

Documentacion Proyecto Multidestinos

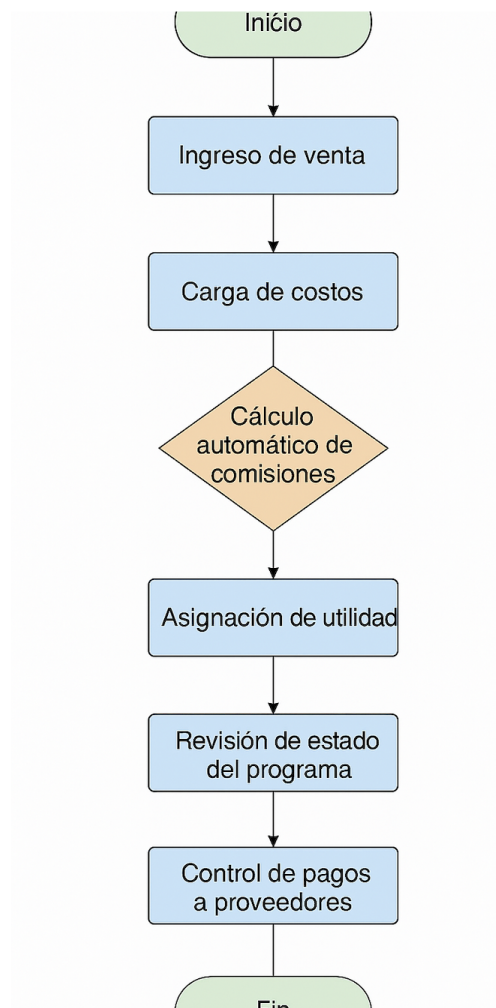
yaml

1. Descripción del proyecto

Este proyecto consiste en el desarrollo de una solución automatizada para la agencia de viajes *Multidestinos*. El objetivo es gestionar el control de ventas, cálculo de comisiones, y visualización de utilidades mediante una aplicación desarrollada en Shiny. Esta app será alimentada con información que los mismos ejecutivos registrarán de forma directa, evitando el uso de hojas de cálculo separadas.

2. Diagrama de flujo

A continuación se presenta el flujo de trabajo del sistema:



3. Documentación del desarrollo

Esta sección detalla el trabajo realizado, desde la carga de los datos hasta la preparación del producto mínimo viable (MVP).

3.1. Carga y exploración de los datos

Se utilizó un archivo `.xlsx` que contiene información de las ventas realizadas por los ejecutivos. Se cargó con el paquete `readxl`:

```
library(readxl)
ventas <- read_excel("ventas_expa.xlsx")
```

3.2. Limpieza y transformación

Se eliminaron los símbolos \$ y puntos de miles del archivo Excel original para facilitar el análisis. Además, se aplicaron transformaciones con `dplyr`:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
ventas <- ventas %>%
  mutate(
    Comision_total = total_pago_cliente - (costo_hotel_netto + costo_vuelo_netto + costo_seguro_netto + costo_otros_netto),
    Comision_ejecutivo = Comision_total * 0.5,
    Comision_agencia = Comision_total * 0.5,
    Utilidad = Comision_agencia,
    Estado_programa = "Pendiente"
  )
```

Esta transformación permite calcular la utilidad y las comisiones de forma automatizada, minimizando errores.

3.3. Herramientas de documentación y visualización

Para documentar el desarrollo del proyecto, se integraron herramientas propias del entorno de RStudio junto con recursos visuales que permiten representar de manera clara el proceso automatizado implementado.

Se utilizó `RMarkdown` para generar este informe reproducible, facilitando la trazabilidad del código, las transformaciones de datos y los resultados obtenidos. Además, se incorporó un diagrama de flujo creado con una herramienta externa y embebido previamente en el documento.

También se incluyó una Model Card adaptada al contexto de Vetiver, que resume los detalles del modelo de negocio, su uso, los datos procesados y consideraciones éticas:

```
model_card:
  model_details:
    developer: "Alejandra Robles"
    version: "MVP v1"
    description: "Este proyecto automatiza el cálculo de comisiones y utilidades para la agencia Multid"
  intended_use:
    primary_uses:
      - "Control de comisiones"
      - "Gestión de utilidad por venta"
    primary_users:
      - "Ejecutivos de ventas"
      - "Encargados financieros"
  data:
    input_data:
      - "Archivo Excel con registros de ventas"
    data_cleaning:
      - "Remoción de símbolos en Excel"
      - "Conversión a variables numéricas en R"
  metrics:
    - "Cálculo automatizado de utilidad y comisiones"
  ethical_considerations:
    - "Privacidad de datos de clientes"
  caveats_recommendations:
    - "La app aún está en fase MVP"
```

5. Despliegue

La solución será llevada a producción mediante el despliegue de una aplicación web construida con **Shiny**, alojada inicialmente en Shinyapps.io, un servicio que permite publicar aplicaciones desarrolladas en R sin requerir servidores propios.

El proceso de despliegue contempla los siguientes pasos:

- **Preparación del producto mínimo viable (MVP):** Validación local de que la aplicación funcione correctamente y que los cálculos sean consistentes.
- **Versionamiento con Git y GitHub:** El código fuente y la documentación se mantienen organizados y rastreados en un repositorio, lo cual facilita futuras actualizaciones.
- **Publicación en Shinyapps.io:** Se utilizará la función `rsconnect::deployApp()` desde RStudio para subir la app. Se podrá limitar el acceso a usuarios autorizados.
- **Escalabilidad futura:** En función del crecimiento del negocio, se evaluará migrar a un servidor propio o usar plataformas como RStudio Connect para una gestión más robusta.

6. Monitoreo

Aunque el proyecto no incluye aún un modelo de machine learning, se implementarán mecanismos de monitoreo para asegurar el funcionamiento estable de la app:

- **Validación de entradas:** Se controlará que los datos ingresados por los ejecutivos cumplan con los formatos esperados (valores numéricos, sin valores faltantes).

- **Indicadores de control:** La aplicación mostrará alertas o validaciones internas si se detectan inconsistencias, como ventas con utilidad negativa.
- **Registros de uso (logs):** Se mantendrán logs básicos de interacción para detectar errores o mal uso.
- **Feedback de usuarios:** Durante la etapa MVP se recogerán comentarios de los usuarios clave para mejorar la aplicación.
- **Actualización continua:** Los ajustes se versionarán mediante Git y cada nueva versión se desplegará de forma ordenada.

En futuras etapas, si se incorporan modelos predictivos, se podrá integrar **Vetiver** para gestionar el versionado, validación y monitoreo de modelos en producción.