**Use Case — Idempotent Operations Handling (UC-SEG-003)**

**Acceptance Criteria**

* **AC-SEG-003-01 (First request — success)**
  + **Given** a valid IdOperacao and a valid payload
  + **When** I send a POST to a financial operation (deposit, withdrawal, transfer)
  + **Then** the system processes the transaction, persists it atomically, stores the result in IdempotencyStore with Status=Succeeded, and returns 200/201.
* **AC-SEG-003-02 (Replay with identical payload)**
  + **Given** the same IdOperacao and identical payload already processed
  + **When** I resend the request
  + **Then** the system returns the exact same response (200/201) without creating a new transaction.
* **AC-SEG-003-03 (Replay with conflicting payload)**
  + **Given** the same IdOperacao but with a different payload (e.g., different value or target account)
  + **When** I resend the request
  + **Then** the system returns 409 Conflict with error CONFLITO\_IDEMPOTENCIA.
* **AC-SEG-003-04 (Failure between posting and saving in store)**
  + **Given** the financial transaction was committed but the IdempotencyStore failed to update
  + **When** I resend the same IdOperacao
  + **Then** the system detects the existing movement via (ContaId, IdOperacao) and reconstructs the original response (200/201).
* **AC-SEG-003-05 (Missing header)**
  + **Given** a POST request for a financial operation without IdOperacao
  + **When** I send the request
  + **Then** the system returns 400 Bad Request with error IDOPERACAO\_OBRIGATORIO.
* **AC-SEG-003-06 (Invalid UUID format)**
  + **Given** an invalid UUID is sent in the IdOperacao header
  + **When** I send the request
  + **Then** the system returns 422 Unprocessable Entity with error IDOPERACAO\_INVALIDO.
* **AC-SEG-003-07 (Idempotency store unavailable)**
  + **Given** the IdempotencyStore is temporarily unavailable
  + **When** I send a financial operation request
  + **Then** the system falls back to the “double barrier” (unique constraint on ContaId + IdOperacao), processes the transaction safely, and attempts to persist best-effort response; if it fails completely, return 503 IDEMPOTENCIA\_INDISPONIVEL.
* **AC-SEG-003-08 (Persistence failure)**
  + **Given** a persistence error occurs after validating idempotency but before storing the transaction
  + **When** I send the request
  + **Then** the system returns 500 Internal Server Error with error ERRO\_PERSISTENCIA while ensuring no duplicate transaction is created (thanks to the unique index).