

## TECHNICAL TEST TO MARTECH POSITION:

1. Restore an AdventureWorks database (indicate the version used and the final name of the restored DB, ex: AdventureWorks\_v2019)
2. Create a new destination database AdHocDB
  - In the destination database, create the tables:
    - tbCustomers with at least: CustomerId (PK), FirstName, LastName, Phone, Email, CreatedAt.
    - tbProduct with at least: ProductId (PK), Name, CurrentPrice (from LastPrice), CategoryName, CreatedAt.
    - tbSalesSummary with at least: SalesSummaryId (PK identity), CustomerId, CustomerName, SummaryDate, Total\_Items, Total\_Sales, Source ('SYSTEM'/'MANUAL'), CreatedAt.
3. Create 3 SSIS packages:
  - Create SSIS Package A:
    - Load Customers into tbCustomers:  
Fields: CustomerId, names, phone, email (use the related tables as needed).
    - Validation: the load must be incremental (only insert new records; do not duplicate).
    - The “new” criteria must be based on CustomerId (not existing in destination)
  - Create SSIS Package B:
    - Load Products into tbProduct:
    - Fields: ProductId, Name, CurrentPrice (from LastPrice), CategoryName (use the related tables as needed).
    - Validation: the load must be incremental (only insert new records; do not duplicate).
    - The “new” criteria must be based on ProductId (not existing in destination).
  - Create SSIS Package C:
    - Load the Sales tables (Header and Detail) and, based on them, generate the sales summary into tbSalesSummary:
    - Group by CustomerId and SummaryDate (date without time).
    - Calculate: Total\_Items\_Quantity and Total\_Sales\_Amount
    - Insert with Source='SYSTEM'.
    - Mandatory validation: the load must be incremental (do not duplicate).
- ✓ Mandatory use of SQL objects in the destination database: Views, UDFs and Stored Procedures
- ✓ These objects must be used in the SSIS and/or report/API queries, not only created.
4. Create an SSIS package that consumes an endpoint from the StarWars API (<https://swapi.dev>) and writes the data into a table in the AdHocDB database.
5. Create an SSRS report that consumes tbSalesSummary (or a view based on that table):
  - Mandatory parameters: InitialDate and FinalDate.
  - Must filter correctly by date range.

6. Create a .NET REST API connected to the destination database with 2 endpoints:
- GET /api/sales\_summary... Parameter: idCustomer. Returns the summary records for that customer.
  - POST /api/manual\_summary\_entry: Inserts a manual record into tbSalesSummary with Source='MANUAL' (payload with the required fields)  
Include basic validations (required fields, non-negative values, valid date)  
Error handling (400/500).

**Deliverables:**

- SSIS, SSRS and API code in a personal GitHub repository (shared link).
- Evidence in screenshots (minimum):
- AdventureWorks restored.
- Structure created in the destination database (tables/objects).
- Execution of SSIS Packages A, B and C (including re-run proving it does not duplicate).
- Resulting data in tbCustomers, tbProduct, tbSalesSummary.
- Execution of the SSRS report with parameters.
- API tests (GET and POST) working (Swagger/Postman).