**Unit Testing Approaches Summary**

**Branch Name :** foundation/siva.saravana/257964\_unit\_test

**Clone Url :** https://github.com/Advantage-Solutions/dnap-ingestion-framework.git

**Overview of Testing Approaches**

As part of our user story implementation, we employed four distinct unit testing methodologies to ensure comprehensive test coverage:

1. **Notebook API-based Testing**
2. **Mock-based Testing**
3. **Pydantic-based Testing**
4. **Pytest-based Testing**

*Note: The API-based unit testing code provided represents our intended approach but hasn't been fully tested in production.*

**Test Coverage by Approach**

**1. Notebook API-based Testing**

**Type**: Integration/System Testing  
 **Coverage**:

* Notebook execution via Databricks API
* Cluster state validation before execution
* Parameter passing validation
* Notebook output validation
* Data validation against expected schema and values
* Watermark value verification
* Data comparison between expected and actual results

**2. Mock-based Testing**

**Type**: Unit Testing with Isolation  
 **Coverage**:

* Source datasource type validation
* Catalog name resolution
* File location resolution
* Error handling for invalid inputs
* Dynamic file path generation
* Environment-specific configuration handling

**3. Pydantic-based Testing**

**Type**: Data Validation Testing  
 **Coverage**:

* Configuration file loading and validation
* Dynamic file path resolution
* Watermark value retrieval
* Source query construction
* DataFrame schema validation
* Data type validation (strings, integers, decimals, timestamps)
* Null value handling
* Schema evolution testing
* CDC (Change Data Capture) operation validation
* ETL column addition verification

**4. Pytest-based Testing**

**Type**: Unit/Integration Testing  
 **Coverage**:

* Configuration file discovery
* Pipeline configuration loading
* Watermark value retrieval
* Source query generation
* DataFrame retrieval from source
* Schema creation from transformation maps
* Delta table existence verification
* Data comparison assertions