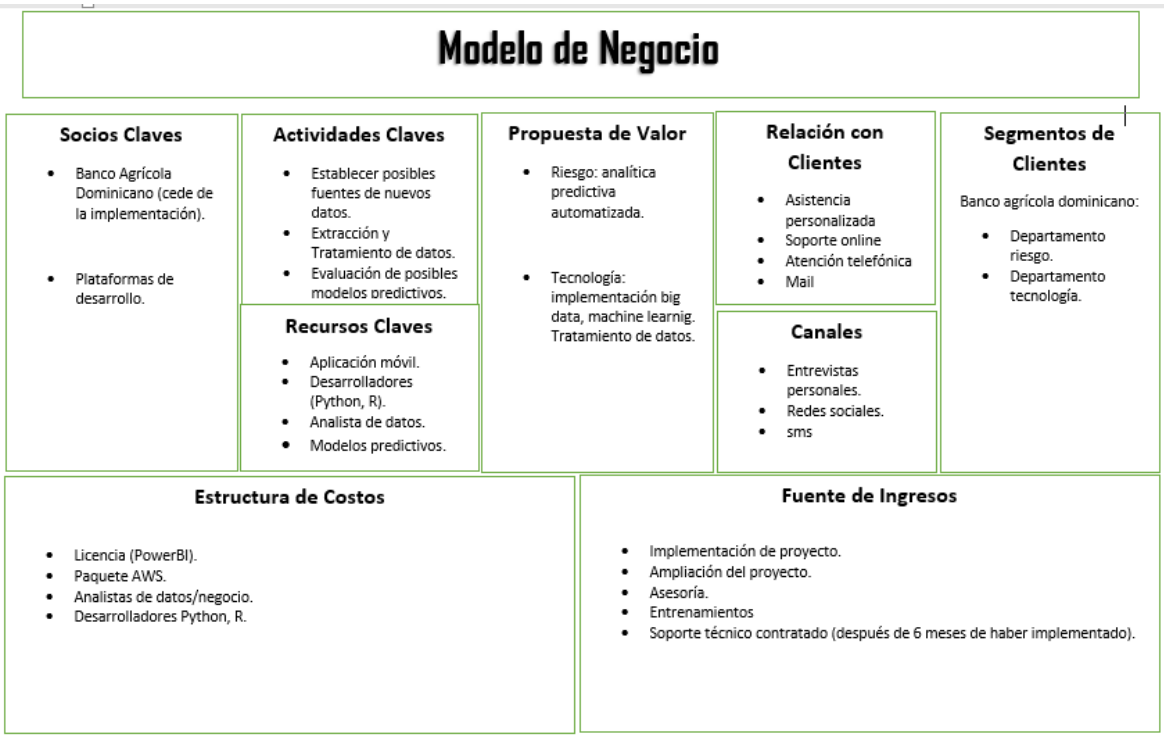



*Ilustración 6*

## 3.2 Plan de Acción

Bagri Credit Scoring busca optimizar el proceso de prevención de impagos, determinación de posibles malos entre los clientes y reducción de la morosidad. Esto temas son propios y comunes del sector bancario y por lo tanto son una situación del día a día.

El Banco Agrícola Dominicano busca encaminar sus operaciones de riesgo a un nivel tecnológico más avanzado y propio de los nuevos tiempos, para esta tarea se ha diseñado una solución de machine learning.

**Bagri Credit Scoring**

| |
|---|
| **¿Qué? Implementación de un sistema de analítica predictiva, que permitirá tener control y predicción de posibles aparición de impagos en el sistema bancario (Banco Agrícola).** |
| **¿Quién? Será desarrollado por profesionales de áreas de tecnología y negocio (analista, desarrollador), en base a las situaciones acontecidas a la entidad financiera.** |
| **¿En qué momento? Este proyecto será desarrollado durante un tiempo de 4 meses en el banco agrícola dominicano el cual dará lugar a un sistema** |

| |
|---|
| **de análisis inteligente de la vida de los créditos otorgados por el banco y este modelo podrá escalar a otros departamentos.** |
| **¿Qué recursos? Recursos humanos como los son los developers y analistas, recursos de hardware computadora con características específicas.** |
| **¿Comunicación? Los datos y las informaciones para la elaboración del proyecto han sido suministradas por el departamento de tecnología del banco agrícola, quien mantiene una contacto directo y constante con los implementadores del sistema Bagri Credit Scoring. Todo el avance del proyecto es compartido con el departamento de tecnología del banco.** |

***Ilustración 7***

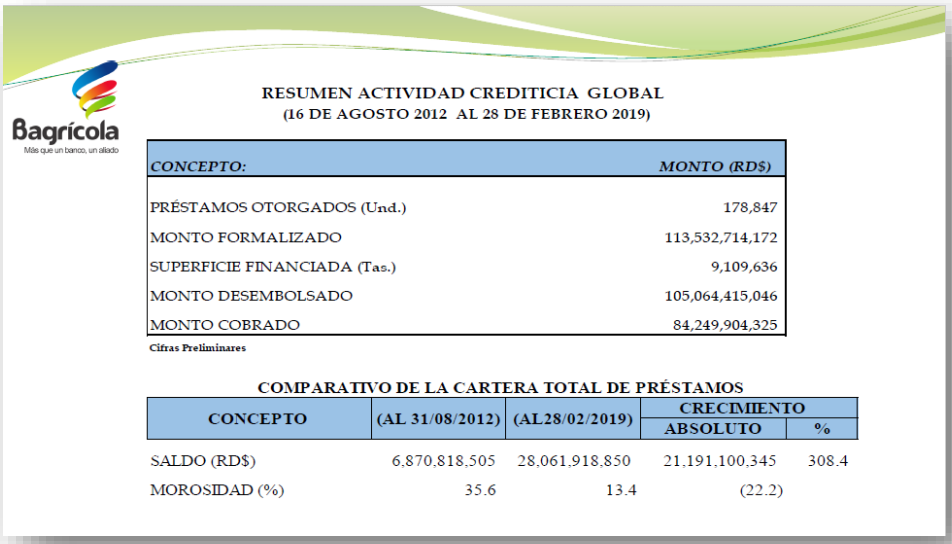
**Alcance:** detención y prevención de los créditos impagos y evaluación de la morosidad.

**Objetivos:**

- Optimizar el proceso de seguimiento y control de los créditos mediante al análisis de modelos predictivos de los datos de la institución.
- Incrementar el roí de la entidad en cuanto a créditos otorgados, maximizando las probabilidades de la recuperación de la inversión.
- Identificar y Disminuir la morosidad mediante técnicas preventivas, que ayudaran a conocer día con día el comportamiento de los clientes mediante el manejo que tengan con sus deudas pendientes.
- Reducir la intervención de la opinión o percepción del usuario, otorgando al banco un sistema que tomara decisiones independientemente de variables producidas por la empata o la falta de esta.
- Proveer al banco de un sistema que será escalable en toda la entidad, pudiendo aplicarse para otros departamentos y no solo en riesgo o tecnología.

**Información:**

Los objetivos del proyecto serán cuantificados mediante los datos recopilados por la entidad (Banco Agrícola Dominicano), los cuales exponen una panorámica de la situación del banco en relación a los créditos impagos y pagos del total otorgado.



RESUMEN ACTIVIDAD CREDITICIA GLOBAL
(16 DE AGOSTO 2012 AL 28 DE FEBRERO 2019)

| CONCEPTO: | MONTO (RD$) |
|---|---|
| PRÉSTAMOS OTORGADOS (Und.) | 178,847 |
| MONTO FORMALIZADO | 113,532,714,172 |
| SUPERFICIE FINANCIADA (Tas.) | 9,109,636 |
| MONTO DESEMBOLSADO | 105,064,415,046 |
| MONTO COBRADO | 84,249,904,325 |

Cifras Preliminares

COMPARATIVO DE LA CARTERA TOTAL DE PRÉSTAMOS

| CONCEPTO | (AL 31/08/2012) | (AL28/02/2019) | CRECIMIENTO | |
|---|---|---|---|---|
| | | | ABSOLUTO | % |
| SALDO (RD$) | 6,870,818,505 | 28,061,918,850 | 21,191,100,345 | 308.4 |
| MOROSIDAD (%) | 35.6 | 13.4 | (22.2) | |

*Ilustración 8*

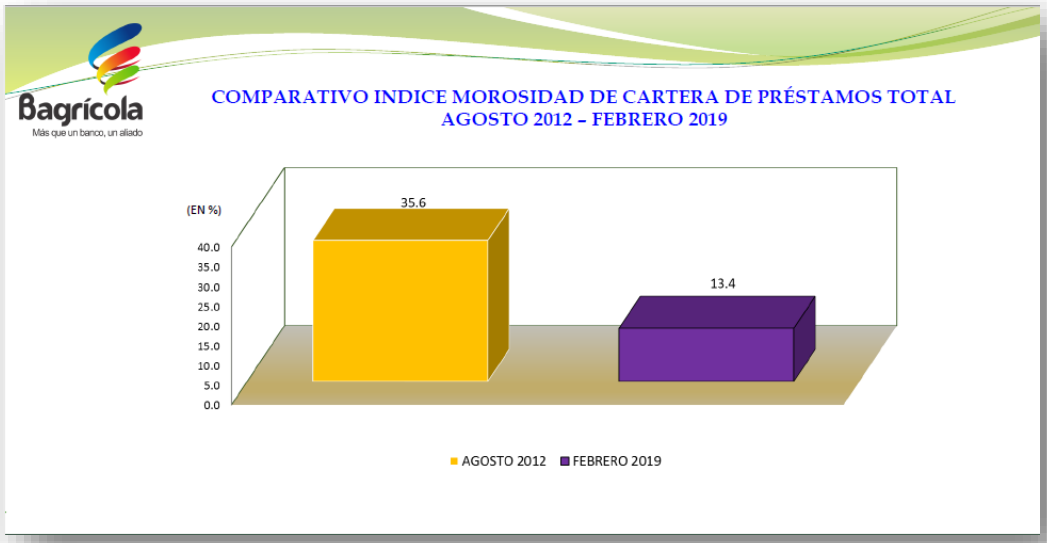



*Ilustración 9*

**Acciones:** actividades que se van a tomar para optimizar el resultado de la decisión:

- Obtención de datos históricos de comportamiento de clientes en el pago.
- Tratamiento de los datos obtenidos, mediante proceso de ETL donde se identificarán las variables que realmente aportan valor (predictivas) y cuales variables fuentes necesarias ser colectadas a futuro para una mejor evaluación del cliente y control de su comportamiento.

- Desarrollando un modelo de scoring de tipo comportamiento, el cual brindara evaluación continua sobre el cliente, pudiendo este aprender de sus errores y de nuevos datos ingresados.
- Automatización del análisis de cobros, determinando con anticipación los posibles impagos que pueden surgir.

El banco Agrícola posee un Plan estratégico 2017-2020, el mismo expone en su página número 43, que se contempla el cobro de los montos impagados agilizando la gestión del sistema de cobro.

CUADRO No. 8
PROYECCIONES PROGRAMA DE COBROS
(EN MILLONES DE RD$)
AÑOS 2017 - 2020

| RENGLONES | AÑOS | | | |
|---|---|---|---|---|
| | 2017 | 2018 | 2019 | 2020 |
| Préstamos a Vencer | 4,995.8 | 5,245.6 | 5,507.9 | 5,783.3 |
| Cuotas a Vencer | 2,336.7 | 2,453.5 | 2,576.2 | 2,705.0 |
| Cuotas Vencida | 687.4 | 721.8 | 757.9 | 795.8 |
| Cartera Vencida | 946.1 | 993.4 | 1,043.1 | 1,095.2 |
| Programa Préstamos | 5,873.2 | 6,166.9 | 6,475.2 | 6,799.0 |
| **TOTAL** | **14,839.2** | **15,581.2** | **16,360.2** | **17,178.2** |

***Ilustración 10***

## 3.3 Definición del Alcance del Proyecto: objetivos y métricas

Este proyecto está orientado hacia la optimización de los procesos en el área de riesgo financiero de la institución. Por un lado, se pretende realizar un estudio de la cartera de créditos y del departamento con el fin de conocer los procesos que se llevan a cabo en la evaluación y toma de decisiones para el otorgamiento de créditos, y así proponer mejoras para la implementación y captura de los datos. Por otro lado, se construirá una herramienta BI que les permitirá conocer de forma cualitativa la clasificación y probabilidad de riesgo de los clientes persistentes y nuevos, además de proveer un cuadro de mandos que desplegará los estados de las carteras en tiempo real, traduciéndose en un incremento de la asertividad en la toma de decisiones y la rentabilidad.

Para llevar a cabo la elaboración de la herramienta BI, estableceremos dos fases:

En la primera fase contrataremos un servicio de computación en la nube el cual será encargado de realizar todo el proceso. Implementaremos en el un Data Lake que será cargado mediante un proceso ETL. Para ello utilizaremos la herramienta Kettle ETL que ofrece la suite de pentaho, para realizar el proceso de extracción, transformación y carga de las carteras de créditos de clientes de la institución. Estos datos serán de relevancia para el estudio en el periodo inicial, y a futuro

impactaran en el mejoramiento del entrenamiento de los modelos ya que el Data Lake mantendrá su crecimiento en función a la cartera de créditos.

En la segunda fase desarrollaremos la herramienta BI, para esto inicialmente realizaremos un análisis exploratorio de los datos utilizando Python, tomando el histórico de un periodo de 2 años y lo apoyaremos con datos de otras entidades del mismo fin. De esta forma, separaremos las variables en numéricas y categóricas e identificamos características como: variables que impactan en la morosidad, valores atípicos y formas de distribución que nos permitirá precisar el modelo teórico a utilizar.

Una vez obtengamos las variables, realizaremos un estudio de las mismas y su totalidad para determinar las variables de negocio que podrían impactar sobre la decisión del crédito. En el mismo sentido, realizaremos un análisis univariante para inferir la potencia predictiva de las variables, es decir, qué tan predictivas son y cuales aportan más a la determinación de posible cliente moroso. A continuación, aplicaremos técnicas de regresión para determinar la correlación que existe entre las variables y tomar las de mayor grado de asociación, discriminando así grupos o variables de menor interés. Posicionándonos en este punto, realizaremos la modelización aplicando las técnicas de datamining y machine learning respectivamente para entrenar y seleccionar el modelo más eficiente para nuestra solución.

**Objetivo general:**

Implementar una herramienta BI para el análisis de riesgo crediticio y optimizar el proceso de acreditación que emplea la entidad financiera “Bagricola”, con el fin de incrementar la generación de la cartera de créditos de clientes en calidad de cumplimientos, reducir las pérdidas y mejorar las condiciones de acceso para este sector.

**Objetivo específicos:**

- Clasificar la cartera de crédito en calidad de morosidad de los clientes.
- Analizar las metodologías de evaluación de riesgo existentes.
- Proveer herramienta BI para la determinación en tiempo real de la calidad del cliente.
- Reducir el riesgo crediticio.
- Incrementar los beneficios.

**Indicadores**

Para dar un seguimiento de forma eficiente de la implementación de nuestra herramienta para el análisis de riesgo crediticio en el banco agrícola, se han definido indicadores clave de rendimiento o KPIs. Estos indicadores nos ayudarán a cuantificar el rendimiento en función de la consecución de los objetivos propuestos.

**Mejora de calidad de la cartera de crédito:** Se realizará una comparativa entre la evolución de dicha cartera en sentido de cumplimiento de pagos, una vez sea implantada la herramienta. De esta forma, se comprobará la eficiencia de los modelos de predicción para determinar el riesgo de morosidad de los clientes en función de los datos históricos.

**Ahorro de tiempo en los procesos:** Con la finalidad de cuantificar el tiempo que toma al personal realizar los diversos análisis e investigaciones para determinar la fiabilidad de un cliente, realizaremos varias simulaciones enfocados en este punto, de manera que determinemos el valor en calidad / tiempo que Bagri Credit Scoring aporta.

**Aumento de clientes anuales:** Cuantificamos los resultados de la implementación de la herramienta y realizaremos una comparativa entre los años pasados.

A continuación, se desglosan los indicadores claves de manera esquematizada, en esta se definen los objetivos, indicadores, optimizaciones y tiempo esperado para la consecución de los objetivos.

| Objetivo | Indicador | Actual | Target | % | Tiempo |
|---|---|---|---|---|---|
| Optimización de los procesos de acreditación | Mejora de las tomas de decisiones | -- | +70% | -- | 3 meses |
| | Reducción de tiempos de los procesos | 3 horas | 1 hora | 300 | 3 meses |
| | Formación de personal | -- | 5 | 500 | -- |
| Implementación de herramienta BI | Incremento de clientes anuales | -- | +15% | -- | 18 meses |
| | Estandarización de las carteras | -- | 60% | - | 3 meses |
| | Incremento en la calidad de los créditos | -- | 70% | -- | 12 meses |

***Ilustración 11***

## 3.4 Aplicación Web

Hemos diseñado una aplicación web como parte de nuestro prototipo de solución BI, el cual permitirá a la institución evaluar las solicitudes de sus clientes para determinar, en base a un histórico de la cartera de créditos y criterios establecidos por los mismos, la probabilidad de que un cliente pueda incurrir en morosidad. Lo que se traducirá como una mejora en las tomas de decisiones para el otorgamiento de créditos.

La aplicación es accesible a través de la siguiente uril: http://35.236.51.89:8888/scoring/

A continuación, una captura de la aplicación luego de validar el estado crediticio de un determinado cliente.

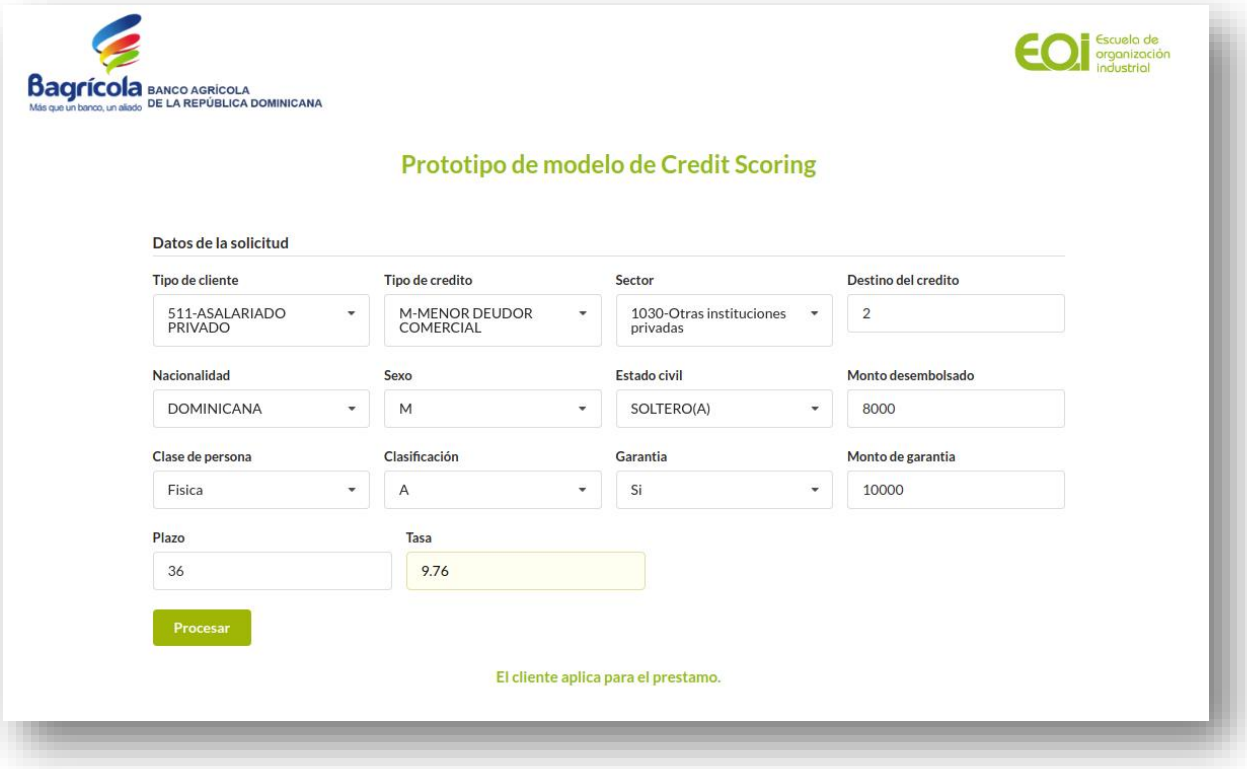



*Ilustración 12*

**Criterios de Aceptación del Riesgo**

El criterio de aceptación o rechazo para las solicitudes dependen de la experiencia de la unidad de riesgo de la entidad financiera. El score que arroja la aplicación permite cuantificar el nivel de riesgo de una determinada persona, con lo que, permitirá servir de referencia para que la unidad de riesgo tome las decisiones de manera más objetiva.

**Desarrollo**

Nuestro prototipo web ha sido desarrollado utilizando el framework Django por ser de código abierto, y además por su gran versatilidad y escalabilidad.

La aplicación consta de un API Rest el cual puede ser consumido desde cualquier aplicación cliente. Esta cuenta con los endpoint *get_prediction* y *get_resources*, que son los canales de comunicación entre el cliente y el sistema respectivamente.

- **get_prediction**: Realiza la predicción sobre los datos del cliente a evaluar.
- **get_resources**: Extrae desde la base de datos, las informaciones que son requeridas para autocompletar los campos del formulario.

En cada petición a (*get_prediction*), se realiza un proceso de validación y transformación de los datos enviados para convertir las variables categóricas en representaciones numéricas, de manera que se ajuste conforme a los requerimientos del modelo, luego se carga el modelo predictivo previamente entrenado, y se le aplican los datos procesados para realizar la predicción sobre los mismos.

**Dependencias**

Para la funcionalidad y desarrollo de nuestra herramienta web, instalamos en nuestro servidor de pruebas las siguientes dependencias:

- Python 3.7
- Docker ~1.8.0
- GitLab 12.3

las mismas deberán ser instaladas en el ambiente de producción en el momento de la implementación de la herramienta.

**Deployment**

La integración continua (CI) consiste en hacer integraciones automáticas de un proyecto cada cierto periodo de tiempo, y de esta manera, tener control de los posibles errores que puedan ocurrir en la actualizaciones del código fuente. Este mecanismo permite mejorar la calidad del software, ya que en él se crean las fases de compilación y prueba de los cambios realizados en el código fuente. Una vez las fases se ejecutan con éxito, se combinan los cambios en el repositorio.

La distribución continua (CD) consiste en la automatización de los procesos para el despliegue de la aplicación. Es el siguiente paso a la integración continua, y en este proceso se definen una serie de fases que deben ejecutarse en orden y de forma satisfactoria para llevar el despliegue a cabo.

GitLab es un servicio de control de versiones (repositorio) basado en git, desarrollado bajo la licencia de código abierto, el cual incluye integración continua y distribución continua (CI / CD).

Docker es un servicio para el despliegue automatizado de aplicaciones dentro de contenedores, de código abierto, el cual proporciona la capacidad de abstracción y virtualización de aplicaciones en diversos sistemas operativos.

Para llevar a cabo el despliegue de la aplicación web en el servidor de producción, nos apoyaremos de GitLab el cual será el repositorio central, y en él crearemos los ciclos de CI / CD sobre docker. Para la consecución, realizaremos los siguientes pasos:

- Crearemos 2 runners, que se utilizaran para realizar las acciones de CI y CD definidas en el archivo descriptor de GitLab “.gitlab-ci.yaml”. En este archivo yaml se definen los pasos y pipelines del proyecto que se deben ejecutar en cada commit al repositorio central y para el despliegue apropiado de la aplicación web.
- Definiremos el Dockerfile de la aplicación el cual contendrá las instrucciones necesarias para crear la imagen.
- Utilizaremos la versión LTS de la imagen de mongoDB desde el repositorio de docker.
- Crearemos un docker-compose.yml, el cual definirá los contenedores para las imágenes de la aplicación web y mongoDB respectivamente.
- Registramos los runners definidos para ejecutarse dentro de un contenedor de docker. Para ello es necesario que el runner se comunique con el servidor de docker, por lo tanto, como punto de montaje del archivo gitlab-runner se especificará el docker sock.

**Infraestructura**

La aplicación está alojada en una instancia de google cloud, y está a su vez alberga todos los recursos necesarios para el análisis y modelación de los datos. Durante el periodo de desarrollo y pruebas se ha utilizado la instancia con recursos limitados, sin embargo, dada la flexibilidad que

ofrece la plataforma para escalar vertical y horizontalmente, posteriormente los recursos serán ajustados conforme la demanda de uso de los servicios.

## 3.5 Análisis de actividades: modelo lógico - arquitectura técnica

### 3.5.1 Modelo lógico

1. Identificación de los datos.

La información se obtendrá de la base de datos interna del Banco Agrícola Dominicano. Utilizando Pentaho Data Integrator como herramienta de ETL para conectarnos a la base de datos del Banco.

Tipos de datos contenidos en el archivo de entrada.

| Campo | Tipo |
|---|---|
| id_sucursal | Entero |
| id_cuenta | Entero |
| id_cliente | Entero |
| tipo_de_cliente | Alfanúmerico |
| destino_credito | Alfanúmerico |
| estado_civil | Alfanúmerico |
| clase_de_persona | Alfanúmerico |
| Sexo | Alfanúmerico |
| nacionalidad | Alfanúmerico |
| estado_prestamo | Alfanúmerico |
| fecha_desembolso | Fecha |
| fecha_vencimiento | Fecha |
| Tasa | Decimal |
| monto_desembolsado | Decimal |
| Saldo | Decimal |
| atraso_de_0-30 | Decimal |
| atraso_de_31-90 | Decimal |
| atraso_mayor_a_90 | Decimal |
| atraso_resst_0-30 | Decimal |
| atraso_resst_31-90 | Decimal |
| atraso_resst_mayor_90 | Decimal |
| Interés | Decimal |
| interes_31-90 | Entero |
| interes_mayor_90 | Decimal |
| Suspenso | Decimal |
| interes_resst | Decimal |
| interes_resst_31-90 | Entero |
| interes_resst_mayor_90 | Decimal |
| suspenso_resst | Decimal |
| saldo_interes | Decimal |

| | |
|---|---|
| clasificacion | Alfanúmerico |
| tipo_de_credito | Alfanúmerico |
| Sector | Alfanúmerico |
| numero_garantia | Entero |
| monto_garantia | Decimal |
| fecha_tasacion | Fecha |

*Ilustración 13*

**2. Captura de datos.**

Se estudiará la posibilidad de crear alertas para mostrar al usuario si los datos no tienen la calidad suficiente, esto debe aplicarse tanto en el sistema como en lo que sería la capacitación de los analistas que van a intervenir en los proceso..

**3. Almacenamiento de los datos.**

Repositorio independiente en MongoDB donde se almacenarán los datos utilizados por el modelo y los resultados de las diferentes ejecuciones de entrenamientos.

**4. Transformación y validación de los datos.**

Dentro del proceso de ETL tenemos la lectura del archivo CSV, reemplazo de valores nulos, se excluyen en un archivo llamado "Registros inconsistentes" los préstamos que no cumplen. Con los registros válidos se cambian los valores de algunos campos, se calcula el plazo del préstamo y se cambia el formato para finalmente almacenar en la base de datos de MongoDB. Ver siguiente imagen:

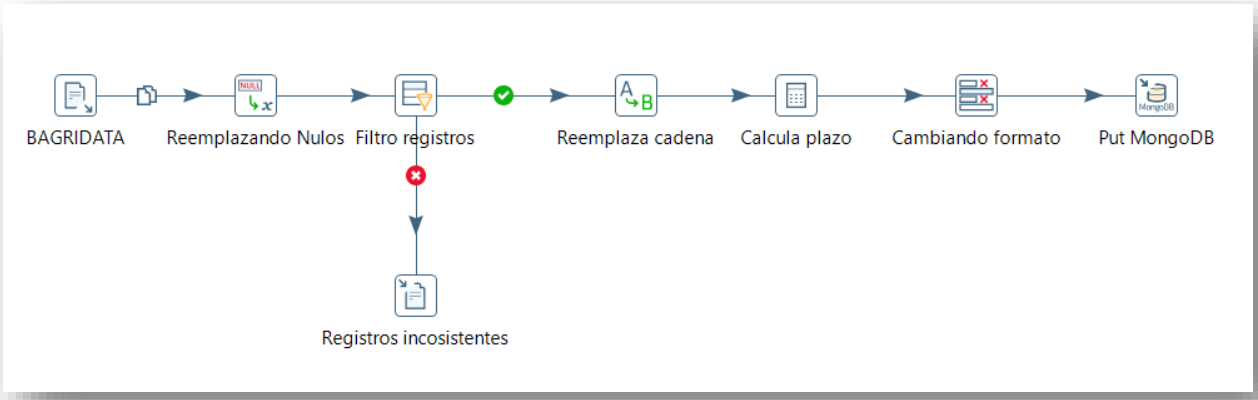



*Ilustración 14*

### 3.5.1 Descubrimiento y Modelado.

El proyecto utiliza algoritmos de clasificación tomando como entrenamiento los registros de préstamos anteriores, con esto generamos modelos predictivos que estimaran si el cliente pagará o no el préstamo solicitado, pero antes de llegar a ese punto es necesario trabajar con una analítica exploratoria (EDA) siguiendo los siguientes métodos:

- Iniciamos el proceso de análisis, estudiando los datos disponibles.
- Comprobamos que no existen nulos.
- Se trata de una data compuesta por 4 tipos de columnas, tenemos datos en coma flotante (float64), datos enteros (int64), datos fechas (datetime64) y los datos tipo object son strings, es decir variables categóricas, las cuales posteriormente tendremos que linealizar usando un one-hot-enconding.
- Comprobamos la distribución del dataset. Al no ser un dataset especialmente grande hemos decidido no balancear el dataset, ya que reduce drásticamente el número de muestras penalizando el modelo.

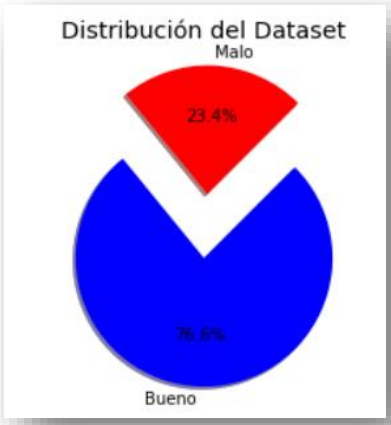



*Ilustración 15*

- Comprobar la correlación de las variables usando una matriz de correlación para determinar todas las variables que aporten una componente explicativa alta a la variable objetivo y que tengan una baja correlación entre ellas. Dos variables con una alta correlación significan que, con un grado de error bajo, una puede ser deducida a partir de la otra, y por tanto aportarán información redundante al modelo.
- Eliminamos las variables que no aportan valor al modelo. Quedando solo 3 tipos de columnas y 14 variables (4 variables coma flotantes, una variable entera y 9 categóricas).
- Convertimos las variables categóricas a numéricas.
- Normalizamos las variables numéricas para igualar el peso de las variables entre una escala de -1 y 1.

Para el proceso de ML realizamos lo siguiente:

- Separar las variables independientes "X" (Variables para entrenar) y la variable dependiente "y" (Variable a predecir).
- Separar los datos para entrenar el modelo con un 70% y los datos para probar el modelo con el restante 30%.
- Hemos usado 3 modelos de clasificación (RandomForestClassifier, AdaBoostClassifier, KNeighborsClassifier) para comprobar su desempeño con la data que actualmente tenemos.
- Se entrenan los distintos modelos pasando como primer parámetro las variables independientes y un segundo parámetro la variable dependiente de los datos de entrenamiento.
- Se realiza la predicción de la variable dependiente usando los datos de pruebas que no se han utilizado para entrenar.
- Realizamos una matriz de confusión utilizando los valores reales de variable dependiente y los valores predicho con los datos de pruebas. Dicha métrica nos muestra que el modelo que mejor trabajo bajo estas condiciones es AdaBoostClassifier debido a que tiene menor número falsos positivos (199) y menor número falsos negativos (277).

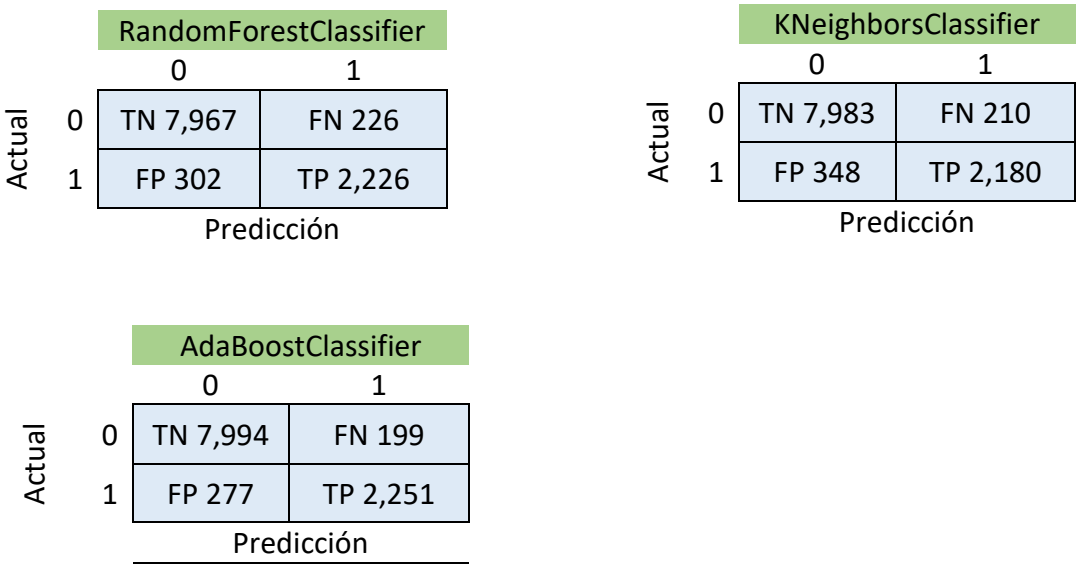



***Ilustración 16***

- AdaBoostClassifier tiene el mejor score en la precisión (91.88%) y el recall de (89.04%). El modelo RandomForestClassifier logra tener una precisión de 90.78% y un recall de 88.05% mientras KNeighborsClassifier tiene una precisión de 91.21% y un recall de 86.23%. La precisión es la proporción de préstamos que se predijeron como malos y realidad fueron malos (2,251 / (2,251 + 199)) mientras que la métrica de recall es la proporción de préstamos que realmente son malos y que el modelo predijo como malo (2,251 / (2,251 + 277)). El F1-score de AdaBoostClassifier es de 90.44% mejora que RandomForestClassifier con 89.40% y KNeighborsClassifier con 88.65%, en el caso del F1-score se toma en cuenta la precisión y el recall (2*91.88*89.04)/(91.88+89.04) obteniendo una puntuación única de ambas métricas.

- Las variables relevantes para el modelo de RandomForestClassifier se muestra en la siguiente gráfica:

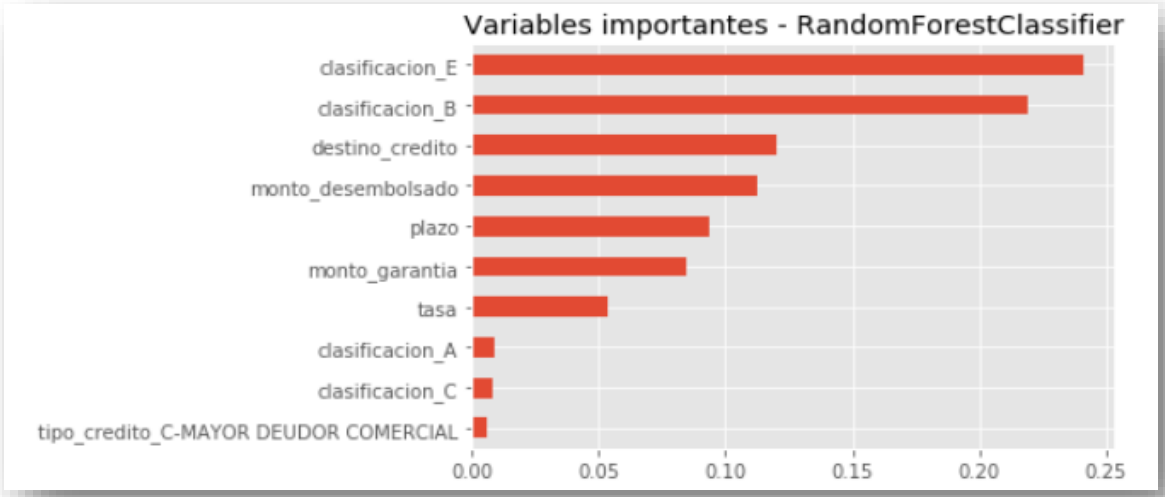



***Ilustración 17***

*Las variables relevantes para el modelo de AdaBoostClassifier se muestra en la siguiente gráfica:*

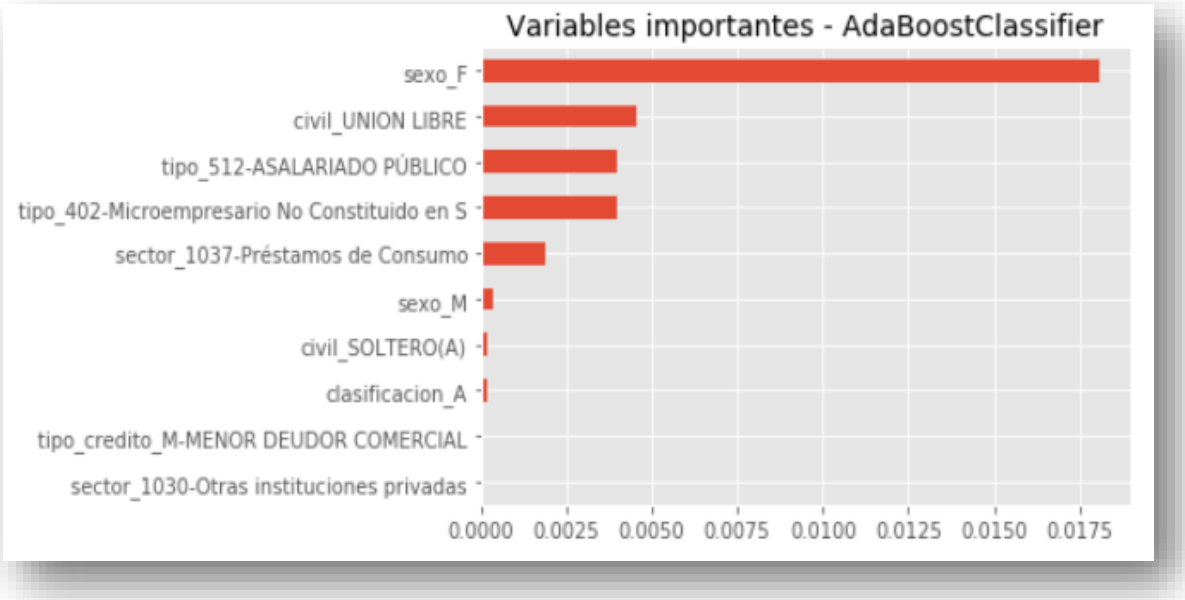



***Ilustración 18***

- El modelo KNeighborsClassifier no fue posible obtener las variables importantes.

- La curva ROC indica que el mejor el modelo AdaBoostClassifier dado que cubre un 93.31% a diferencia de los demás modelos RandomForestClassifier (92.65%) y KNeighborsClassifier (91.84%).

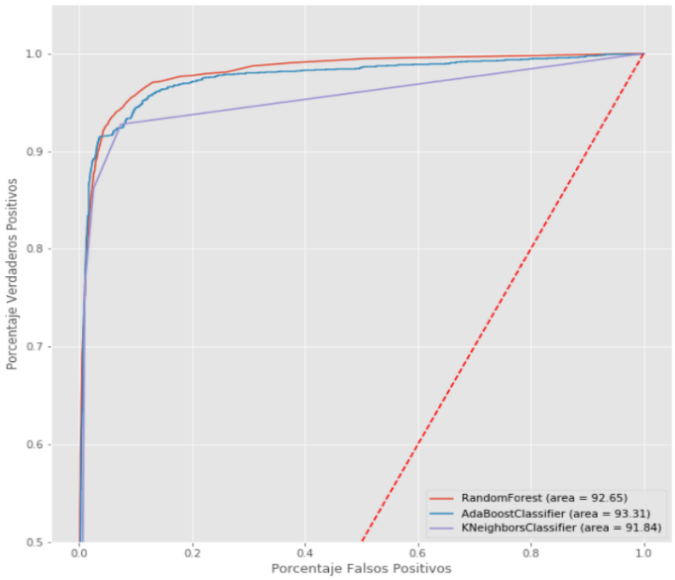



*Ilustración 19*

- Guardamos el modelo AdaBoostClassifier en un archivo pkl para que pueda ser usado por la aplicación para realizar la predicción. Este archivo será actualizado cada vez que el modelo vaya perdiendo el poder de predicción.
- Para evaluar si el cliente pagará o no el préstamo solicitado, el usuario del sistema enviará una determinada petición, que tras la ejecución del modelo predictivo podrá visualizar su condición de pagará o no.

### 3.5.2 Visualización.

Se realizará mediante una aplicación web y Power BI. Para realizar la conexión a MongoDB es necesario crear un ODBC con la conexión a la base de datos.

- Es necesario cambiar el tipo de datos a numérico del campo numero_garantia.
- Se agrega la columna garantía con la condición “Tiene” si el numero_garantia es mayor a cero de lo contrario es “No tiene”.
- Se agrega la columna Condición con la condición “Malo” si el estado_prestamo es 5 – VENCIDO o 11-REESTRUCTURADO de lo contrario es “Bueno”.

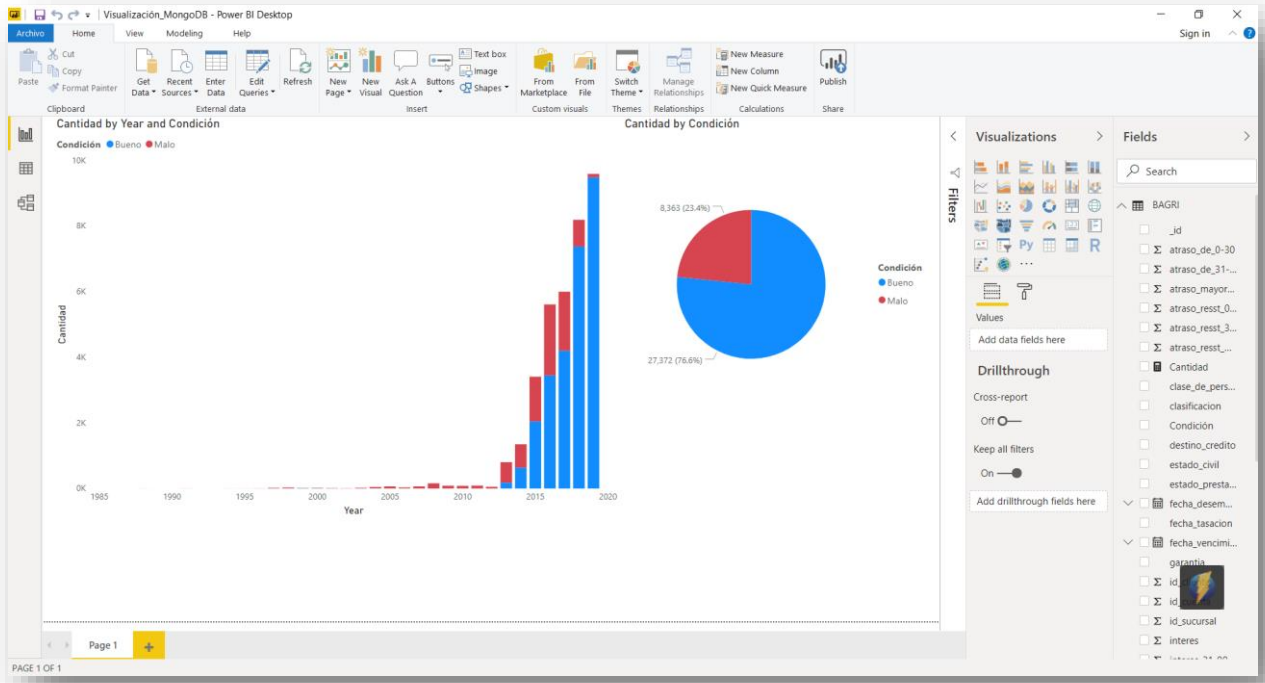



*Ilustración 20*

## 3.6 Solución tecnológica: Arquitectura técnica

Para efectuar el proyecto con el Banco Agrícola Dominicano, hemos utilizado Pentaho para ETL, Python como herramienta de análisis de datos y programación. La herramienta de BI será PowerBI. Para poder tener acceso a todas las funcionalidades de Power BI, el Banco tendrá que adquirir su licencia.

*Arquitectura técnica de la solución.*

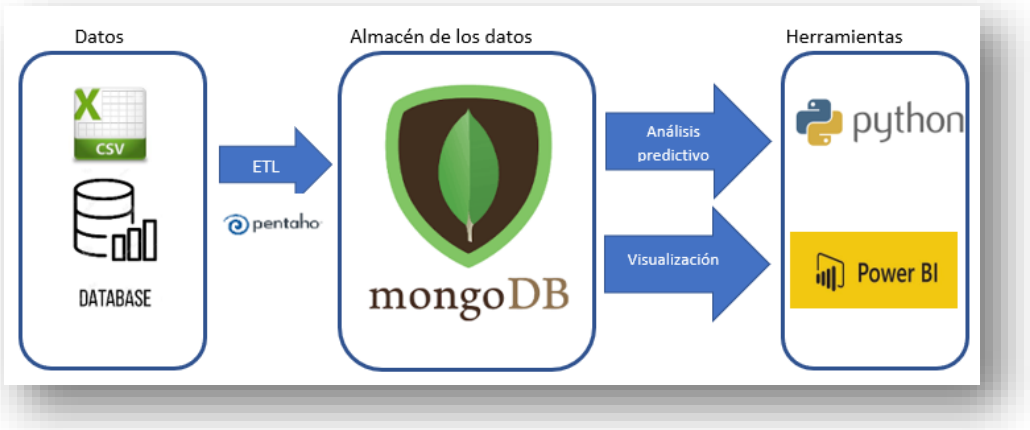



*Ilustración 21*

**ETL:** Pentaho será el encargado de conectarse a todos los orígenes de datos y cargarlos en un único destino. En esta fase se limpiarán todos los campos, verificando tanto sus valores como si el formato es el adecuado.

**Almacén de los datos:** Se construye el Data Lake sobre el cual trabajará los modelos predictivos y las visualizaciones.

**Python:** Para la parte analítica predictiva que conectaremos a MongoDB.

**PowerBI:** Utilizaremos esta herramienta para realizar los diferentes gráficos de análisis financiero.

## 3.7 Análisis de Recursos: Talento Humano y Recursos Físicos

### Talento Humano

La función de recursos humanos resulta estratégica para la consecución de los objetivos del Banco Agrícola a corto, mediano y largo plazo.

De una correcta gestión de los recursos humanos depende el control de los costes de personal, disponer del número de profesionales adecuados en cada etapa de las actividades, la calidad del talento disponible a corto y largo plazo, el clima y compromiso laboral, así como el desarrollo de los valores y principios de gestión que la estrategia de negocio requiere en cada etapa, es imprescindible disponer de la gestión de personas para recopilar información y gestionar el desempeño de los empleados. La parte objetiva va ganando terreno, y las personas no solo se validarán por su actitud, sino también por los hechos.

El departamento de recursos humanos se ha esforzado en realizar actividades que potencian el espíritu de equipo y cooperación y se incorpore el Big Data aplicado a la organización.

Es, por ello, que las tendencias actuales de la práctica de recursos humanos en el Banco Agrícola, son radicalmente diferentes a las conocidas no hace más de cinco años, están en continua redefinición, afectando en consecuencia la forma de trabajo desarrollada en tiempos anteriores.
Se han estado afrontando los retos de la revolución tecnológica en la gestión de la fuerza de trabajo; como, por ejemplo, integrar al equipo TI en Inteligencia de Negocios, Big Data, Machine Learning, entre otras tecnologías en pos del avance de la institución con miras de estar a la vanguardia acorde a los nuevos tiempos en el uso de esta tecnología.

Acciones propuestas en la Transformación Digital de la institución con el personal:

| Vector de Transformación Digital Personas | Acciones "Exploit" | Acciones "Explore" |
|---|---|---|
| **Cultura** | Desarrollo profesional y técnico de los individuos y equipos de trabajo en tecnología de Big Data y Analítica de datos. | Implementar herramientas que puedan medir la gestión y el análisis de los datos enfocado en medir el credit scoring de los clientes. |
| **Talento** | Reclutamiento de Perfiles con Talento en Análisis de los Datos.<br><br>Incentivación del Cross-Training entre colaboradores.<br><br>Promoción y retención del talento interno. | Identificación de perfiles "especiales" en los equipos tradicionales. |
| **Liderazgo** | Capacitación para crear líderes en nuevas tecnologías de Big Data.<br><br>Flexibilidad en cambios.<br><br>Capacidad de Innovación. | - Enfoque de transformación hacia equipos y no personas.<br><br>Capacitación de equipos para lograr las siguientes cualidades:<br>- Establecer y enfocarse en el logro de resultados.<br>- Coordinar tareas, roles y funciones.<br>- Gestionar la incertidumbre.<br>- Migración de los sistemas de organización jerárquicos a redes de equipos que pueden adaptarse a un entorno en constante cambio. |

***Ilustración 22***

### 3.7.1 Estructura Organizativa Actual, Departamento TI.

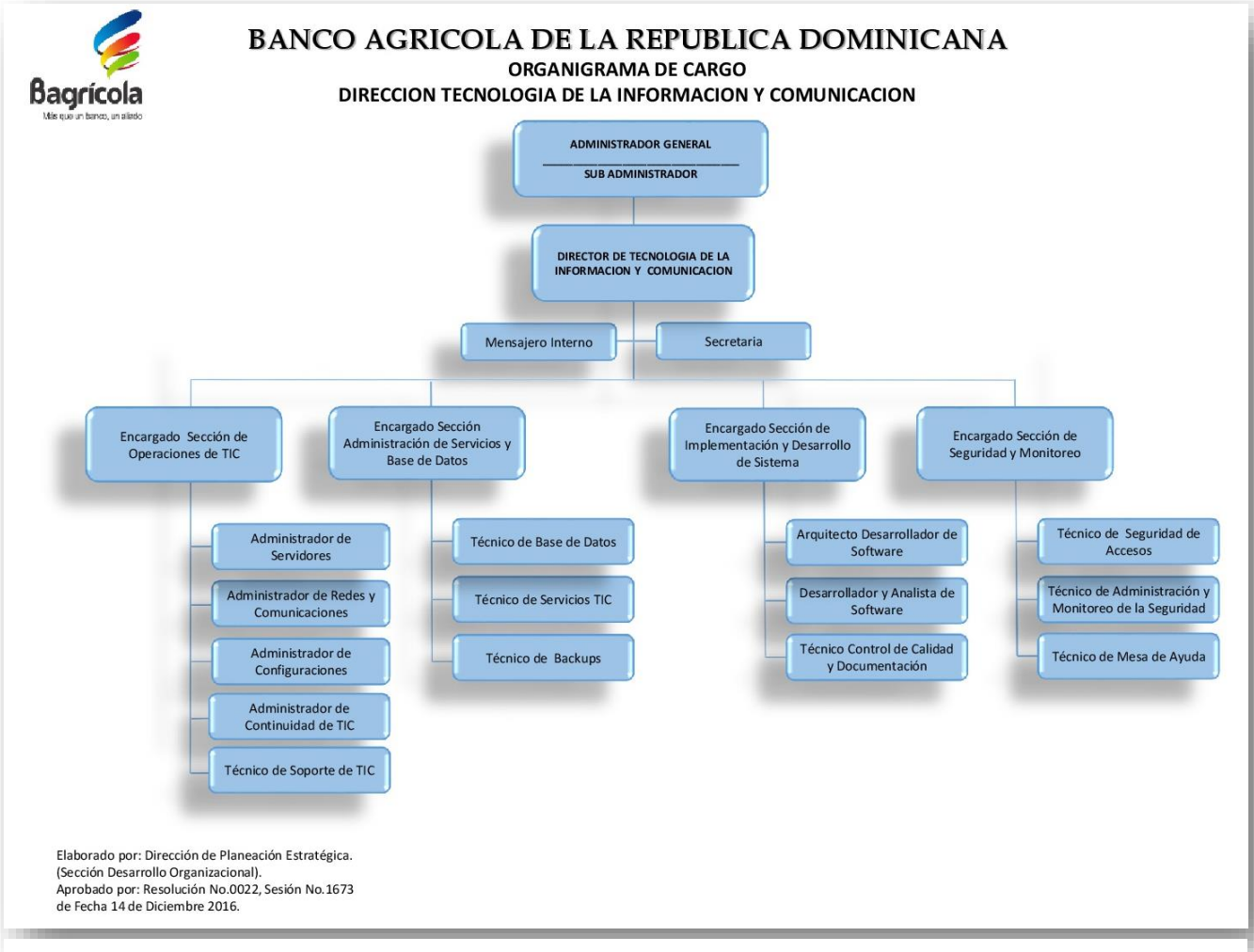



*Ilustración 23*

### 3.7.2 Personal Sugerido para Integrarlo a la Estructura Departamento TI.

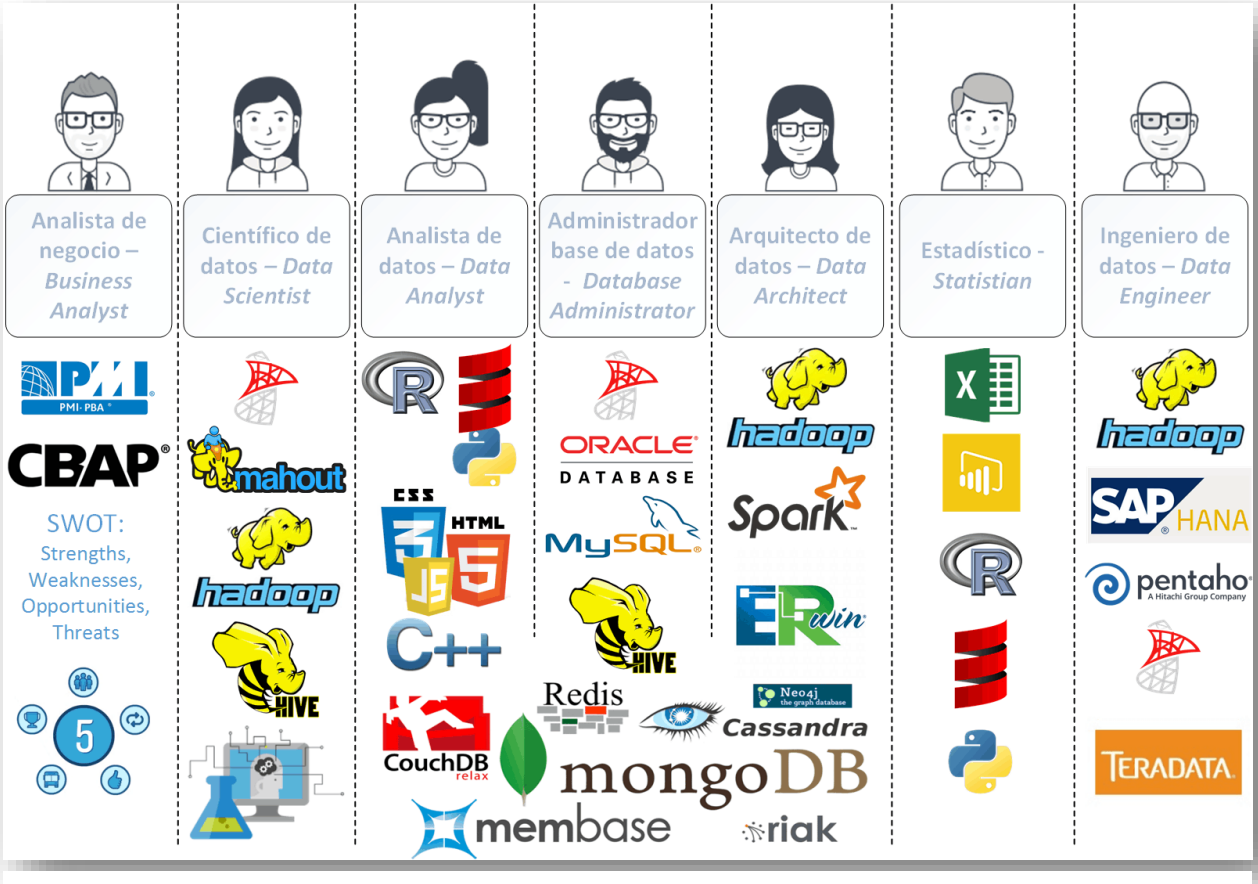



*Ilustración 24*

**Data Analyst (DA)**

Este es un rol bastante generalista, abarcando una amplia gama de funciones que incluyen la minería, obtención y/o recuperación de datos, así como su procesado, estudio avanzado y visualización.

**Data Scientist**

Es la "evolución del Data Analyst". En muchos casos los consideran el mismo perfil con diferente enfoque. Al igual que el DA, requiere saber de matemáticas, estadística y Machine Learning, de lenguajes de programación como R o Python, de uso de notebooks y de ecosistemas Big Data.

**Data Engineer**

Enfocándonos en el almacenamiento y procesado de datos, nos encontramos con el rol de Data Engineer. Debe conocer cómo se modelan los datos, así como tener un amplio conocimiento de las

BBDD SQL, ya que en el mundo del Big Data éstas no se excluyen y siguen en muchos casos siendo el origen de los datos.

### 3.7.3 Infraestructura Física Actual.

Debajo listaremos los equipos físicos actuales que dispone la institución como parte de sus activos tecnológicos para el desarrollo de los objetivos en la organización.

| 751 PC de Escritorio | | | | | |
|---|---|---|---|---|---|
| CANT | Hardware | | | Software | Seguridad |
| | Tipo | Marca | Modelo | | |
| 375 | PC Desktop | DELL | Optiplex GX 9010 | Windows 7 Profesional | System Center Endpoint Protection |
| 280 | PC Desktop | DELL | Optiplex GX 790 | | |
| 46 | PC Desktop | DELL | Optiplex GX 755 | Windows 7 Profesional | |
| 42 | PC Desktop | DELL | Optiplex GX 620 | | |
| 8 | Laptop | DELL | Inspiron | Windows 7 Profesional | |
| | | | | | |
| **12 Servidores Físicos** | | | | | |
| 1 | Servidor | HP | ProLiant ML33G6 | Windows Server 2008 R2 | System Center Endpoint Protection |
| 1 | Servidor | HP | ProLiant DL120 G6 | | |
| 3 | Servidor | DELL | Power Edge 650 | | |
| 2 | Servidor | DELL | Power Edge R910 | | |
| 1 | Servidor | DELL | Power Edge R420 | | |
| 7 | Servidor | DELL | Power Edge R720 | Linux Suset | |
| 2 | Servidor | DELL | Power Edge 8450 | | |
| 2 | Unidad de Cinta | DELL | Power Vault | | |
| **Total de Servidores 12** | | | | | |

| 14 Servidores Virtuales | | | | | |
|---|---|---|---|---|---|
| 14 | Servidores Virtualizado | Hyper-V | Distribuidos en 2 Servidores Físicos Power Edge R910 | Windows Server 2008 R2 | System Center Endpoint Protection |

| 12 Servidores LAB | | | | | |
|---|---|---|---|---|---|
| 12 | Servidores Tipo PC | Dell | Optiplex GX 790 | Windows Server 2008 R2 | System Center Endpoint Protection |

| Redes y Telecomunicaciones<br>57 Router – 21 Switch - 290 Telefonos IP – 1 Central Cisco | | | | | |
|---|---|---|---|---|---|
| 1 | Router | Avaya | AG2330 | AVAYA | NA |
| 1 | Router | Cisco | 3800 | Cisco IOS 15.0 | |
| 1 | Router | Cisco | 2800 | | |
| 1 | Router | Cisco | Serie 800 | | |
| 1 | Switch | Cisco | Catalyst 3750 | | |
| 18 | Switch | Cisco | Catalyst 2960 24-p | | |
| 1 | Switch | Cisco | Catalyst 2960 48-p | | |
| 1 | Switch | Cisco | Catalyst 3760 | | |
| 1 | Firewall | Cisco | PIX-ASA | | Firewall |
| | | | | IOS 8.5.6 | |
| 1 | Central Telefónica IP | Cisco | Cisco | | |
| 290 | Teléfonos IP 6941- 6921 - 3509 | Cisco | Cisco | | |
| 21 | Router Cisco 887 | Cisco | 887-VA | Cisco IOS 14.2 | Conexión Oficina de Servicios VPN |
| 32 | Router Cisco 887 | Cisco | 887-VA | Cisco IOS 14.2 | Conexión Sucursal ATM |
| | | | | | |

| Seguridad | | | | | |
|---|---|---|---|---|---|
| 1 | Aplaappliance | Symantec | | | Firewall web Gateway |
| 1 | Servidor Antivirus | Dell | Power Edge R720 | System Center Endpoint Protection | Antivirus |
| 1 | NetBox | NetBox | NetBox | Control de Temperatura | Temperatura |

*Ilustración 25*

### 3.7.4 Equipos Físicos Sugeridos a ser Adicionados al Proyecto.

Debajo listaremos los equipos físicos, proponemos se incorporen a los activos tecnológicos actuales acorde al proyecto de "Bacri Credit Scoring" para los trabajos de Machine Learning.

| EQUIPO A AGREGARSE | | | | | |
|---|---|---|---|---|---|
| 3 | PC Desktop | DELL OPTIPLEX GX 620 | 16 GB RAM | 1 TB en disco | System Center Endpoint Protection |
| 1 | Software | Licencia Office 365 | Power BI | Microsoft Office | N/A |
| 1 | Servicio en la nube | Google Cloud Engine | GCE Console | Google | GCE |

*Ilustración 26*

## 3.8 Gestión del Tiempo (cronograma)

Para definir las etapas y el control de tareas en los tiempos de entrega, se realizó el siguiente cronograma de trabajo detallado para el análisis, desarrollo e implementación de la solución de Big Data en el Banco Agrícola Dominicano.

| Tareas del Proyecto | Duration | Start | Finish |
|---|---|---|---|
| Preparación del proyecto (tercer trimestre 2019): | 36 days | Mon 29/07/19 | Sun 15/09/19 |
| Formación del equipo de trabajo. | 14 days | | |
| Definir el alcance del proyecto (anteproyecto). | 14 days | | |
| Definir la planificación y la gestión del proyecto. | 30 days | | |
| Identificación de áreas involucradas. | 20 days | | |
| Identificación de Infraestructura. | 15 days | | |
| Identificar requerimientos de información. | 12 days | | |
| Análisis funcional y arquitectura tecnológica (tercer y cuarto trimestre 2019) | 10 days | Fri 16/08/19 | Thu 29/08/19 |
| Revisar los requerimientos de negocio (tiempos de extracción de datos, tipos de datos, etc.) | 7 days | | |
| Reunión de equipo. | 5 days | | |
| Definir la arquitectura tecnológica (hardware propuesto por el Banco Agrícola). | 15 days | | |
| Definir las recomendaciones de configuración que se ajuste al Banco Agrícola. | 7 days | | |
| Instalación del software (Python). | 1 day | | |
| Usabilidad y prototipo (Tercer trimestre 2019) | 11 days | Tue 01/10/19 | Tue 15/10/19 |
| Desarrollar modelo de datos. | 14 days | | |
| Analizar las fuentes de datos. | 7 days | | |
| Limpieza de datos. | 7 days | | |
| Diseño de ETL. | 5 days | | |
| Modelo de análisis predictivo del conjunto de datos. | 5 days | | |
| Desarrollo y programación de la solución (tercer trimestre 2019) | 14 days | Mon 14/10/19 | Thu 31/10/19 |
| Revisar el alcance y la planificación. | 3 days | | |
| Diseñar y desarrollar la integración de datos. | 7 days | | |
| Pruebas (cuarto trimestre 2019) | 7 days | Mon 21/10/19 | Tue 29/10/19 |
| Puesta en marcha plan de pruebas y aseguramiento de la calidad. | | | |

*Ilustración 27*

### 3.8.1 Diagrama de Gantt:

El siguiente cronograma representa las actividades generales de diseño e implementación del proyecto.

| Actividades del Proyecto | Recursos | Duration | Start | Finish |
|---|---|---|---|---|
| **ANÁLISIS** | | | | |
| Estudio de requerimientos | Cliente, Consultor, Analista | 14 days | Mon 29/07/19 | Thu 15/08/19 |
| Análisis fuentes de datos | Analista | 21 days | Mon 02/09/19 | Mon 30/09/19 |
| Análisis modelos predictivos | Analista, Desarrollador | 1 day | Tue 01/10/19 | Tue 01/10/19 |
| Estudio alternativas tecnológicas | Analista, Desarrollador | 31 days | Mon 19/08/19 | Mon 30/09/19 |
| **DISEÑO** | | | | |
| Selección modelo análisis | Analista | 32 days | Fri 16/08/19 | Mon 30/09/19 |
| Definición arquitectura técnica | Analista, Desarrollador | 3 days | Thu 05/09/19 | Mon 09/09/19 |
| **IMPLEMENTACIÓN** | | | | |
| Desarrollo modelos predictivos | Analista, Desarrollador | 11 days | Tue 01/10/19 | Tue 15/10/19 |
| Integración solución técnica | Analista, Desarrollador | 4 days | Fri 04/10/19 | Wed 09/10/19 |
| **VALIDACIÓN Y DESPLIEGUE** | | | | |
| Testeo modelos predictivos | Analista, Desarrollador | 4 days | Tue 01/10/19 | Fri 04/10/19 |
| Pruebas del sistema | Cliente, Analista, Desarrollador | 5 days | Tue 01/10/19 | Mon 07/10/19 |
| Puesta en preproducción | Cliente, Implementador | 1 day | Mon 14/10/19 | Mon 14/10/19 |
| **SOPORTE Y MANTENIMIENTO** | | | | |
| Formación de usuarios | Cliente, Implementador | 14 days | Tue 01/10/19 | Fri 18/10/19 |
| Soporte a usuarios | Soporte, Analista, Desarrollador | 9 days | Tue 01/10/19 | Fri 11/10/19 |
| Seguimiento | Consultor, Cliente | 15 days | Fri 08/11/19 | Thu 28/11/19 |

*Ilustración 28*

**Alcance.**

Este proyecto está orientado a la construcción de un prototipo para el análisis Big Data en la Institución del Banco Agrícola Dominicano, que comprende el siguiente alcance:

• Desarrollar un ejemplo básico de programación para analítica de datos con Python, donde se analiza un dataset con cierta cantidad de información, dado a que es un prototipo y no se cuenta con los requerimientos suficientes para hacerlo con una gran cantidad de datos.

• En la implementación se incluye pasar el ejemplo de programación junto con los datos, además de realizar el análisis de datos para finalmente mostrar un resultado.

Las fases definidas para el proyecto son las siguientes:

**1. Fase de levantamiento de información e investigación**: Durante esta fase se realizará el levantamiento de información de las características que debe tener el modelo predictivo, según la necesidad planteada, además, se recopilará el conjunto de conocimientos necesarios para llevar a cabo este proyecto.

**2. Fase de diseño:** Durante esta fase se realizará el diseño del prototipo para el análisis de Big Data, que permitirá procesar documentos de texto según el programa ejecutado, aplicando los conocimientos adquiridos en la fase de Levantamiento de información e investigación.

**3. Fase de instalación y montaje:** Durante esta fase se implementará el prototipo según él diseño realizado durante la fase de diseño.

**4. Fase de desarrollo:** Durante esta fase se desarrollará un ejemplo de programación en Python que permita analizar un dataset, de acuerdo con lo especificado en la fase de diseño.

**5. Fase de implementación:** Durante esta fase se implementará el prototipo elaborado en la fase de instalación para ejecutar el ejemplo programado, ejecutado en la fase de desarrollo y se mostrará el resultado del análisis.

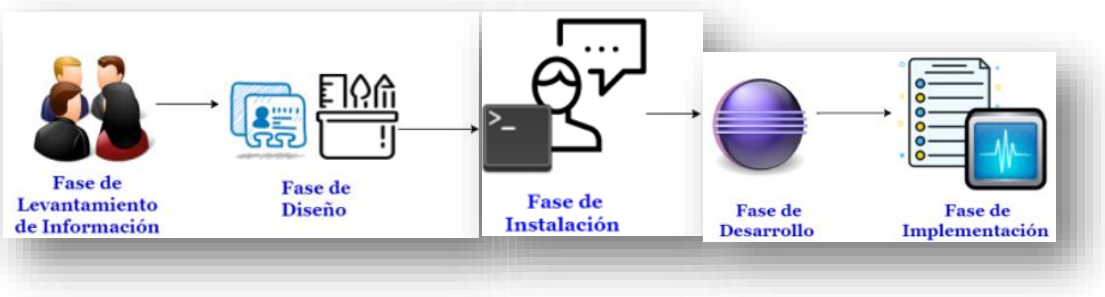



*Ilustración 29*

En estas fases resaltamos las actividades que tienen que ver con las mejores propuestas en la institución y se definan los requisitos y supuestos, cuya meta sea alcanzar los objetivos estratégicos del proyecto con éxito.

## 4. Proyecto de optimización

Bagri Credit Scoring es una solución BD/BI desarrollada para un cliente especifico el cual es el Banco Agrícola Dominicano. Esta entidad financiera compartió información para ser utilizada en la

implementación del proyecto, con el fin de obtener optimización en el proceso de otorgamientos de créditos, por ser la depuración de cliente el primer paso para ceder un crédito y por lo mismo es el más decisivo en lo que será en un futuro la cartera de préstamos.

## ❖ Beneficios Tangibles:

### ➢ Generación de ingresos:

- Optimizar la atención al cliente.
- Reducir el lapso de tiempo invertido en depurar al cliente.
- Optimizar la captura de datos de importancia.
- Reducir la tasa de morosidad al incrementar la probabilidad de retorno de la inversión concedida.

- Aumentar el ingreso por crecimiento de la cartera de préstamos vigentes y disminuyendo los casos vencido y de cuotas vencidas.
- Aumentar los resultados anuales asegurando la inversión en proyectos agrícolas viables.
- Aumentar la captación de clientes “buenos” y rechazar los clientes “malos”.
- Evitar pérdidas generadas por créditos impagos.
- Disminuir el personal involucrado en el proceso de depuración del cliente.
- Prever comportamientos decrecientes dentro de la cartera de préstamos, causados por incumplimientos.

El proyecto contempla una disminución en la morosidad en un 5% para el primer año de uso luego de implementado y un aumento de 1% anual mediante el uso continuo.

| Optimizacion Proyecciones Cobro X Disminucion de Morosidad | | | | |
|---|---|---|---|---|
| | Año 1 | Año 2 | Año 3 | Año 4 |
| Actual | 14,839,000.20 | 15,581,000.20 | 16,360,000.20 | 17,178,000.20 |
| saldo recuperado | 741,950.01 | 934,860.01 | 1,145,200.01 | 1,374,240.02 |
| % gradual | 5% | 6% | 7% | 8% |

*Ilustración 30 de elaboración propia*

En el ***Plan Estratégico 2017-2020*** de nuestro cliente banco agrícola dominicano en su apartado **Programa de Préstamos 2017-2020** (pág. 5), se verá beneficiado con la utilización de la solución BCS, ya que este asegura que el programa de préstamo pautado se verá incrementado en el mismo porcentaje en que reduzca la morosidad, ya al mismo tiempo disminuyen las provisiones y se convierten en capital a utilizar en los planes del programa de préstamos.

| Optimizacion Proyecciones de Prestamos | | | | |
|---|---|---|---|---|
| | Año 1 | Año 2 | Año 3 | Año 4 |
| Actual | 17,864,000.60 | 18,757,000.80 | 19,695,000.70 | 20,680,000.50 |
| saldo incrementado | 893,200.03 | 1,125,420.05 | 1,378,650.05 | 1,654,400.04 |
| % gradual | 5% | 6% | 7% | 8% |

*lustración 31 de elaboración propia*

Los productores a ser beneficiados con créditos, contemplados en el apartado **Beneficiarios Directos e Indirectos *del Plan Estratégico 2017-2020*** (pag.54), de nuestro cliente banco agrícola dominicano, se verán consolidados en el porcentaje que ofrece nuestra solución BI/BD como "buenos", asegurando la no otorgación de préstamos a productores con altas probabilidades de impago. *(los datos del cuadro están por cantidad de clientes)*.

| PRODUCTORES DIRECTOS A SER BENEFICIADOS CON LA EJECUCION DEL PLAN ESTRATÉGICO | | | | |
|---|---|---|---|---|
| Plan en la Actualidad | | | | |
| | 2017 | 2018 | 2019 | 2020 |
| Productores Beneficiados | 30,968 | 32,516 | 34,142 | 35,849 |
| Plan utilizando BCS | | | | 91.80% |
| Productores Beneficiados | 28,429 | 29,850 | 31,342 | |

*Ilustración 30 de elaboración propia*

- **Reducción de costes:**

- Disminución de herramientas a utilizar, ya que se centralizará en un único sistema.
- Reducir el personal que interviene en la depuración del cliente.
- Aumentar la captación de mejores y confiables datos, evitando retrasos.
- Disminuir gestión del departamento de cobros.
- Disminuir gestión en créditos con alta probabilidad de impago.
- Controlar pérdidas determinando clientes con menor capacidad de cumplimiento.
- Reducir los créditos sin posibilidad de cobro.
- Disminuir las provisiones.
- Reducir el sobreendeudamiento de la clientela.
- Reducir el tiempo de espera para otorgamientos de créditos.

### ❖ Beneficios Intangibles:

Contar con mejor información recogida, no solo para la evaluación del cliente, sino también para análisis más amplios sobre el estado futuro de un préstamo y su comportamiento a través del tiempo.

- Mejorar el manejo de la información recogida mediante el uso de la herramienta BI/BD
- Obtener información más precisa y confiable.
- Minimizar los errores humanos, por el sentido imparcial de la herramienta.
- Mayor unificación de los datos recogidos de diversas fuentes.

### ❖ Beneficios Estratégicos:

- Mayor captación de clientes buenos.
- Mejora en el plan estratégico de la institución.
- Aumentar la toma de mejores decisiones.
- Mejor gestión de los recursos.

**ANÁLISIS RENTABILIDAD DEL PROYECTO**

| | AÑO 0 | AÑO 1 | AÑO 2 | AÑO 3 |
|---|---|---|---|---|
| **INVERSIÓN** | | | | |
| Inversion Inicial Arraque del Proyecto | 139,518.36 DOP | | | |
| TOTAL INVERSIONES | 139,518.36 DOP | - DOP | - DOP | - DOP |
| **INGRESOS/BENEFICIOS** | | | | |
| Optimizacion Proyecciones Cobros | - DOP | 741,950.01 DOP | 934,860.01 DOP | 1,145,200.01 DOP |
| Optimizacion Proyecciones de Prestamos | - DOP | 893,200.03 DOP | 1,125,420.05 DOP | 1,378,650.05 DOP |
| TOTAL INGRESOS/BENEFICIOS | - DOP | 1,635,150.04 DOP | 2,060,280.06 DOP | 2,523,850.06 DOP |
| **GASTOS** | | | | |
| | AÑO 0 | AÑO 1 | AÑO 2 | AÑO 3 |
| Desarrollo app | 67,518.36 DOP | 78,408.72 DOP | 78,408.72 DOP | 78,408.72 DOP |
| powerBI | 72,000.00 DOP | 72,000.00 DOP | 72,000.00 DOP | 72,000.00 DOP |
| MongoDB Atlas | - DOP | - DOP | 18,000.00 DOP | 18,000.00 DOP |
| Honorarios de Implementacion | 2,669,000.00 DOP | - DOP | - DOP | - DOP |
| Soporte Tecnico | - DOP | - DOP | 180,000.00 DOP | 180,000.00 DOP |
| TOTAL GASTOS | 2,808,518.36 DOP | 150,408.72 DOP | 348,408.72 DOP | 348,408.72 DOP |
| FLUJO DE CAJA OPERATIVO | - 2,948,036.72 DOP | 1,484,741.32 DOP | 1,711,871.34 DOP | 2,175,441.34 DOP |
| VALOR ACTUAL | - 2,948,036.72 DOP | 1,349,764.84 DOP | 1,414,769.70 DOP | 1,634,441.28 DOP |
| ACUMULADO | - 2,948,036.72 DOP | - 1,598,271.88 DOP | - 183,502.18 DOP | 1,450,939.10 DOP |
| VAN: | 6,963,924.80 | | | |
| TIR: | 34% | | | |

*Ilustración 31*

El flujo de caja se visualiza en aumento incremental lo cual es beneficioso para la rentabilidad del proyecto y obtenemos los siguientes valores:

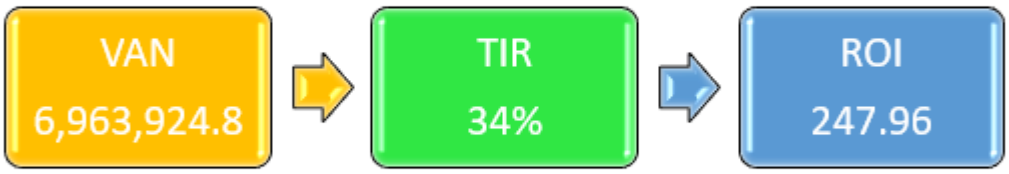



*Ilustración 32*

### 4.1 Proyecto Creación

Bagri Credit Scoring es un solución de BI/BD que ha sido desarrollada específicamente y a la medida como método de optimización e inserción en las nuevas tecnologías, para el banco agrícola dominicano, único patrocinador de la implementación de la solución.

#### 4.1.1 Plan de Inversión:

El proyecto de Bagri Credit Scoring ha sido desarrollado utilizando recursos propios de los analistas y los desarrolladores disponibles y libre de pago, por lo menos para la fase del prototipo que se mostrara al cliente final Banco Agrícola Dominicano, para la app desarrollada si se tuvo que disponer de recursos para la utilización de Compute Engine.

| Herramientas | Inversión Propia |
|---|---|
| Pentaho | Free |
| PowerBI | Free por estar incluida en office 360 de Microsoft |
| MongoDB | free |
| Python | free |
| GCE de Desarrollo | 24.27 us |

***Ilustración 33 de elaboración propia***

### 4.1.2 Plan de Financiación:

La naturaleza de la empresa tiene un propósito específico y concreto que es el desarrollo de una herramienta BI/BD para la optimización de la depuración del cliente al momento de solicitar un préstamo. de forma tal, que el ingreso por concepto de honorarios por el servicio brindado seria el ingreso de forma inmediata.

| Honorarios | | | | |
|---|---|---|---|---|
| | Al día | Mensual | Días de Implementación (# días) | Costo de Implementación (por día) |
| **Analista** | 3,000.00 | 90,000.00 | 157 | 471,000.00 |
| **Analista** | 3,000.00 | 90,000.00 | 157 | 471,000.00 |
| **Analista** | 3,000.00 | 90,000.00 | 157 | 471,000.00 |
| **Desarrollador** | 4,000.00 | 120,000.00 | 157 | 628,000.00 |
| **Desarrollador** | 4,000.00 | 120,000.00 | 157 | 628,000.00 |
| **Total =** | 17,000.00 | 510,000.00 | 157 | 2,669,000.00 |

*Ilustración 34 de elaboración propia*

Luego de terminada la implementación del proyecto y puesta en funcionamiento en producción, el cliente contara con un año de soporte gratis, luego de este año se deberá pagar las sumas estipuladas en el siguiente cuadro dependiendo de la elección del cliente y como este se sienta más cómodo contratando:

| | Hora | Dia | Mes |
|---|---|---|---|
| **Soporte** | | | |
| **Analista** | 100 | 2,000 | 60,000.00 |
| **Desarrollador** | 200 | 4,000 | 120,000.00 |
| | | **TOTAL =** | 180,000.00 |

*Ilustración 35 de elaboración propia*

# BIBLIOGRAFÍA


- **Informaciones de planeación estratégica del banco agrícola:** https://www.bagricola.gob.do/transparencia/index.php/plan-estrategico-de-la-institucion/plan-estrategico-2017-2020.
- **Definición morosidad:** https://es.wikipedia.org/wiki/Moroso.
- **Imágenes: Google**
- **Tablas: creaciones propias**

# ANEXOS

```
# Importando las librerías
from pymongo import MongoClient
import pandas as pd
import numpy as np
from datetime import datetime
import seaborn as sns
import matplotlib
from matplotlib import pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import confusion_matrix, classification_report, precision_score, recall_score, accuracy_score,f1_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import AdaBoostClassifier, RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
import joblib

scaler = MinMaxScaler()
#%matplotlib inline
matplotlib.style.use('ggplot')
random=42
```

```
mongoClient = MongoClient('mongodb://root:root@localhost:27017/')
```

```
db = mongoClient.BAGRI
```

```
collection = db.BAGRI
```

```
cursor = collection.find()
```

```
data = pd.DataFrame(list(cursor))
```

## EDA

**Iniciamos el proceso de análisis, estudiando los datos disponibles**

```
data.head()
```

| | _id | id_sucursal | id_cuenta | id_cliente | tipo_de_cliente | destino_credito | estado_civil | clase_de_persona | sexo | nacionalidad | ... | inter |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 5db40b03718cc02f6009e361 | 1 | 1126807 | 566183 | 511-ASALARIADO PRIVADO | CAÑA DE AZUCAR | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |
| 1 | 5db40b03718cc02f6009e362 | 1 | 1198203 | 467876 | 511-ASALARIADO PRIVADO | SALON DE BELLEZA | SOLTERO(A) | FISICA | F | DOMINICANA | ... | |
| 2 | 5db40b03718cc02f6009e363 | 1 | 1216760 | 589226 | 511-ASALARIADO PRIVADO | ADQUISICION MOTOCICLETA EMPLEADO | SOLTERO(A) | FISICA | F | DOMINICANA | ... | |
| 3 | 5db40b03718cc02f6009e364 | 1 | 1156500 | 564423 | 511-ASALARIADO PRIVADO | ADQUISICION MOTOCICLETA EMPLEADO | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |
| 4 | 5db40b03718cc02f6009e365 | 1 | 1200845 | 587535 | 511-ASALARIADO PRIVADO | ADQUISICION MOTOCICLETA EMPLEADO | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |

5 rows × 38 columns

```
data.tail()
```

| | _id | id_sucursal | id_cuenta | id_cliente | tipo_de_cliente | destino_credito | estado_civil | clase_de_persona | sexo | nacionalidad | ... | i |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 35730 | 5db40b11718cc02f600a6ef3 | 33 | 1013755 | 281731 | 511-ASALARIADO PRIVADO | AJO (FOMENTO) | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |
| 35731 | 5db40b11718cc02f600a6ef4 | 33 | 1084354 | 129808 | 511-ASALARIADO PRIVADO | REPOLLO (FOMENTO) | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |
| 35732 | 5db40b11718cc02f600a6ef5 | 33 | 918793 | 313971 | 511-ASALARIADO PRIVADO | PAPA (FOMENTO) | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |
| 35733 | 5db40b11718cc02f600a6ef6 | 33 | 1288286 | 282391 | 511-ASALARIADO PRIVADO | ZANAHORIA (FOMENTO) | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |
| 35734 | 5db40b11718cc02f600a6ef7 | 33 | 1015745 | 282851 | 511-ASALARIADO PRIVADO | TRACTORES (AGRICOLA) | SOLTERO(A) | FISICA | M | DOMINICANA | ... | |

5 rows × 38 columns

```
data.isnull().values.any()
```

```
False
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35735 entries, 0 to 35734
Data columns (total 38 columns):
_id                     35735 non-null object
id_sucursal             35735 non-null int64
id_cuenta               35735 non-null int64
id_cliente              35735 non-null int64
tipo_de_cliente         35735 non-null object
destino_credito         35735 non-null object
estado_civil            35735 non-null object
clase_de_persona        35735 non-null object
sexo                    35735 non-null object
nacionalidad            35735 non-null object
estado_prestamo         35735 non-null object
fecha_desembolso        35735 non-null datetime64[ns]
fecha_vencimiento       35735 non-null datetime64[ns]
tasa                    35735 non-null float64
monto_desembolsado      35735 non-null float64
saldo                   35735 non-null float64
atraso_de_0-30          35735 non-null float64
atraso_de_31-90         35735 non-null float64
atraso_mayor_a_90       35735 non-null float64
atraso_resst_0-30       35735 non-null float64
atraso_resst_31-90      35735 non-null float64
atraso_resst_mayor_90   35735 non-null float64
interes                 35735 non-null float64
interes_31-90           35735 non-null int64
interes_mayor_90        35735 non-null float64
suspenso                35735 non-null float64
interes_resst           35735 non-null float64
interes_resst_31-90     35735 non-null int64
interes_resst_mayor_90  35735 non-null float64
suspenso_resst          35735 non-null float64
saldo_interes           35735 non-null float64
clasificacion           35735 non-null object
tipo_de_credito         35735 non-null object
sector                  35735 non-null object
numero_garantia         35735 non-null int64
monto_garantia          35735 non-null float64
```

```
fecha_tasacion            35735 non-null object
plazo                     35735 non-null int64
dtypes: datetime64[ns](2), float64(17), int64(7), object(12)
memory usage: 10.4+ MB
```

**Como podemos ver, se trata de un dataframe compuesto por 4 tipos de columnas, tenemos datos en coma flotante (float64), datos enteros (int64), datos fechas (datetime64) y los datos tipo object son strings, es decir variables categóricas, las cuales posteriormente tendremos que linearizar**

```
#Linealizamos las variables categóricas usando un one-hot-enconding
#Primer paso es encontrar todas las variables categóricas, seran aquellas que tengan el tipo object
data.select_dtypes(include=['object']).head()
#El dataset tiene 12 variables categóricas. No linealizaremos _id,estado_prestamo y fecha_tasacion(realmente es fecha) porque se rán borradas.
```

| | _id | tipo_de_cliente | destino_credito | estado_civil | clase_de_persona | sexo | nacionalidad | estado_prestamo | clasificacion | tipo_de_cre |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 5db40b03718cc02f6009e361 | 511-ASALARIADO PRIVADO | CAÑA DE AZUCAR | SOLTERO(A) | FISICA | M | DOMINICANA | 4 -VIGENTE | B | M-MEN DEUD COMERC |
| 1 | 5db40b03718cc02f6009e362 | 511-ASALARIADO PRIVADO | SALON DE BELLEZA | SOLTERO(A) | FISICA | F | DOMINICANA | 4 -VIGENTE | B | M-MEN DEUD COMERC |
| 2 | 5db40b03718cc02f6009e363 | 511-ASALARIADO PRIVADO | ADQUISICION MOTOCICLETA EMPLEADO | SOLTERO(A) | FISICA | F | DOMINICANA | 4 -VIGENTE | B | O-CONSU |
| 3 | 5db40b03718cc02f6009e364 | 511-ASALARIADO PRIVADO | ADQUISICION MOTOCICLETA EMPLEADO | SOLTERO(A) | FISICA | M | DOMINICANA | 4 -VIGENTE | E | O-CONSU |
| 4 | 5db40b03718cc02f6009e365 | 511-ASALARIADO PRIVADO | ADQUISICION MOTOCICLETA EMPLEADO | SOLTERO(A) | FISICA | M | DOMINICANA | 4 -VIGENTE | B | O-CONSU |

```
data.describe(include='all')
```

| | _id | id_sucursal | id_cuenta | id_cliente | tipo_de_cliente | destino_credito | estado_civil | clase_de_persona | sexo | naciona |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 35735 | 35735.000000 | 3.573500e+04 | 35735.000000 | 35735 | 35735 | 35735 | 35735 | 35735 | 3 |
| unique | 35735 | NaN | NaN | NaN | 13 | 327 | 5 | 2 | 3 | |
| top | 5db40b0e718cc02f600a5bf5 | NaN | NaN | NaN | 511-ASALARIADO PRIVADO | ARROZ (FOMENTO) | SOLTERO(A) | FISICA | M | DOMINIC |
| freq | 1 | NaN | NaN | NaN | 34500 | 3545 | 23403 | 34886 | 28958 | 3 |
| first | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| last | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| mean | NaN | 15.340255 | 1.160822e+06 | 336292.706674 | NaN | NaN | NaN | NaN | NaN | |
| std | NaN | 9.957620 | 1.793543e+05 | 194714.259688 | NaN | NaN | NaN | NaN | NaN | |
| min | NaN | 1.000000 | 1.200000e+02 | 6.000000 | NaN | NaN | NaN | NaN | NaN | |
| 25% | NaN | 6.000000 | 1.107506e+06 | 183073.000000 | NaN | NaN | NaN | NaN | NaN | |
| 50% | NaN | 13.000000 | 1.203389e+06 | 329588.000000 | NaN | NaN | NaN | NaN | NaN | |
| 75% | NaN | 24.000000 | 1.269914e+06 | 515533.000000 | NaN | NaN | NaN | NaN | NaN | |
| max | NaN | 33.000000 | 1.303185e+06 | 665763.000000 | NaN | NaN | NaN | NaN | NaN | |

13 rows × 38 columns

```
data.shape
```

```
(35735, 38)
```

```
#Cambiamos el numero de la garantia a el campo garantia que indica si tiene-1 o no tiene-0.
data['garantia'] = 0
data.loc[data['numero_garantia'] != 0, 'garantia'] = 1
```

```
#Cambiamos el plazo de días a meses
data['plazo'] = round((data['plazo']/365), 2)
```

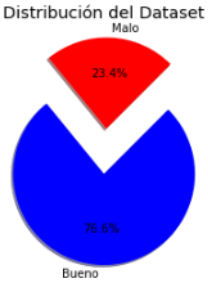
```
#Creamos nuestra variable objetivo identificando los clientes buenos-0 de los clientes malos-1.
data['condicion'] = 0
data.loc[data['estado_prestamo'].isin(['5 -VENCIDO','11-REESTRUCTURADO']), 'condicion'] = 1
```

**Distribución del Dataset**

```
data.condicion.value_counts()
```

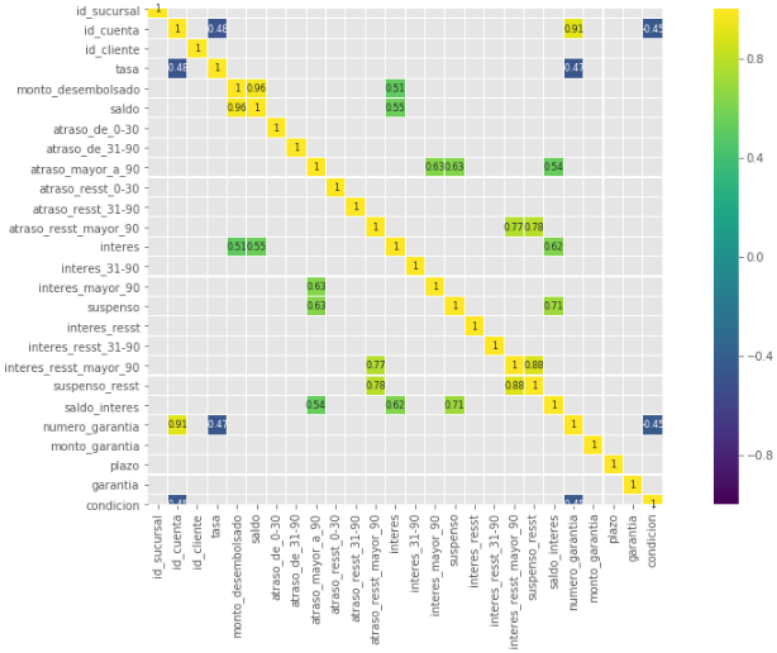
```
0    27372
1     8363
Name: condicion, dtype: int64
```

```
count_classes = pd.value_counts(data['condicion'], sort = True).sort_index()
labels = 'Malo', 'Bueno'
sizes = [count_classes[1]/(count_classes[1]+count_classes[0]), count_classes[0]/(count_classes[1]+count_classes[0])]
explode = (0, 0.5,)
colors = ['red', 'blue']
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, colors=colors, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=45)
ax1.axis('equal')
plt.title("Distribución del Dataset")
plt.show()
```




**Correlación de variables**

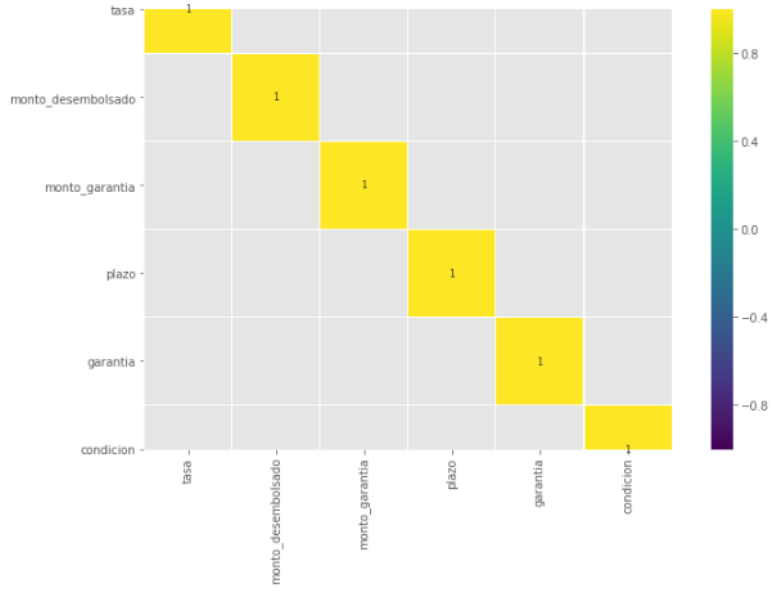
```
#Buscamos incluir en el modelo todas las variables que aporten una componente explicativa alta a la variable objetivo
#y que tengan una baja correlación entre ellas. Dos variables con una alta correlación significa que,
#con un grado de error bajo, una puede ser inferida a partir de la otra, y
#por tanto aportarán información redundante al modelo
corr_base = data.corr()
plt.figure(figsize=(16, 8))

sns.heatmap(corr_base[(corr_base >= 0.5) | (corr_base <= -0.4)],
            cmap='viridis', vmax=1.0, vmin=-1.0, linewidths=0.1,
            annot=True, annot_kws={"size": 8}, square=True);
```



**Eliminando las columnas que no aportan al modelo**

```
data.drop('_id', axis=1, inplace=True)
data.drop('id_sucursal', axis=1, inplace=True)
data.drop('id_cuenta', axis=1, inplace=True)
data.drop('id_cliente', axis=1, inplace=True)
data.drop('estado_prestamo', axis=1, inplace=True)
data.drop(['fecha_desembolso'], axis=1, inplace=True)
data.drop(['fecha_vencimiento'], axis=1, inplace=True)
data.drop(['saldo'], axis=1, inplace=True)
data.drop(['atraso_de_0-30'], axis=1, inplace=True)
data.drop(['atraso_de_31-90'], axis=1, inplace=True)
data.drop(['atraso_mayor_a_90'], axis=1, inplace=True)
data.drop(['atraso_resst_0-30'], axis=1, inplace=True)
data.drop(['atraso_resst_31-90'], axis=1, inplace=True)
data.drop(['atraso_resst_mayor_90'], axis=1, inplace=True)
data.drop(['interes'], axis=1, inplace=True)
data.drop(['interes_31-90'], axis=1, inplace=True)
data.drop(['interes_mayor_90'], axis=1, inplace=True)
data.drop(['suspenso'], axis=1, inplace=True)
data.drop(['interes_resst'], axis=1, inplace=True)
data.drop(['interes_resst_31-90'], axis=1, inplace=True)
data.drop(['interes_resst_mayor_90'], axis=1, inplace=True)
data.drop(['suspenso_resst'], axis=1, inplace=True)
data.drop(['saldo_interes'], axis=1, inplace=True)
data.drop(['numero_garantia'], axis=1, inplace=True)
data.drop(['fecha_tasacion'], axis=1, inplace=True)
```

```
corr_base = data.corr()
plt.figure(figsize=(12, 7))

sns.heatmap(corr_base[(corr_base >= 0.5) | (corr_base <= -0.4)],
            cmap='viridis', vmax=1.0, vmin=-1.0, linewidths=0.1,
            annot=True, annot_kws={"size": 8}, square=True);
```



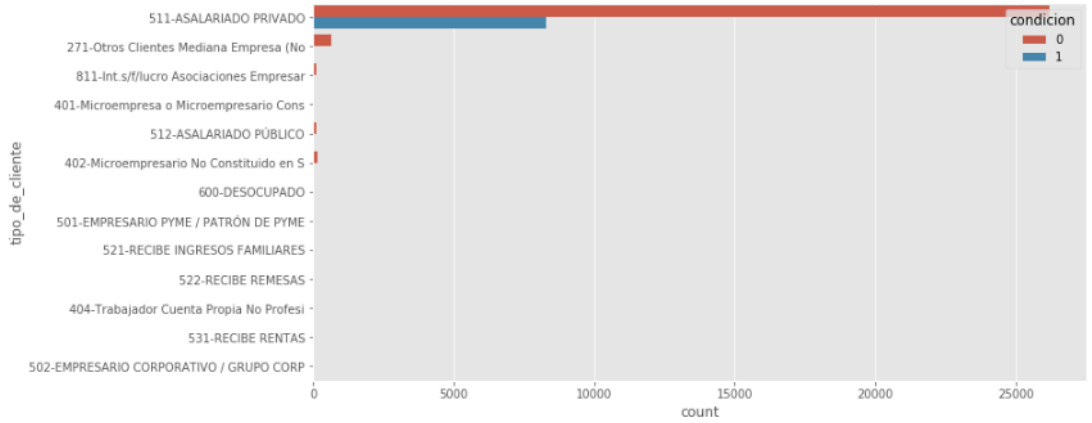
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35735 entries, 0 to 35734
Data columns (total 15 columns):
tipo_de_cliente       35735 non-null object
destino_credito       35735 non-null object
estado_civil          35735 non-null object
clase_de_persona      35735 non-null object
sexo                  35735 non-null object
nacionalidad          35735 non-null object
tasa                  35735 non-null float64
monto_desembolsado    35735 non-null float64
clasificacion         35735 non-null object
tipo_de_credito       35735 non-null object
sector                35735 non-null object
monto_garantia        35735 non-null float64
plazo                 35735 non-null float64
garantia              35735 non-null int64
condicion             35735 non-null int64
dtypes: float64(4), int64(2), object(9)
memory usage: 4.1+ MB
```

### Variables categóricas

```
# Convirtiendo las variables categóricas a numéricas
plt.figure(figsize=(12, 6))
sns.countplot(y='tipo_de_cliente', data=data, hue='condicion')
```

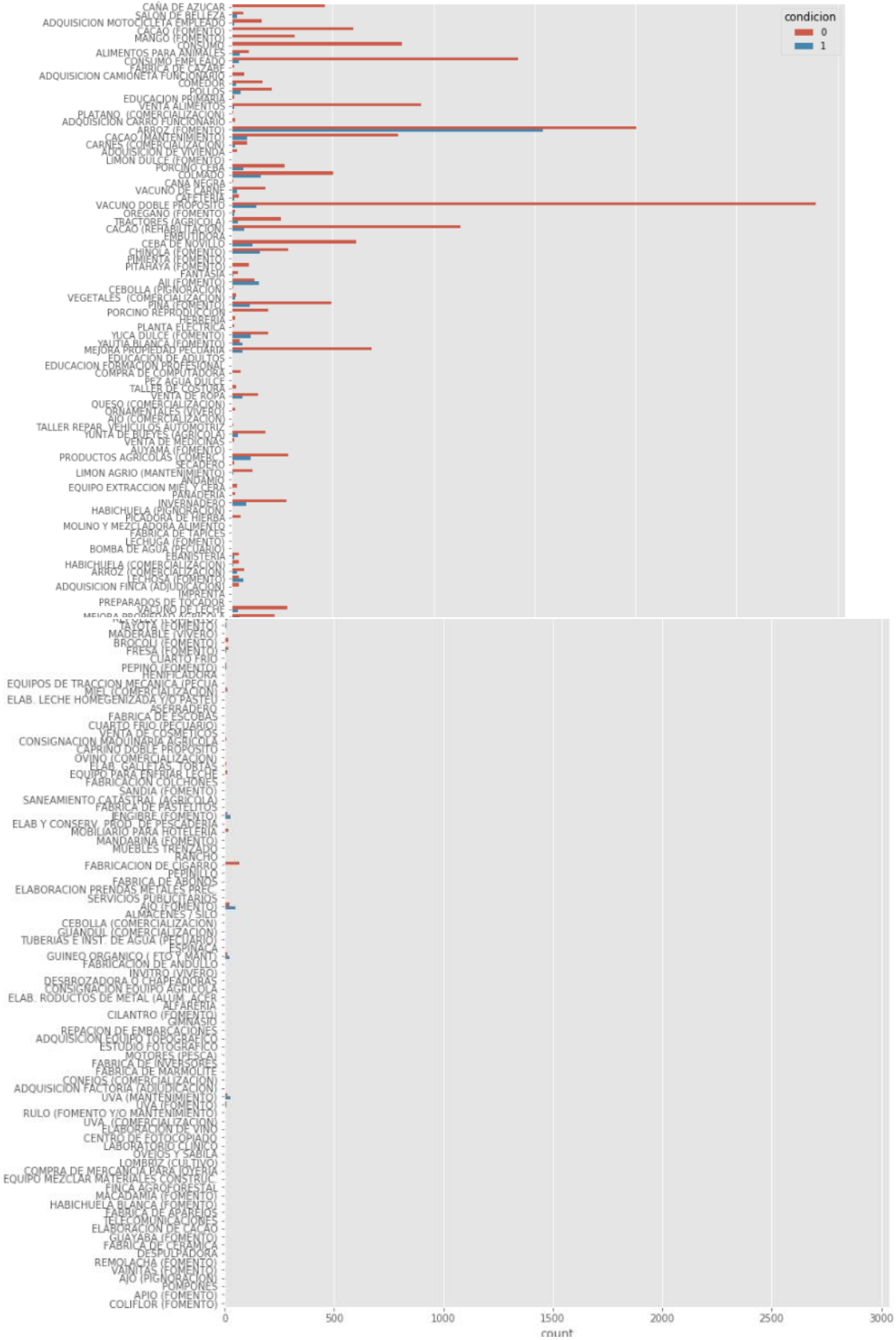
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624c683c8>
```




```
data = pd.concat([data, pd.get_dummies(data['tipo_de_cliente'],
                                       prefix='tipo')], axis=1).drop(['tipo_de_cliente'], axis=1)
```

```
plt.figure(figsize=(12, 50))
sns.countplot(y='destino_credito', data=data, hue='condicion')
```

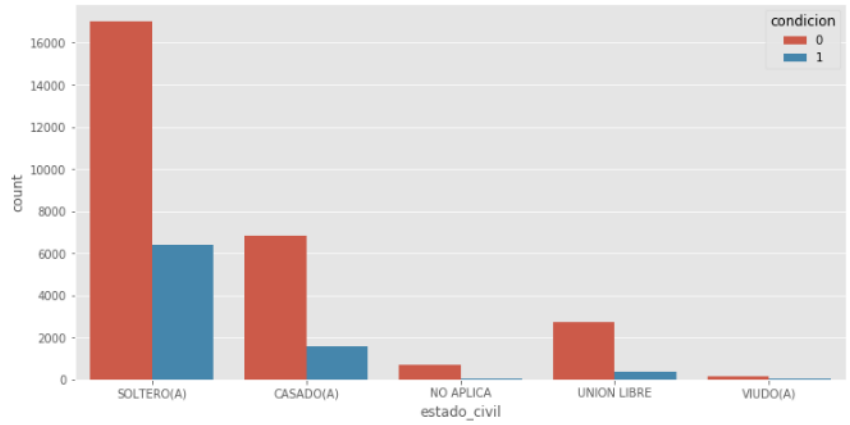
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624cb79c8>
```


condicion
0
1
count
0
500
1000
1500
2000
2500
3000

```
le = LabelEncoder()
data['destino_credito'] = le.fit_transform(data.destino_credito.values)
```

```
plt.figure(figsize=(12, 6))
sns.countplot(x='estado_civil', data=data, hue='condicion')
```

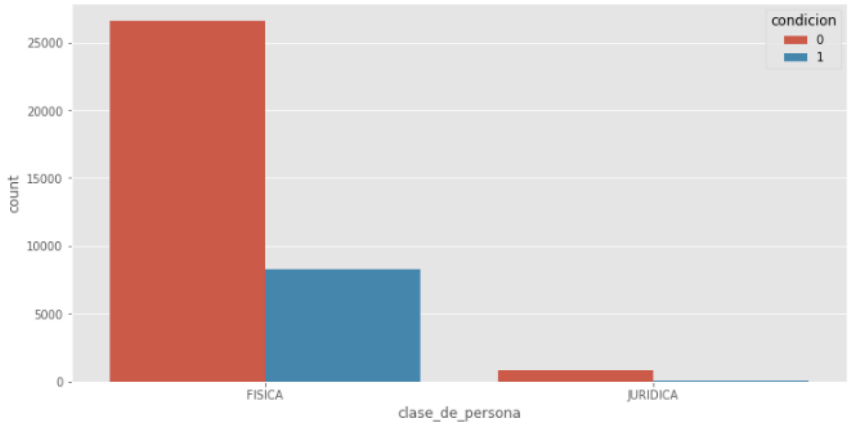
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624ed6ac8>
```




```
data = pd.concat([data, pd.get_dummies(data['estado_civil'],
                                       prefix='civil')], axis=1).drop(['estado_civil'], axis=1)
```

```
plt.figure(figsize=(12, 6))
sns.countplot(x='clase_de_persona', data=data, hue='condicion')
```

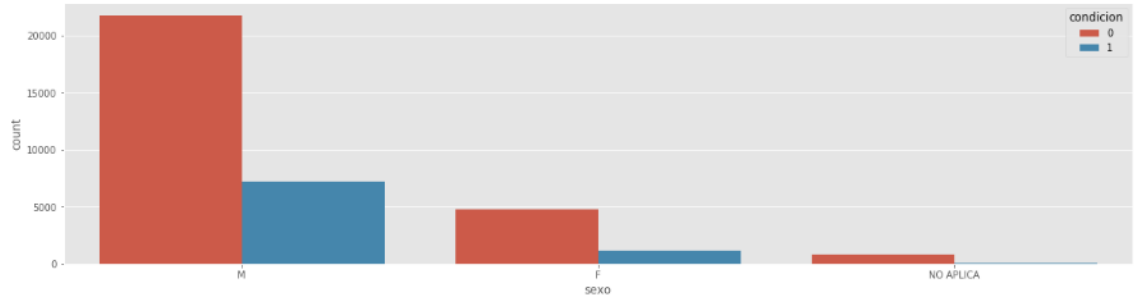
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624c4e3c8>
```




```
data = pd.concat([data, pd.get_dummies(data['clase_de_persona'], prefix='clase',
                                       drop_first=True)], axis=1).drop(['clase_de_persona'], axis=1)
```

```
plt.figure(figsize=(20, 5))
sns.countplot(x='sexo', data=data, hue='condicion')
```

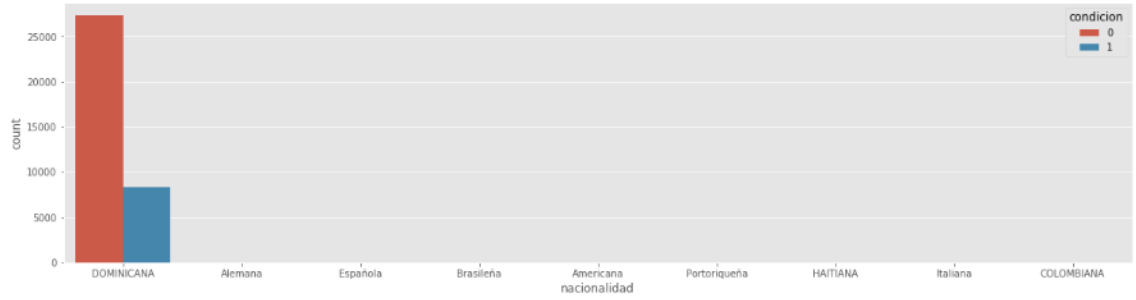
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624bfdd48>
```




```
data = pd.concat([data, pd.get_dummies(
    data['sexo'], prefix='sexo')], axis=1).drop(['sexo'], axis=1)
```

```
plt.figure(figsize=(20, 5))
sns.countplot(x='nacionalidad', data=data, hue='condicion')
```

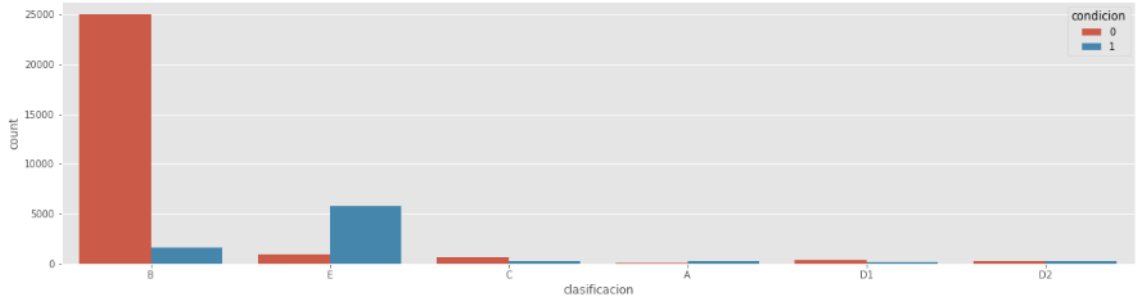
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624bce448>
```




```
data = pd.concat([data, pd.get_dummies(data['nacionalidad'],
                                       prefix='nacion')], axis=1).drop(['nacionalidad'], axis=1)
```

```
plt.figure(figsize=(20, 5))
sns.countplot(x='clasificacion', data=data, hue='condicion')
```

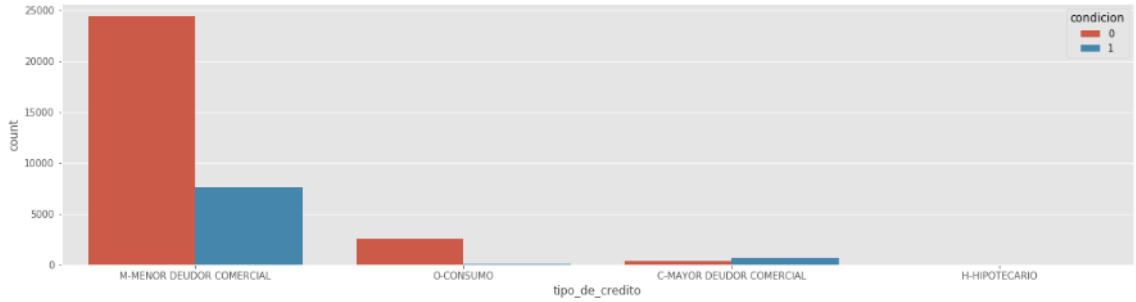
```
<matplotlib.axes._subplots.AxesSubplot at 0x21624b6a788>
```




```
data = pd.concat([data, pd.get_dummies(data['clasificacion'],
                                       prefix='clasificacion')], axis=1).drop(['clasificacion'], axis=1)
```

```
plt.figure(figsize=(20, 5))
sns.countplot(x='tipo_de_credito', data=data, hue='condicion')
```

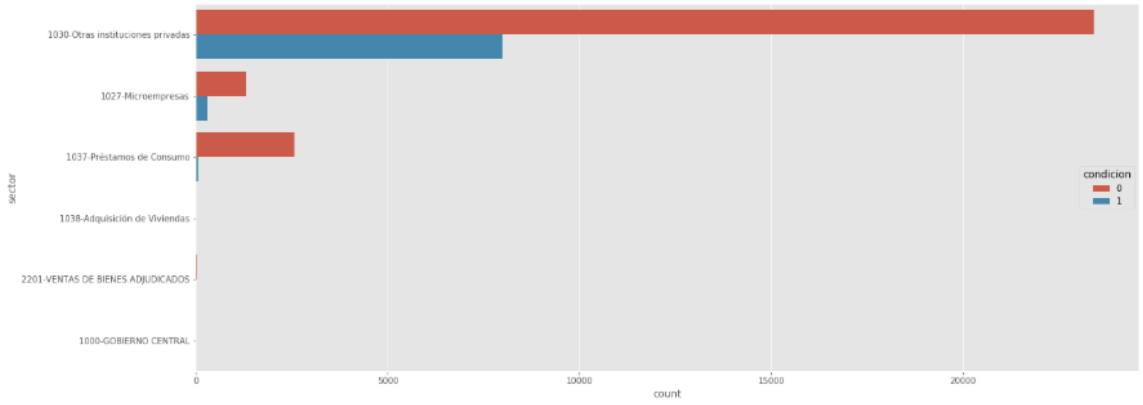
```
<matplotlib.axes._subplots.AxesSubplot at 0x216250f1788>
```




```
data = pd.concat([data, pd.get_dummies(data['tipo_de_credito'],
                                       prefix='tipo_credito')], axis=1).drop(['tipo_de_credito'], axis=1)
```

```
plt.figure(figsize=(20, 8))
sns.countplot(y='sector', data=data, hue='condicion')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x216250eccc8>
```

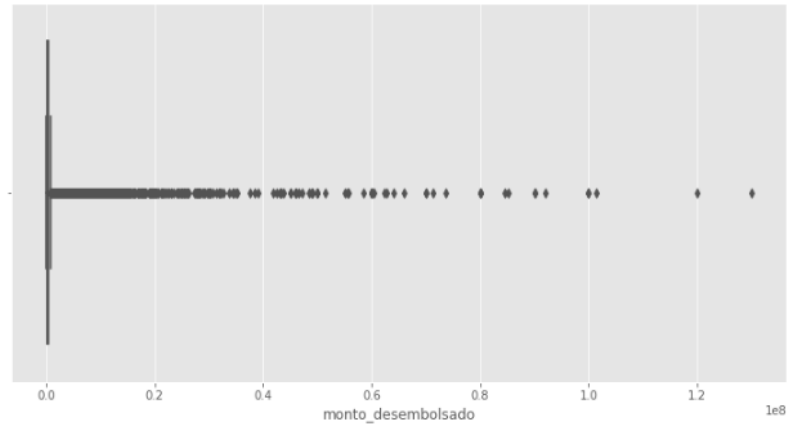



```
data = pd.concat([data, pd.get_dummies(
    data['sector'], prefix='sector')], axis=1).drop(['sector'], axis=1)
```

**Normalización del dataset convirtiendo la escala entre -1 y 1 de las variables numéricas**

```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['monto_desembolsado'])
```

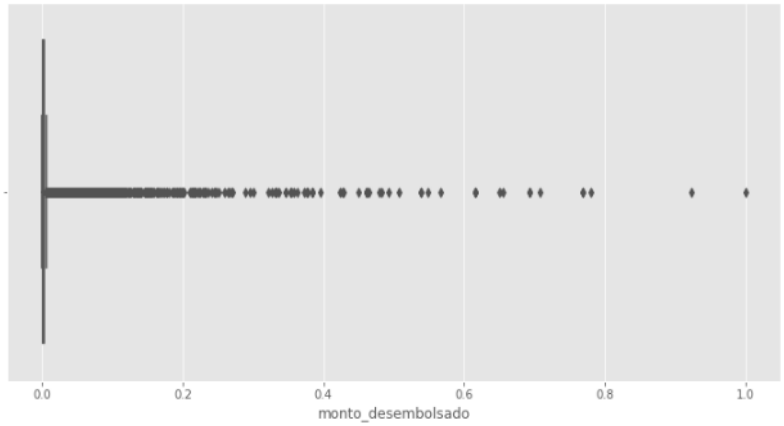
```
<matplotlib.axes._subplots.AxesSubplot at 0x21625153d08>
```




```
data['monto_desembolsado'] = scaler.fit_transform(
    data['monto_desembolsado'].values.reshape(-1, 1))
```

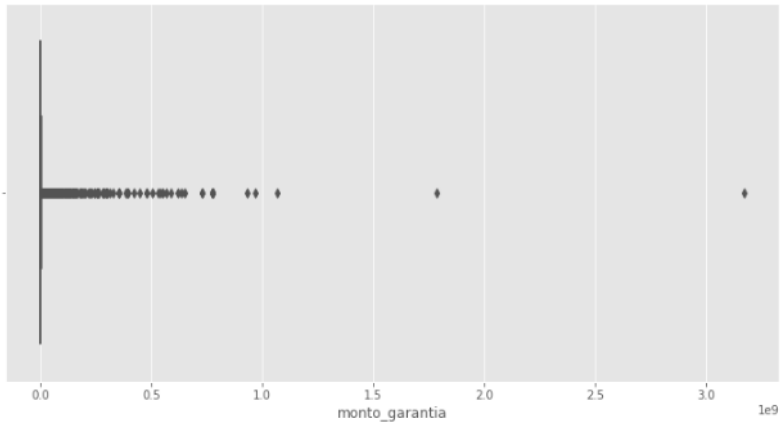
```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['monto_desembolsado'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2162517f7c8>
```




```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['monto_garantia'])
```

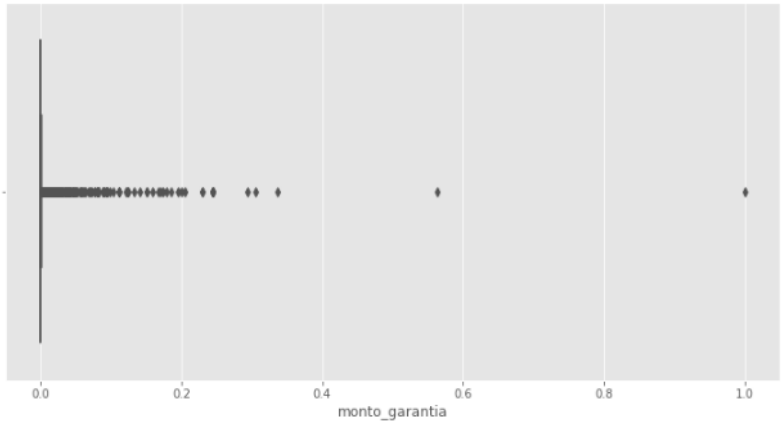
```
<matplotlib.axes._subplots.AxesSubplot at 0x216251b7588>
```




```
data['monto_garantia'] = scaler.fit_transform(
    data['monto_garantia'].values.reshape(-1, 1))
```

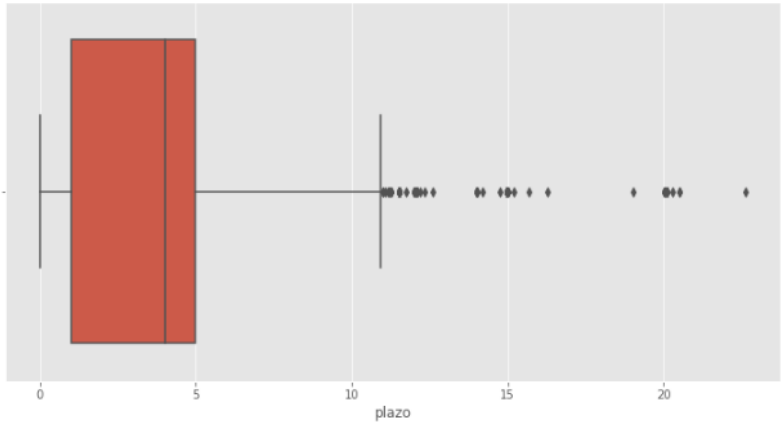
```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['monto_garantia'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x216251e4748>
```




```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['plazo'])
```

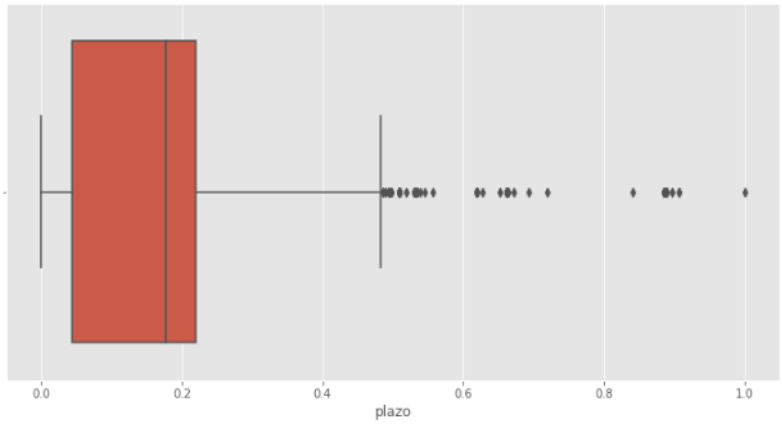
```
<matplotlib.axes._subplots.AxesSubplot at 0x2162518c808>
```




```
data['plazo'] = scaler.fit_transform(data['plazo'].values.reshape(-1, 1))
```

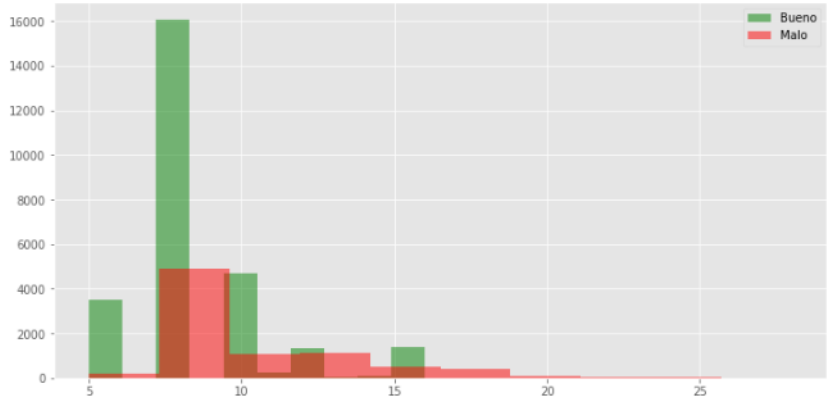
```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['plazo'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x21625163048>
```



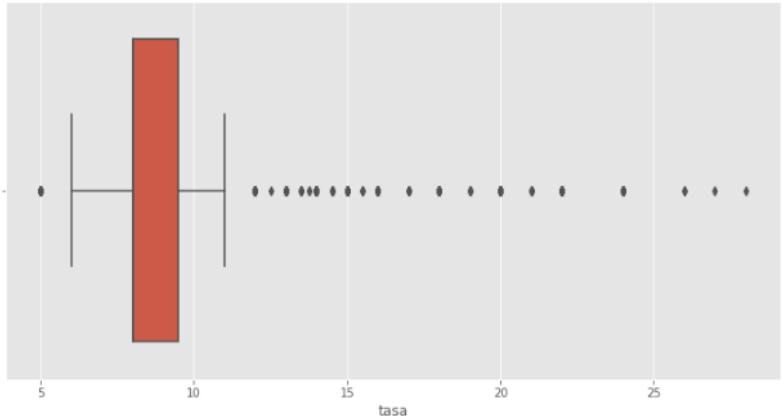

```
plt.figure(figsize=(12, 6))
plt.hist(data[data['condicion'] == 0]['tasa'],
         color='green', alpha=0.5, label='Bueno')
plt.hist(data[data['condicion'] == 1]['tasa'],
         color='red', alpha=0.5, label='Malo')

plt.legend()
```

```
<matplotlib.legend.Legend at 0x2162527f548>
```




```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['tasa'])
```

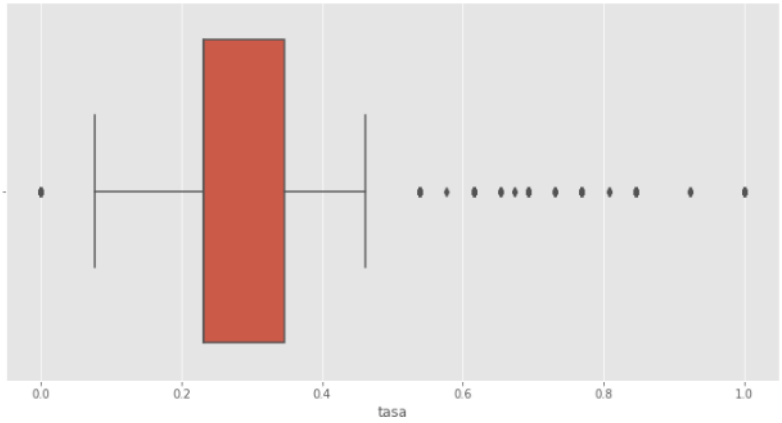
```
<matplotlib.axes._subplots.AxesSubplot at 0x21625243a08>
```




```
outliers = data[data['tasa'] > data['tasa'].quantile(.99)].index
data.loc[outliers, 'tasa'] = data['tasa'].quantile(.99)
data['tasa'] = scaler.fit_transform(data['tasa'].values.reshape(-1, 1))
```

```
plt.figure(figsize=(12, 6))
sns.boxplot(x=data['tasa'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x216252e3308>
```




```
data.info()
```

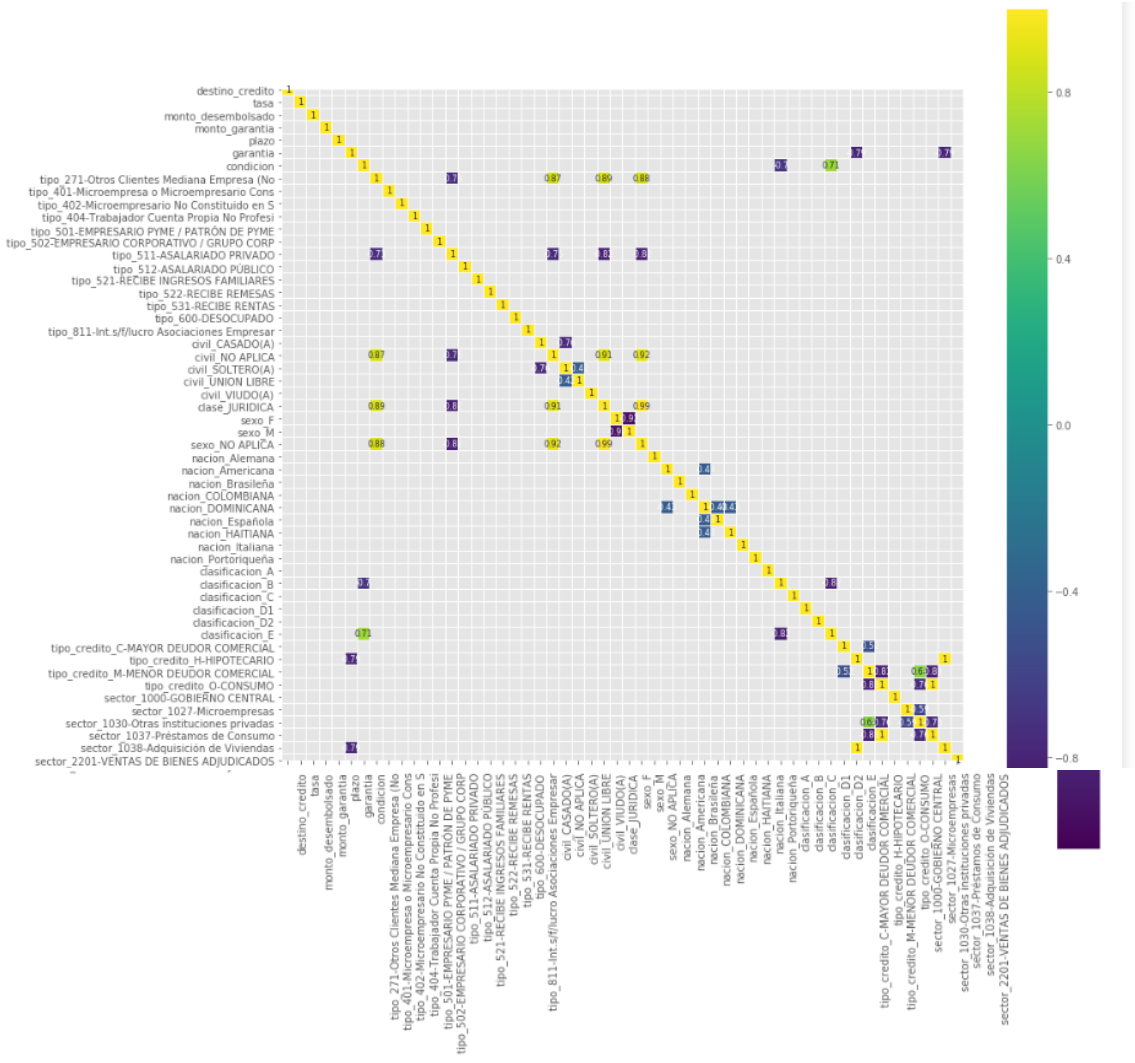
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35735 entries, 0 to 35734
Data columns (total 54 columns):
destino_credito                                 35735 non-null int32
tasa                                            35735 non-null float64
monto_desembolsado                              35735 non-null float64
monto_garantia                                  35735 non-null float64
plazo                                           35735 non-null float64
garantia                                        35735 non-null int64
condicion                                       35735 non-null int64
tipo_271-Otros Clientes Mediana Empresa (No     35735 non-null uint8
tipo_401-Microempresa o Microempresario Cons    35735 non-null uint8
tipo_402-Microempresario No Constituido en S    35735 non-null uint8
tipo_404-Trabajador Cuenta Propia No Profesi    35735 non-null uint8
tipo_501-EMPRESARIO PYME / PATRÓN DE PYME       35735 non-null uint8
tipo_502-EMPRESARIO CORPORATIVO / GRUPO CORP    35735 non-null uint8
tipo_511-ASALARIADO PRIVADO                     35735 non-null uint8
tipo_512-ASALARIADO PÚBLICO                     35735 non-null uint8
tipo_521-RECIBE INGRESOS FAMILIARES             35735 non-null uint8
tipo_522-RECIBE REMESAS                         35735 non-null uint8
tipo_531-RECIBE RENTAS                          35735 non-null uint8
tipo_600-DESOCUPADO                             35735 non-null uint8
tipo_811-Int.s/f/lucro Asociaciones Empresar    35735 non-null uint8
civil_CASADO(A)                                 35735 non-null uint8
civil_NO APLICA                                 35735 non-null uint8
civil_SOLTERO(A)                                35735 non-null uint8
civil_UNION LIBRE                               35735 non-null uint8
civil_VIUDO(A)                                  35735 non-null uint8
clase_JURIDICA                                  35735 non-null uint8
sexo_F                                          35735 non-null uint8
sexo_M                                          35735 non-null uint8
sexo_NO APLICA                                  35735 non-null uint8
nacion_Alemana                                  35735 non-null uint8
nacion_Americana                                35735 non-null uint8
nacion_Brasileña                                35735 non-null uint8
nacion_COLOMBIANA                               35735 non-null uint8
nacion_DOMINICANA                               35735 non-null uint8
nacion_Española                                 35735 non-null uint8
nacion_HAITIANA                                 35735 non-null uint8
nacion_Italiana                                 35735 non-null uint8
nacion_Portoriqueña                             35735 non-null uint8
clasificacion_A                                 35735 non-null uint8
clasificacion_B                                 35735 non-null uint8
clasificacion_C                                 35735 non-null uint8
clasificacion_D1                                35735 non-null uint8
clasificacion_D2                                35735 non-null uint8
clasificacion_E                                 35735 non-null uint8
tipo_credito_C-MAYOR DEUDOR COMERCIAL           35735 non-null uint8
tipo_credito_H-HIPOTECARIO                      35735 non-null uint8
tipo_credito_M-MENOR DEUDOR COMERCIAL           35735 non-null uint8
tipo_credito_O-CONSUMO                          35735 non-null uint8
sector_1000-GOBIERNO CENTRAL                    35735 non-null uint8
sector_1027-Microempresas                       35735 non-null uint8
sector_1030-Otras instituciones privadas        35735 non-null uint8
sector_1037-Préstamos de Consumo                35735 non-null uint8
sector_1038-Adquisición de Viviendas            35735 non-null uint8
sector_2201-VENTAS DE BIENES ADJUDICADOS        35735 non-null uint8
dtypes: float64(4), int32(1), int64(2), uint8(47)
memory usage: 3.4 MB
```

```
data.describe(include='all')
```

| | destino_credito | tasa | monto_desembolsado | monto_garantia | plazo | garantia | condicion | tipo_271-Otros Clientes Mediana Empresa (No | tipo_401-Microempresa o Microempresario Cons | Microen No Co |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 35735.000000 | 35735.000000 | 35735.000000 | 35735.000000 | 35735.000000 | 35735.000000 | 35735.000000 | 35735.000000 | 35735.000000 | 3573 |
| mean | 146.148034 | 0.294476 | 0.005771 | 0.001021 | 0.156417 | 0.998825 | 0.234028 | 0.018749 | 0.001175 | |
| std | 110.002639 | 0.192871 | 0.025063 | 0.009087 | 0.106283 | 0.034263 | 0.423396 | 0.135640 | 0.034263 | |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 25% | 42.000000 | 0.230769 | 0.000769 | 0.000106 | 0.043786 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 50% | 93.000000 | 0.230769 | 0.001544 | 0.000222 | 0.176471 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 75% | 247.000000 | 0.346154 | 0.003306 | 0.000526 | 0.220699 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | |
| max | 326.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | |

8 rows × 54 columns

```
corr_base = data.corr()
plt.figure(figsize=(15, 15))

sns.heatmap(corr_base[(corr_base >= 0.5) | (corr_base <= -0.4)],
            cmap='viridis', vmax=1.0, vmin=-1.0, linewidths=0.1,
            annot=True, annot_kws={"size": 8}, square=True);
```




## Dividimos la data de entrenamiento y de prueba

```
sum(data['condicion']) / len(data)
```

```
0.23402826360710788
```

```
X = data.drop(['condicion'], axis=1)
y = data['condicion']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=random)
```

```
rf_base = RandomForestClassifier(n_estimators=100, random_state=random)
ada_base = AdaBoostClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100, random_state=random)
neigh_base = KNeighborsClassifier(n_neighbors=3)
```

```
rf_base.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=42, verbose=0,
                       warm_start=False)
```

```
ada_base.fit(X_train, y_train)
```

```
AdaBoostClassifier(algorithm='SAMME.R',
                   base_estimator=DecisionTreeClassifier(class_weight=None,
                                                         criterion='gini',
                                                         max_depth=None,
                                                         max_features=None,
                                                         max_leaf_nodes=None,
                                                         min_impurity_decrease=0.0,
                                                         min_impurity_split=None,
                                                         min_samples_leaf=1,
                                                         min_samples_split=2,
                                                         min_weight_fraction_leaf=0.0,
                                                         presort=False,
                                                         random_state=None,
                                                         splitter='best'),
                   learning_rate=1.0, n_estimators=100, random_state=42)
```

```
neigh_base.fit(X_train, y_train)
```

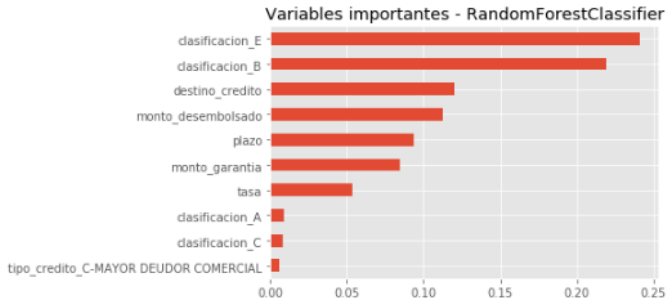
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

```
y_rf_base_pred = rf_base.predict(X_test)
```

```
y_ada_base_pred = ada_base.predict(X_test)
```

```
y_neigh_base_pred = neigh_base.predict(X_test)
```

```
#Gráfico de las variables importantes para una mejor visualización
variables_importantes = pd.Series(rf_base.feature_importances_, index=X_train.columns)
variables_importantes.nlargest(10).sort_values().plot(kind='barh')
plt.title("Variables importantes - RandomForestClassifier")
plt.show()
```




```
#Gráfico de las variables importantes para una mejor visualización
np.seterr(divide='ignore', invalid='ignore')
variables_importantes = pd.Series(ada_base.feature_importances_, index=X_train.columns)
variables_importantes.nlargest(10).sort_values().plot(kind='barh')
plt.title("Variables importantes - AdaBoostClassifier")
plt.show()
```


```
print("RandomForestClassifier")
print(confusion_matrix(y_test, y_rf_base_pred))
print('\n')
print("AdaBoostClassifier")
print(confusion_matrix(y_test, y_ada_base_pred))
print('\n')
print("KNeighborsClassifier")
print(confusion_matrix(y_test, y_neigh_base_pred))
```

```
RandomForestClassifier
[[7967  226]
 [ 302 2226]]


AdaBoostClassifier
[[7994  199]
 [ 277 2251]]


KNeighborsClassifier
[[7983  210]
 [ 348 2180]]
```

```
#Exactitud
print("{0:.2f}%".format(accuracy_score(y_test, y_rf_base_pred)*100))
print("{0:.2f}%".format(accuracy_score(y_test, y_ada_base_pred)*100))
print("{0:.2f}%".format(accuracy_score(y_test, y_neigh_base_pred)*100))
```

```
95.08%
95.56%
94.80%
```

```
#Precisión
print("{0:.2f}%".format(precision_score(y_test, y_rf_base_pred)*100))
print("{0:.2f}%".format(precision_score(y_test, y_ada_base_pred)*100))
print("{0:.2f}%".format(precision_score(y_test, y_neigh_base_pred)*100))
```

```
90.78%
91.88%
91.21%
```

```
#Sensibilidad
print("{0:.2f}%".format(recall_score(y_test, y_rf_base_pred)*100))
print("{0:.2f}%".format(recall_score(y_test, y_ada_base_pred)*100))
print("{0:.2f}%".format(recall_score(y_test, y_neigh_base_pred)*100))
```

```
88.05%
89.04%
86.23%
```

```
#F1-score
print("{0:.2f}%".format(f1_score(y_test, y_rf_base_pred)*100))
print("{0:.2f}%".format(f1_score(y_test, y_ada_base_pred)*100))
print("{0:.2f}%".format(f1_score(y_test, y_neigh_base_pred)*100))
```

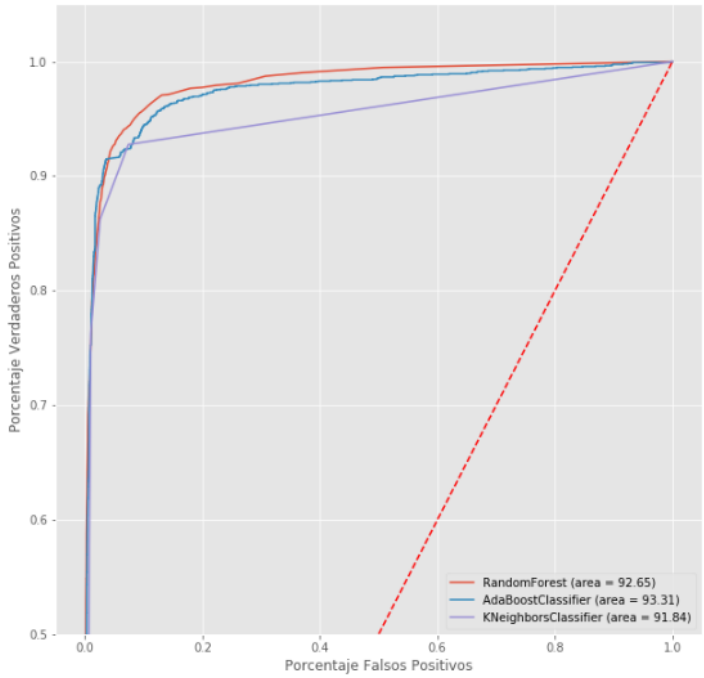
```
89.40%
90.44%
88.65%
```

```
rf_ROC_auc = roc_auc_score(y_test, y_rf_base_pred)
fpr_rf, tpr_rf, thresholds_rf = roc_curve(
    y_test, rf_base.predict_proba(X_test)[:, 1])
```

```
ada_ROC_auc = roc_auc_score(y_test, y_ada_base_pred)
fpr_ada, tpr_ada, thresholds_ada = roc_curve(
    y_test, ada_base.predict_proba(X_test)[:, 1])
```

```
neigh_ROC_auc = roc_auc_score(y_test, y_neigh_base_pred)
fpr_neigh, tpr_neigh, thresholds_neigh = roc_curve(
    y_test, neigh_base.predict_proba(X_test)[:, 1])
```

```
plt.figure(figsize=(10, 10))
plt.plot(fpr_rf, tpr_rf, label="RandomForest (area = %0.2f)" % (rf_ROC_auc * 100))
plt.plot(fpr_ada, tpr_ada, label="AdaBoostClassifier (area = %0.2f)" % (ada_ROC_auc * 100))
plt.plot(fpr_neigh, tpr_neigh, label="KNeighborsClassifier (area = %0.2f)" % (neigh_ROC_auc * 100))
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([-0.05, 1.05])
plt.ylim([0.5, 1.05])
plt.xlabel('Porcentaje Falsos Positivos')
plt.ylabel('Porcentaje Verdaderos Positivos')
plt.legend(loc="lower right")
plt.savefig('ROC')
plt.show()
```




```
joblib.dump(ada_base, 'modelo.pkl')
```

```
['modelo.pkl']
```

```
X_test.to_csv('X_test.csv', index=False)
y_test.to_csv('y_test.csv', index=False, header=False)
```

WebApp

```
FROM python:3

# development, review, staging, demo, production
ARG ENVIRONMENT=production

ENV BUILD_TAG=$TAG

WORKDIR /app

COPY requirements/ ./requirements
# TODO support env requirements
RUN pip install --no-cache-dir -r requirements/dev.txt
RUN mkdir -p /var/log/uwsgi

# copy production files
COPY environments/production/etc /etc
COPY environments/production/bin/envs.sh ./bin/

COPY src ./
RUN chown -R www-data:www-data ./

COPY environments/production/config ./config
COPY environments/${ENVIRONMENT}/config ./config

EXPOSE 80

CMD ["/app/start.sh"]
```

Docker File

```yaml
version: '2'

services:
  mongodb:
    restart: always
    image: registry.gitlab.com/mongodb:latest
    ports:
      - "27017:27016"
  bcs:
    build:
      context: .
      dockerfile: ./dev/Dockerfile
    depends_on:
      - "mongodb"
    ports:
      - "80:80"
    stdin_open: true
    tty: true
    volumes:
      - "./:/app"
    links:
      - mongodb:mongo-db-connection
```

docker-compose.yml

```
services:
- docker:18-dind

variables:
  # build container docker client running on a k8s pod,
  DOCKER_HOST: "tcp://localhost:2375"
  DOCKER_DRIVER: "overlay2"
  NAMESPACE: default

stages:
  - test
  - build
  - deploy

.build: &build
  stage: build
  script:
    # Build the image
    # Run tests
    - docker build --build-arg ENVIRONMENT=$ENVIRONMENT --build-arg TAG=$TAG .

    # Deploy to the gitlab registry
    - docker push registry.gitlab.com/$CI_PROJECT_PATH:$TAG

.deploy: &deploy
  stage: deploy
  script:
    - echo Tag $TAG
    - echo Release name $RELEASE_NAME
    - helm init --client-only
    - helm upgrade --install --atomic --timeout 1200 --set image.repository=
    registry.gitlab.com/$CI_PROJECT_PATH,image.tag=$TAG
    # print logs from deployment first running pod
    - kubectl -n default logs $(kubectl -n default get po | grep -E
    "^${DEPLOY_NAME}.*Running" | head -n 1 | awk '{print $1}')

test.branch:
  stage: test
  variables:
    COMPOSE_HTTP_TIMEOUT: "600"
  script:
    # Build the image
    # Test are run as part of docker build
    - docker-compose  --verbose up -d
    - docker exec -t bcs-app_1 bash ./bin/test.sh
  tags:
    - bcs-dev
```

.gitlab-ci.yaml