****

**Formato Alcance de la solución F-P17-03 Ver. 01**

Demo RFM

**Transformación digital y Analítica**

**APROBACIONES**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Nombre:** |  | | **Empresa:** |  | | **Email:** |  | | **Fecha:** |  | | **Firma:** |  | |
|  |
|  |
| |  |  | | --- | --- | | **Nombre:** |  | | **Empresa:** |  | | **Email:** |  | | **Fecha:** |  | | **Firma:** |  | |

ÍNDICE DE CONTENIDO

[1. INFORMACIÓN SOBRE ESTE DOCUMENTO. 4](#_Toc184979908)

[2. ALCANCE. 5](#_Toc184979909)

[2.1. Entregables DATECSA 5](#_Toc184979910)

[2.2. Entregables Cliente 5](#_Toc184979911)

[2.3. Fuera del alcance de la presente solución 6](#_Toc184979912)

[3. CONTROL DE VERSIONES 7](#_Toc184979913)

[4. EQUIPO DE PROYECTO. 8](#_Toc184979914)

[5. GLOSARIO. 9](#_Toc184979915)

[6. DESCRIPCIÓN DEL PROCESO 11](#_Toc184979916)

[6.1. Estructura del Proyecto 12](#_Toc184979917)

[6.2. Detalles de los Módulos 12](#_Toc184979918)

[6.2.1. Configuración 12](#_Toc184979919)

[6.2.2. Ingestión de Datos 13](#_Toc184979920)

[6.2.3. Preprocesamiento 13](#_Toc184979921)

[6.2.4. Cálculo de Métricas RFM 13](#_Toc184979922)

[6.2.5. Asignación Score 13](#_Toc184979923)

[6.2.6. Generación de Segmento 13](#_Toc184979924)

[6.2.7. Exporte Resultados Finales 14](#_Toc184979925)

[6.3. Visualización 14](#_Toc184979926)

[6.3.1. Estructura Visualización Demo Qlik Sense: 14](#_Toc184979927)

[6.4. Escalabilidad del Sistema 15](#_Toc184979928)

[6.4.1. Arquitectura Local 15](#_Toc184979929)

[6.4.2. Arquitectura On-Premise 16](#_Toc184979930)

[6.4.3. Arquitectura en la Nube 16](#_Toc184979931)

[6.4.4. Arquitectura Híbrida 17](#_Toc184979932)

[7. ADAPTACIÓN DEL CÓDIGO A PYSPARK 18](#_Toc184979933)

[8. CONFIGURACIÓN DE DEPENDENCIAS 18](#_Toc184979934)

[9. PROCESO DE EJECUCIÓN DEL DEMO RFM LOCAL (PRUEBAS) 19](#_Toc184979935)

[9.1.1. Descargar el Repositorio 19](#_Toc184979936)

[9.1.2. Configuración del entorno 19](#_Toc184979937)

[9.1.3. Creación y/o Ajuste de Funcionalidades 19](#_Toc184979938)

[9.1.4. Ejecutar el Script Principal 19](#_Toc184979939)

[9.1.5. Integración con Herramientas de Visualización 19](#_Toc184979940)

[9.1.6. Revisión y Ajustes Continuos 20](#_Toc184979941)

[10. ANEXOS 21](#_Toc184979942)

[10.1. Repositorio 21](#_Toc184979943)

[10.2. Reporte Demo en Qlik Sense 21](#_Toc184979944)

[11. CONTROLES DE CAMBIO 22](#_Toc184979945)

# INFORMACIÓN SOBRE ESTE DOCUMENTO.

El DEMO RFM está diseñado como una herramienta adaptable para analizar y segmentar clientes en función de su Recencia, Frecuencia y Monto de transacciones. Este manual está orientado a profesionales técnicos, como científicos de datos, que necesiten modificar o extender el DEMO para ajustarlo a los requerimientos específicos de un cliente.

El proyecto está desarrollado en Python, pero contempla diferentes arquitecturas de implementación, como la integración con bases de datos o herramientas de visualización como Qlik. La modularidad del código permite personalizaciones rápidas y eficaces.

# ALCANCE.

* **Extracción de Datos:** Manejo de datos desde múltiples fuentes (CSV, Excel, SQL, Parquet).
* **Transformación:** Incluye manejo de nulos, eliminación de duplicados, conversión de tipos de datos, eliminación de valores negativos y manejo de outliers.
* **Cálculo Score RFM:** Cálculo de Recencia, Frecuencia y valor monetario, junto con puntuaciones basadas en cortes configurados (Percentiles o Jenks).
* **Segmentación:** Segmentación de clientes basada en puntajes RFM.
* **Visualización:** Generación de reportes y visualizaciones personalizadas según los parámetros configurados.

## Entregables DATECSA

* **Archivo de configuración YAML:** Debidamente parametrizado para análisis RFM adaptable.
* **Código Python:** Proyecto modular y reutilizable que ejecute el análisis RFM en base a la configuración.
* **Visualizaciones:** Dashboard RFM inicial con gráficos de distribución, segmentación y perfiles de clientes.
* **Manual de usuario:** Guía para actualizar el archivo YAML y ejecutar el código.

## Entregables Cliente

* ***Provisión de datos históricos:*** El análisis RFM requiere datos históricos para identificar patrones de comportamiento de los clientes. Elegir el período adecuado de cobertura es esencial para obtener insights representativos y alineados con las estrategias del negocio. La selección del rango de tiempo depende de factores como la frecuencia de compra, las metas comerciales y la naturaleza de la industria.
* ***Infraestructura de acceso:*** El cliente debe proporcionar **datos históricos** de transacciones en un formato **accesible y procesable**.
* **Archivos Locales:**
  + CSV, Excel, txt
  + Archivos bien estructurados con columnas estándar.
  + Ubicación accesible (servidores locales o unidades compartidas).
* **Bases de Datos Relacionales:**
  + MySQL, PostgreSQL, SQL Server.
  + Credenciales de acceso (usuario, contraseña, host, puerto).
  + Query de extracción definido (campos obligatorios: ID de cliente, fecha, monto).
* **Almacenamiento en la Nube:**
  + AWS S3, Azure Blob, Google Cloud Storage.
  + Credenciales de acceso (claves API, URL de bucket).
  + Estructura de carpetas y nombres de archivos clara.
* ***Definición de categorías de negocio****:* El cliente debe colaborar en la identificación de categorías clave para los resultados de RFM, tales como segmentos de clientes (Platino, Oro, Plata, etc.).
* **Ajustes a los parámetros de RFM:** El cliente podrá personalizar los límites de los percentiles, fechas, y otros parámetros según sus necesidades de negocio.

## Fuera del alcance de la presente solución

* **Migración de datos desde otras plataformas**: No se incluye migración de datos desde sistemas no compatibles con los formatos definidos en este manual.
* **Desarrollo de nuevos modelos de predicción**: El proyecto se centrará en la adaptación del modelo RFM, no incluye el desarrollo de modelos adicionales (como modelos de predicción avanzados).
* **Entrenamiento o formación del equipo del cliente**: No se incluye capacitación formal sobre ciencia de datos o uso avanzado del sistema RFM.

# CONTROL DE VERSIONES

| **Versión** | **Fecha del cambio** | **Razón del Cambio** | **Solicitado por (incluir área)** | **Resumen del cambio** |
| --- | --- | --- | --- | --- |
| 1 | 10/12/2024 | Versión Inicial |  | Versión Inicial |

# EQUIPO DE PROYECTO.

El desarrollo e implementación del RFM requiere un equipo multidisciplinario, compuesto por roles estratégicos y técnicos, para garantizar la calidad y eficacia del proyecto:

| **Empresa** | **Dependencia** | **Nombre** | **Rol en el proyecto** |
| --- | --- | --- | --- |
| Cliente |  |  | Responsable de Negocio |
| Cliente |  |  | Analista de Negocio |
| Cliente |  |  | Administrador de Datos |
| Cliente |  |  | Especialista Técnico |
| Datecsa |  |  | Líder de Proyecto |
| Datecsa |  |  | Científico de Datos |
| Datecsa |  |  | Ingeniero de Datos |
| Datecsa |  |  | Especialista en BI |
| Datecsa |  |  | Especialista en Infraestructura |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**NOTA:**

Dependiendo la complejidad del proyecto, podría ser necesario un **Ingeniero de Datos** para la implementación.

Los miembros del equipo de proyecto pueden cambiar una vez iniciada la etapa de implementación.

# GLOSARIO.

| **Palabra** | **Descripción** |
| --- | --- |
| **RFM** | **(Recencia, Frecuencia, Monto):** Modelo analítico utilizado para segmentar clientes según el tiempo desde su última compra (Recencia), la cantidad de compras realizadas (Frecuencia) y el monto total gastado (Monto). |
| **Recencia** | Mide el tiempo transcurrido desde la última transacción del cliente, proporcionando información sobre su nivel de interés reciente. |
| **Frecuencia** | Indica la cantidad de interacciones o compras realizadas por un cliente en un período determinado, |
| **Valor Monetario** | Representa el valor económico total que un cliente ha generado en sus transacciones, ayudando a identificar clientes de alto valor. |
| **Percentil** | Método estadístico utilizado para calcular puntos de corte en las métricas RFM, dividiendo los datos en segmentos basados en proporciones. |
| **Método Jenks** | Técnica de clasificación que minimiza la varianza dentro de los grupos y maximiza la diferencia entre ellos, utilizada para definir puntos de corte en el análisis RFM. |
| **YALM** | Archivo de texto estructurado que permite definir parámetros clave para el demo, como categorías, periodos de análisis, el tipo y las rutas de fuentes de datos, etc. |
| **Outliers** | Valores atípicos en los datos que pueden influir negativamente en el análisis. El demo incluye métodos para identificarlos y manejarlos según configuraciones predefinidas. |
| **Breaks (Puntos de Corte)** | Límites que dividen las métricas RFM en rangos, utilizados para calcular puntuaciones y segmentar clientes. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# DESCRIPCIÓN DEL PROCESO

El *Demo RFM* es una solución analítica modular y flexible diseñada para calcular y segmentar clientes según las métricas de **Recencia**, **Frecuencia** y **Monto** (RFM). Permite identificar patrones clave de comportamiento del cliente y asignar categorías de negocio personalizables mediante configuraciones en YAML. El demo incluye funcionalidades como manejo de outliers, cálculo dinámico de puntos de corte y generación de reportes visuales interactivos en Qlik.

Flujo del Proceso

A screenshot of a diagram

Description automatically generated

* 1. **Captura de Datos:** El cliente proporciona sus datos de transacciones de clientes, que deben incluir detalles como la fecha de compra, el monto de cada transacción y la cantidad de compras realizadas por cada cliente.
  2. **Preprocesamiento:** Se limpia y transforma la información para adaptarla al modelo RFM. Esto puede implicar la eliminación de registros duplicados, el tratamiento de valores nulos, etc.
  3. **Cálculo de RFM:** Se calculan las métricas de Recencia, Frecuencia y Valor Monetario basadas en las transacciones de los clientes.
  4. **Segmentación de Clientes:** Los clientes se segmentan en categorías específicas basadas en sus puntuaciones RFM.
  5. **Visualización:** Se genera un informe que detalla las categorías asignadas a cada cliente y su perfil de comportamiento.

## Estructura del Proyecto

La estructura del proyecto está organizada de manera modular, permitiendo una fácil personalización y extensión según las necesidades del cliente. La estructura básica es la siguiente:

A screenshot of a computer program

Description automatically generated

## Detalles de los Módulos

### Configuración

El archivo configuracion.yaml centraliza los parámetros clave que controlan el comportamiento del cálculo de RFM, desde las rutas donde debe consultarse la data para la ejecución del modelo, definición de ventana de tiempo (historia) para el cálculo, categorías de segmentación, entre otros.

A screenshot of a computer

Description automatically generated

### Ingestión de Datos

* **Modulo:** data\_loader.py
* **Descripción:** Este módulo es responsable de la carga de los datos desde diversas fuentes (CSV, SQL, etc.).
* **Personalización:**
  + Modificar configuracion.yaml para definir la fuente de datos.
  + Añadir nuevas funciones para formatos no soportados.

### Preprocesamiento

* **Modulo:** preprocessing.py
* **Descripción:** Este módulo realiza las tareas de limpieza y transformación de los datos.
* **Personalización:**
  + Modificar configuracion.yaml para definir tablas y campos a los cuales aplicar los pasos de preprocesamiento.
  + Añadir nuevas funciones o ajustar las existentes para preprocesamiento de datos específicos del cliente.

### Cálculo de Métricas RFM

* **Modulo:** rfm\_calculator.py
* **Descripción:** Calculo de las variables Recencia, Frecuencia y Monto.
* **Personalización:** 
  + Modificar configuracion.yaml los nombres de los campos a tomar para el cálculo de Recencia, Frecuencia y Monto.
  + Añadir nuevas funciones o ajustar las existentes para el cálculo de estas variables de acuerdo con los requerimientos del cliente.

### Asignación Score

* **Modulo:** rfm\_processing.py
* **Descripción:** Manejo de outliers, definición de puntos de corte y asignación de Score RFM.
* **Personalización:** Las configuraciones para el manejo de outliers y métodos de cálculo de puntos de corte son personalizables mediante el archivo YAML.

### Generación de Segmento

* **Modulo:**
* **Descripción:** Este módulo realiza la asignación del segmento final a cada cliente en función de las puntuaciones RFM calculadas previamente.
* **Personalización:** Utiliza las reglas y categorías definidas en el archivo de configuración YAML para clasificar a los clientes en segmentos de negocio como 'Platino', 'Oro', 'Plata', entre otros.

### Exporte Resultados Finales

* **Modulo**: exporter.py
* **Descripción:** Los resultados finales (incluidas las categorías de clientes) se exportan en el formato especificado en el archivo de configuración para facilitar su integración con otros sistemas o su visualización en plataformas como Qlik.
* **Personalización:** 
  + Modificar configuracion.yaml para definir ruta y formato para el exporte de resultados.

## Visualización

El proceso del DEMO RFM en Python realiza el cálculo de las métricas (Recency, Frequency, Monetary), la asignación de segmentos, y la exportación de los resultados en un formato compatible con herramientas de visualización como Power BI, Qlik Sense u otras. Estas herramientas permiten crear tableros interactivos para analizar los resultados y facilitar la toma de decisiones.

RFM genera un archivo consolidado con los datos segmentados, listo para ser utilizado en una herramienta de visualización. Las características del archivo son las siguientes:

* **Formato:**
  + CSV, Excel, o Parquet (según se configure en el módulo exporter.py).
* **Estructura del Archivo Exportado**: El archivo debe contener como mínimo las siguientes columnas:
  + 1. **CustomerID**: Identificador único de cada cliente.
    2. **Recency**: Días desde la última compra.
    3. **Frequency**: Número de compras realizadas en el período definido.
    4. **Monetary**: Total gastado por el cliente en el período definido.
    5. **RFM\_Score:** Puntuación compuesta (por ejemplo, 555, 123).
    6. **RFM**\_**Segment**: Segmento asignado al cliente (Platino, Oro, Bronce, etc.)
* **Ubicación del Archivo Exportado:** El archivo se genera en la ruta definida en el archivo de configuración (exporter.py).
* **Entregables de Visualización**

Como parte del DEMO RFM, se entregan los siguientes elementos relacionados con la visualización:

* + **Archivo Consolidado:** El archivo exportado listo para cargar en herramientas de visualización.
  + **Ejemplo de Tablero:** Dashboard en Qlik Sense preconfigurado.

### Estructura Visualización Demo Qlik Sense:

El *Demo RFM* incluye un dashboard interactivo en Qlik Sense, diseñado para facilitar la interpretación y exploración de los resultados obtenidos a partir del análisis RFM.

La visualización se desarrolla trabajando desde dos capas principales:

1. **Capa de Extracción de Datos:** Centraliza, limpia y prepara la información.
2. **Capa de Negocio:** Integra el modelo de datos utilizando un enfoque dimensional que permite construir visualizaciones dinámicas y personalizadas. Esta capa toma como insumo principal la segmentación RFM generada por el proyecto en Python, ofreciendo insights accionables para la toma de decisiones estratégicas.

## Escalabilidad del Sistema

La arquitectura para implementar el análisis RFM dependerá de las necesidades y capacidades del cliente. Como el DEMO RFM está diseñado para ser **preconfigurable y adaptable**, puede ajustarse a distintos escenarios, desde pequeños volúmenes de datos procesados localmente hasta grandes volúmenes en arquitecturas distribuidas. La decisión final depende de tres factores principales:

* Recursos y capacidades técnicas del cliente.
* Infraestructura tecnológica ya existente.
* Volumetría de datos que se manejará en el análisis RFM.

### Arquitectura Local

A diagram of a computer program

Description automatically generated with medium confidence

* **Descripción:**
  + Ideal para volúmenes de datos pequeños o moderados.
  + Todo el procesamiento se realiza en una computadora o servidor personal del cliente.
* **Flujo de trabajo:**
  + - **Fuentes de datos:** Archivos como CSV, Excel o Parquet.
    - **Transformación y procesamiento:** Utilizando scripts en Python ejecutados en el entorno local.
    - **Almacenamiento:** Resultados almacenados en bases de datos locales o directamente en archivos.
    - **Visualización:** Integración con herramientas como Qlik o Power BI.
* **Ventajas:**
  + Baja complejidad técnica.
  + Menor costo inicial.
* **Limitaciones:**
  + No escalable para grandes volúmenes de datos.
  + Dependencia de recursos locales del cliente (capacidad de memoria y CPU).

### Arquitectura On-Premise

A diagram of a computer program

Description automatically generated with medium confidence

* **Descripción:**
  + Diseñada para clientes con infraestructura propia dentro de un data center o red empresarial.
* **Flujo de trabajo:**
  + **Fuentes de datos:** Bases de datos empresariales (PostgreSQL, Oracle, etc.) o archivos gestionados internamente.
  + **Transformación y procesamiento:** Ejecución del modelo en servidores locales mediante Python o adaptaciones en PySpark para procesamiento paralelo si el volumen lo requiere.
  + **Almacenamiento:** En sistemas locales controlados por el cliente.
  + **Visualización:** Dashboards generados en herramientas como Qlik o PowerBI conectadas a la infraestructura.
* **Ventajas:**
  + Control total sobre los datos.
  + Adecuado para empresas con políticas estrictas de seguridad.
* **Limitaciones:**
  + Requiere inversión en hardware y mantenimiento de la infraestructura.

### Arquitectura en la Nube

A diagram of a computer program

Description automatically generated with medium confidence

* **Descripción:**
  + Orientada a clientes que manejan grandes volúmenes de datos o requieren escalabilidad automática.
* **Flujo de trabajo:**
  + **Fuentes de datos:** Sistemas en la nube como Amazon S3, Azure Data Lake o Google Cloud Storage.
  + **Transformación y procesamiento:** Uso de servicios escalables como AWS Lambda, Databricks o Google DataProc, ejecutando scripts en Python o adaptaciones en PySpark para procesamiento distribuido.
  + **Almacenamiento:** Bases de datos en la nube o formatos como Parquet para optimizar consultas.
  + **Visualización:** Dashboards en herramientas como Power BI o Qlik conectadas directamente a los datos procesados.
* **Ventajas:**
  + Escalabilidad dinámica.
  + Reducción de costos iniciales en infraestructura física.
* **Limitaciones:**
  + Dependencia de servicios en la nube.
  + Costo recurrente dependiendo del uso.

### Arquitectura Híbrida

* **Descripción:**
  + Combina elementos de infraestructura local y en la nube, ofreciendo flexibilidad para diferentes escenarios.
* **Flujo de trabajo:**
  + **Fuentes de datos:** Datos locales sincronizados con servicios en la nube.
  + **Transformación y procesamiento:** Parte del modelo puede ejecutarse en servidores locales (Python) y parte en servicios distribuidos (PySpark) para Big Data.
  + **Almacenamiento:** Mixto, con bases de datos locales y almacenamiento en la nube para optimizar costos.
  + **Visualización:** Dashboards integrados con múltiples fuentes.
* **Ventajas:**
  + Flexibilidad para ajustar el procesamiento según las necesidades.
  + Optimización de costos y tiempo.

## ADAPTACIÓN DEL CÓDIGO A PYSPARK

Cuando el cliente maneja grandes volúmenes de datos, el modelo puede adaptarse a **PySpark** para aprovechar procesamiento paralelo en clústeres de datos. Esto implica:

* Migrar las operaciones de transformación y cálculo a funciones distribuidas en PySpark.
* Utilizar APIs de PySpark para manipular DataFrames y optimizar el tiempo de ejecución.
* Configurar la infraestructura para gestionar tareas distribuidas en entornos como Databricks o Hadoop.

## CONFIGURACIÓN DE DEPENDENCIAS

El modelo RFM utiliza varias librerías de Python para realizar el procesamiento, análisis, y exportación de datos.

**Python 3.8** o superior instalado.

**Archivo requirements.txt**

El archivo requirements.txt contiene todas las librerías necesarias para ejecutar el DEMO. Estas librerías son las siguientes:

* Numpy
* Pandas
* Jenkspy
* PyYAML
* SQLAlchemy

**Nota importante:**

Si se modifican o adicionan funcionalidades al código que requieran de otras librerías, estas deben ser incluidas en el archivo **requirements.txt** para asegurar la correcta ejecución del DEMO. Esto permite que todos los usuarios y entornos cuenten con las mismas dependencias y se eviten conflictos de versiones.

## PROCESO DE EJECUCIÓN DEL DEMO RFM LOCAL (PRUEBAS)

Esta sección describe los pasos necesarios para ejecutar el DEMO RFM y obtener los resultados. Garantizar haber configurado previamente las dependencias y de contar con los datos necesarios según lo indicado en las secciones anteriores.

### Descargar el Repositorio

Para comenzar la implementación, lo primero es obtener el código fuente, este se encuentra alojado en un repositorio de GitHub.

### Configuración del entorno

1. Verificar que todas las dependencias estén instaladas. Archivo **requirements.txt**
2. Garantizar que el archivo de configuración en formato YAML esté correctamente configurado con los parámetros del análisis.

El archivo de configuración debe ser personalizado según las necesidades del cliente. Este archivo contiene configuraciones clave, como las categorías de negocio, los percentiles para el cálculo de las puntuaciones y otros parámetros relacionados con el cálculo de RFM.

**Nota:** Si esta es la primera vez que se ejecuta Demo RFM, se recomienda realizar un análisis exploratorio de los datos mediante el módulo **Notebook** incluido en el proyecto. Esto permite identificar características importantes de la data y ajustar los parámetros en el archivo de configuración antes de ejecutar el modelo

1. Contar con los datos transaccionales en el formato adecuado (CSV, Excel, Parquet, o conectado a una base de datos).

### Creación y/o Ajuste de Funcionalidades

Una vez que se haya descargado el código y configurado el archivo YAML, se puede comenzar a adaptar el cálculo de RFM según las necesidades específicas del cliente. Es importante tener en cuenta que, de acuerdo con los requerimientos del cliente, puede ser necesario **añadir o ajustar funcionalidades** del código fuente original para cumplir con la personalización requerida.

### Ejecutar el Script Principal

Una vez realizados los ajustes, se puede ejecutar el script principal (main.py). Este script orquesta todos los módulos y genera los resultados finales del RFM.

### Integración con Herramientas de Visualización

El modelo RFM generará resultados en un formato estructurado. Estos archivos contienen las puntuaciones de RFM de los clientes junto con sus categorías. Este formato es ideal para ser importado a herramientas de visualización como Power BI o Qlik Sense.

### Revisión y Ajustes Continuos

Hay que recordar que el cálculo de RFM puede requerir ajustes periódicos a medida que los datos cambian. Puede modificar las configuraciones en el archivo YAML o ajustar los módulos de procesamiento de acuerdo con los nuevos requisitos del cliente.A

# ANEXOS

## Repositorio

El código fuente del modelo RFM está disponible en el siguiente repositorio:

[Repositorio RFM](https://github.com/Datecsa-ArquitecturaDatos/CienciaDatos-RFM)

Es importante destacar que el código fuente ha sido **debidamente comentado** para facilitar su comprensión y adaptación. Cada módulo y función tiene comentarios explicativos que describen su propósito, los parámetros utilizados, los procesos que se realizan y cómo pueden personalizarse según los requisitos del cliente. Esto permitirá que cualquier persona encargada de implementar o ajustar el modelo pueda hacerlo de manera eficiente,

## Reporte Demo en Qlik Sense

El reporte de demo realizado en Qlik Sense que visualiza los resultados del análisis RFM está disponible en el siguiente enlace:

Reporte en Qlik Sense:

[ReporteRFM](https://datecsaint.us.qlikcloud.com/sense/app/3953ad5e-897d-4352-a1ca-67ab1a9c8492/sheet/e40d258b-8f6a-4f36-a31f-f8743933a17e/state/analysis/hubUrl/%2Fanalytics%2Fhome)

# CONTROLES DE CAMBIO

A continuación, se relacionan los controles de cambio que surgieron durante el proyecto y fueron aprobados:

No aplica a la primera versión de este documento