

CDI Bonus Assignment

Objectives

We are seeking to create a data product that calculates the CDI Bonus (a very short-term bond issued by banks in Brazil). Users' wallet balances will generate daily interest, which will be deposited back into their wallets (transactional production database). Here are the core requirements:

- **Wallet History:**
 - We provide raw "change data capture" files with updates to the users' wallets over time.
 - Part of the objective is to create an intermediate table history table to identify wallet balances over time.
- **Interest Calculation:**
 - Users will earn interest on their cash balances above \$100.
 - Users will earn interest on the balance in their wallet that hasn't been moved for at least 24 hours.
- **Interest Rate:** The interest rate can change daily.
- **Time Frame:** The interest is calculated based on a day that begins at 00:00 and ends at 23:59.
- **Daily Payout:** Interest should be paid out daily to all users who qualify.

Although this is a simplified project, please approach it as you would a real production environment.

Non-functional requirements

This service is mission-critical. If this service doesn't work, we're breaching our commitment to our users, and this provides a really bad experience to end users. Visibility into the process is crucial: if something breaks, if something could be wrong in the source data, and clear documentation of the logic used.

Technical Requirements

We want to evaluate Spark skills, so the main CDI Bonus calculation should use Spark to read/write data files. Beyond that, technical choices are at your discretion: but make sure they're appropriate for the context (handling money and creating new transactions).

Deliverables

A GitHub repository containing:

- The implementation of the data product.
- Instructions on how to install any dependencies, test, and run the service
- Recommendations to ingest this data into a production transactional database.
- A written explanation of the design choices, detailing how the implementation meets both functional and non-functional requirements.
- A written explanation of any compromises or trade-offs made due to time constraints.

All documentation should be in English.