**FGA**

**Mano de una persona

Descripción generada automáticamente con confianza media**Demo Segmentación RFM

**Manual o Guía de Usuario**

**F-P17-09**

**Ver. 01**

|  |  |  |  |
| --- | --- | --- | --- |
| Control de Cambios | | | |
| Encargado | **Fecha** | **Versión** | **Observaciones** |
| Carolina Torres Zapata | 12/12/2024 | 1.0 | Versión inicial |

[**1. Introducción 4**](#_Toc184987868)

[**2. Estructura del Proyecto RFM 5**](#_Toc184987869)

[2.1. ¿Cómo Funciona el Proyecto? 5](#_Toc184987870)

[**3. Configuración Flexible con YAML 5**](#_Toc184987871)

[**4. Implementación RFM 6**](#_Toc184987872)

[4.1. Implementación Local (Computadora Local) 6](#_Toc184987873)

[4.2. Implementación On-Premise 7](#_Toc184987874)

[4.3. Implementación en Nube 7](#_Toc184987875)

[**5. Visualización e Integración con Herramientas de BI: 8**](#_Toc184987876)

[**6. Análisis Exploratorio 9**](#_Toc184987877)

# 

# Introducción

El análisis RFM (Recencia, Frecuencia, Valor Monetario) es una metodología ampliamente utilizada para segmentar a los clientes en función de su comportamiento de compra. Este análisis permite identificar a los clientes más valiosos, optimizar las estrategias de marketing y fortalecer la relación con los consumidores.

Mediante el análisis RFM, los clientes se agrupan según tres factores fundamentales:

* **Recencia (R)**: ¿Cuánto tiempo ha pasado desde la última compra del cliente?
* **Frecuencia (F)**: ¿Con qué frecuencia realiza compras el cliente?
* **Valor Monetario (M)**: ¿Cuánto gasta el cliente en promedio por compra?

El propósito de esta técnica es ayudar a las empresas a tomar decisiones más informadas sobre cómo interactuar con cada grupo de clientes, lo que les permite personalizar sus estrategias de marketing y fidelización de manera más eficiente.

Este manual está diseñado para guiar a los usuarios a través del proceso de implementación del análisis RFM, abordando específicamente la puesta en marcha de un modelo RFM construido en Python a partir de análisis exploratorios de los datos.

La implementación del RFM depende directamente de la infraestructura existente en la empresa, la volumetría de datos a procesar y los recursos tecnológicos disponibles. Estas consideraciones son fundamentales para seleccionar la arquitectura más adecuada, ya sea en local, en servidores on-premise, en la nube o híbrida.

.

# Estructura del Proyecto RFM

Este proyecto de análisis RFM se ha diseñado de manera modular y flexible, lo que facilita su adaptación a diferentes necesidades de cada cliente. Gracias a su estructura modular, cada parte del proceso se encuentra separada, permitiendo que se pueda ajustar fácilmente según los requerimientos específicos del cliente.

## ¿Cómo Funciona el Proyecto?

El proceso se divide en varias partes que trabajan de manera conjunta, pero de forma independiente. Cada parte del análisis RFM está contenida en módulos, lo que permite ajustarlo según las necesidades del cliente:

**Carga de Datos:** Aquí es donde se cargan los datos desde diferentes fuentes, como archivos Excel, CSV o bases de datos. Esta parte del proceso está configurada para ser fácilmente adaptable, sin importar el formato en que se encuentren los datos.

**Preprocesamiento:** Una vez que los datos están cargados, esta parte se encarga de limpiarlos y ajustarlos. Si hay valores faltantes o inconsistencias, este módulo se asegura de que todo esté en orden para que el análisis sea preciso.

**Cálculo de RFM**: Este módulo es el corazón del análisis, donde se calculan las métricas clave de recencia, frecuencia y valor monetario de cada cliente. A partir de estos cálculos, se asignan las puntuaciones que permiten segmentar a los clientes en diferentes grupos.

A screenshot of a diagram

Description automatically generated

# Configuración Flexible con YAML

Una de las ventajas más importantes de este proyecto es que se configura a través de un archivo YAML. Este archivo centraliza todos los parámetros importantes que definen cómo se debe hacer el análisis, tales como:

* Método para calcular el score (promedio, suma, etc.).
* Rangos y límites para segmentar a los clientes en grupos como "Platino", "Oro", "Plata", etc.
* Periodo de análisis: El rango de tiempo que se utilizará para calcular las métricas.

Gracias al archivo YAML, se puede modificar estos parámetros sin necesidad de tocar el código del proyecto. Solo es necesario actualizar el archivo con los valores deseados y ejecutar el análisis nuevamente**.**

# Implementación RFM

La implementación del proyecto RFM puede llevarse a cabo en diferentes entornos, dependiendo de la infraestructura del cliente, la volumetría de los datos y los recursos disponibles. Las opciones disponibles son Local, On-Premise, Nube e Híbrida, y la elección de cada una dependerá de los factores técnicos y operativos del cliente. A continuación, se describen las diferentes opciones de implementación:

## Implementación Local (Computadora Local)

La opción Local es ideal para clientes que cuentan con una infraestructura interna básica y prefieren ejecutar el análisis RFM directamente en sus propios equipos, como una computadora personal o servidor pequeño. Este enfoque es adecuado cuando los datos no son de gran volumen o cuando el cliente desea realizar el análisis sin depender de servidores externos.

A diagram of a computer program

Description automatically generated with medium confidence

**Pasos para la Implementación**:

* **Instalación de Python**: El cliente debe asegurarse de que su máquina local tenga Python instalado, junto con todas las librerías necesarias.
* **Configuración del archivo YAML**: El archivo de configuración debe ser ajustado para acceder a los datos locales (archivos CSV, Excel, etc.) y para definir los parámetros del análisis.

El análisis exploratorio realizado previamente es el insumo para realizar las configuraciones en el archivo Yaml.

* **Automatización**: Para automatizar el proceso de ejecución, el cliente puede usar el Programador de Tareas de Windows o cron jobs en sistemas Linux, para ejecutar el script en intervalos programados (por ejemplo, de forma mensual).

**Beneficios**:

* **Fácil de implementar**: Ideal para clientes que no cuentan con grandes volúmenes de datos o que necesitan un proceso simple y directo.
* **Bajo costo**: No requiere inversión en infraestructura externa ni servicios en la nube.
* **Control local**: Todo el análisis se realiza en el equipo local del cliente, sin depender de recursos externos.

## Implementación On-Premise

La opción On-Premise es adecuada para clientes que cuentan con infraestructura interna robusta, como servidores dedicados o centros de datos, y desean mantener el análisis RFM completamente dentro de sus propios recursos. Esta opción es ideal cuando el cliente maneja grandes volúmenes de datos o requiere mayor control sobre el proceso.

A diagram of a computer program

Description automatically generated with medium confidence

**Pasos para la Implementación:**

* **Instalación de Python:** Garantizar que los servidores On-Premise cuenten con Python y todas las librerías necesarias.
* **Configuración del archivo YAML:** El archivo de configuración debe ser adaptado para conectarse a las bases de datos locales (por ejemplo, SQL Server, MySQL) o leer archivos locales (CSV, Excel).
* **Automatización:** En entornos On-Premise, se pueden usar tareas programadas de Windows o cron jobs en servidores Linux para ejecutar el análisis Beneficios:

**Beneficios**:

**Control total sobre los datos:** Todos los datos y procesos permanecen dentro de la infraestructura local del cliente, lo que garantiza mayor seguridad y cumplimiento normativo.

**Escalabilidad interna:** Si el cliente tiene capacidad de infraestructura, puede escalar el proceso para manejar grandes volúmenes de datos sin depender de la nube.

**Flexibilidad:** Se puede personalizar y adaptar según las necesidades del cliente y los recursos disponibles en su infraestructura.

## Implementación en Nube

La opción en la Nube permite ejecutar el análisis RFM utilizando los recursos escalables de proveedores como AWS, Azure o Google Cloud. Es ideal cuando el cliente busca reducir el mantenimiento de infraestructura local o cuando los datos son de gran volumen y necesitan procesamiento dinámica.

**Pasos para la Implementación**:

* **Instalación de Python**: El cliente puede usar entornos gestionados como AWS Lambda, Google Cloud Functions, Azure Databriks, etc, o configurar máquinas virtuales con Python preinstalado.
* **Configuración del archivo YAML**: El archivo de configuración se almacena en la nube (por ejemplo, en S3 o Google Cloud Storage) para ser accesible de manera remota.
* **Conexión a Bases de Datos en la Nube**: Si el cliente almacena los datos en bases de datos en la nube como Amazon RDS o Google Cloud SQL, el script de Python puede conectarse sin problemas.
* **Automatización**: En la nube, se pueden usar servicios como AWS CloudWatch, Google Cloud Scheduler o Azure Logic Apps para automatizar la ejecución de tareas en intervalos definidos.

**Beneficios**:

* **Escalabilidad**: La nube permite ajustar fácilmente los recursos de acuerdo con la volumetría de los datos y las necesidades del cliente.
* **Bajo mantenimiento**: No es necesario gestionar hardware ni infraestructura física, ya que el proveedor de la nube se encarga de todo.
* **Acceso remoto**: Los análisis y datos pueden ser accesibles desde cualquier ubicación y en cualquier momento.
  1. **Implementación Híbrida**

La opción Híbrida combina las ventajas de las implementaciones Local/On-Premise y Nube. Es adecuada para clientes que necesitan mantener ciertos datos en infraestructura interna, pero también aprovechar la escalabilidad y flexibilidad de la nube para procesar grandes volúmenes de datos.

A diagram of a computer program

Description automatically generated with medium confidence

**Pasos para la Implementación**:

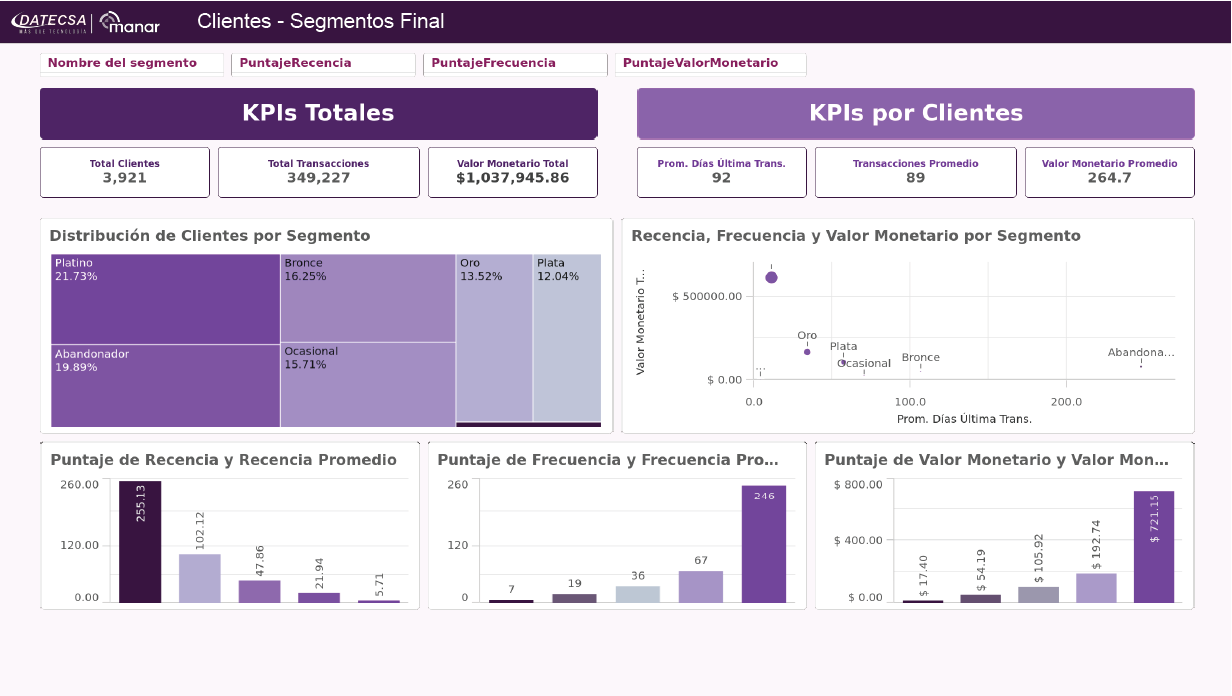
* **Instalación de Python**: El cliente puede instalar Python tanto en sus servidores locales como en la nube, según los componentes utilizados en el proceso.
* **Configuración del archivo YAML**: El archivo de configuración se ajusta para usar tanto datos locales como en la nube, dependiendo de las fuentes de datos y los recursos.
* **Conexión a Bases de Datos Mixtas**: El cliente puede mantener bases de datos locales y en la nube, y el script de Python se adapta para extraer los datos de ambas fuentes.
* **Automatización**: En este enfoque, se pueden usar herramientas de automatización combinadas, como AWS Batch o Google Cloud Pub/Sub junto con cron jobs locales.

**Beneficios**:

* **Flexibilidad y optimización de recursos**: El cliente puede utilizar la infraestructura interna para los datos sensibles y aprovechar la nube para el procesamiento intensivo de datos.
* **Escalabilidad mixta**: Se puede escalar el procesamiento de datos en la nube según sea necesario, manteniendo el control de datos clave dentro de la infraestructura local.

# Visualización e Integración con Herramientas de BI:

* Las salidas del análisis RFM (segmentos de clientes, puntuaciones y métricas) pueden exportarse en formatos estándar como CSV, Excel, Parquet o directamente a tablas en bases de datos locales o en la nube.
* Estas salidas pueden ser consumidas fácilmente por herramientas de análisis de negocio como Qlik, Power BI, Tableau o Looker. Por ejemplo, Configurar un conector directo a la base de datos en la nube (como Amazon Redshift o Azure SQL) o importar los datos desde un archivo exportado.
* Esta integración permite a los usuarios finales interactuar con los datos, realizar análisis detallados para generar y alimentar reportes dinámicos.



# Análisis Exploratorio

Antes de ejecutar completamente el proyecto RFM, es altamente recomendable que se realice un análisis exploratorio de los datos utilizando Notebooks de Python. Esto proporcionará una visión clara de la calidad de los datos y permitirá una configuración más precisa del archivo YAML, lo que ayudará a obtener mejores resultados durante el análisis.

**Beneficios del Análisis Exploratorio**:

* **Visualización de los Datos**: Los Notebooks permiten la visualización rápida de las variables, distribuciones y patrones en los datos, lo que facilita la comprensión de cómo afectan a las métricas de RFM (Recencia, Frecuencia, Monto).
* **Identificación de Anomalías y Outliers**: El análisis exploratorio puede ayudar a identificar anomalías o valores atípicos en los datos que podrían afectar el cálculo del RFM, permitiendo ajustarlos o limpiarlos antes de proceder con el análisis completo.
* **Configuración del archivo YAML**: Con base en los insights obtenidos del análisis exploratorio, se podrán ajustar los parámetros del archivo YAML, Esto permite personalizar el análisis a las características específicas de los datos.

**Herramientas Recomendadas**:

* **Jupyter Notebooks**: Una herramienta interactiva que facilita la ejecución paso a paso del análisis de datos, permitiendo hacer ajustes rápidos y observar los resultados de inmediato.
* **Pandas y Matplotlib/Seaborn / Plotly**: Librerías de Python ideales para la manipulación de datos y la creación de visualizaciones que ayuden en el análisis exploratorio.