WORKSHOP I

SOFIA LOZANO MARTINEZ
20211020088

JOSÉ JESÚS CÉSPEDES RIVERA
20211020118

CARLOS ANDRÉS SIERRA VIRGUEZ

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

FRANCISCO JOSÉ DE CALDAS DISTRICT UNIVERSITY

FACULTY OF ENGINEERING
CURRICULAR PROJECT: SYSTEMS ENGINEERING
DATABASE II

BOGOTÁ, D.C., MAY 13
2025

Key Partners

- Bancolombia: Banking entity that supports and backs Nequi's operations.
- FGA Fondo de Garantías S.A.: Entity that provides guarantees to secure loans granted to users.
- Visa: Partner in issuing digital and physical cards.
- Payoneer: Platform that, in partnership with Nequi, allows users to receive payments in dollars and euros.
- Redeban: Company that facilitates technology for free and instant transfers.
- Merchants and marketplaces: Businesses that accept payments through Nequi.

Key Activities

- Application development and maintenance.
- Security management and fraud prevention.
- Digital marketing and strategic partnerships.
- Customer service and support.
- Innovation in new financial services.

Key Resources

- Digital platform and cloud servers.
- Security and authentication technology.
- Partnerships with banks and merchants.
- Customer database.
- Development and support team.

Value Proportions

- Digital account with no maintenance fees.
- Ease of transfers and payments without the need for a card.
- Integration with multiple services (top-ups, bill payments, insurance purchases, etc.).
- 100% digital experience without physical branches.
- Security through biometric authentication and real-time notifications.
- Credit products.

Customer Relationships

- Self-service through the application.
- Chatbot and online assistance.
- Community on social media and forums.
- Promotions and benefits for frequent use.

Channels

- Mobile application (Android and iOS).
- Official website.
- Social media (Facebook, Instagram, Twitter).
- Integrations with merchants and payment platforms.
- Customer service through in-app chat.

Customer Segments

- Young and digital adults (primarily aged 18-35).
- Unbanked or underbanked individuals.
- Small entrepreneurs and freelancers.
- Users seeking agile financial services without high costs.

Cost Structure

- Platform development and maintenance.
- Technological infrastructure (servers, cloud, security).
- Transaction costs and banking commissions.
- Advertising and digital marketing.
- Customer support and service costs.

Revenue Streams

- Commissions for ATM withdrawals.
- Interest generated from financial products.
- Commissions for additional services (insurance, credits, top-ups).
- Monetization of user data (consumption trends, without compromising privacy).
- Strategic partnerships with merchants.

Designed For:

- **Jose Jesus Cespedes Rivera**
- **Sofia Lozano Martinez**

Date:
17/03/2025

NEQUI

Business Model Canvas

REQUIREMENTS

Functional Requirements

1. User Registration
   - The system must store the user's basic information (name, ID number, email, mobile number, etc.).
   - An account associated with the user must be registered with a unique identifier and status (active/inactive).
2. Account and Balance Management
   - The system must register multiple accounts associated with the same user (savings accounts, pockets, etc.).
   - The available balance of each account must be updated after every transaction.
3. Transaction Management
   - The system must register transactions of income type (top-ups, received transfers) and expense type (withdrawals, payments, sent transfers).
   - The transaction history must be stored with the following fields: ID, date and time, type, amount, origin, destination, status.
4. Funds Validation
   - Before registering an expense transaction, the system must verify that the account has sufficient funds
5. Transfer Control
   - The system must store transfers between users (internal) and to other banks (external), with a status (pending, successful, failed).
6. Data Audit
   - An audit log must be maintained for sensitive changes: modifications to personal data, account deletions, transaction reversals.
7. Support for Multiple Devices/Sessions
   - The system must record from which device and approximate location the transactions originate for further analysis.

Non-Functional Requirements

1. Performance
   - Read and write operations (check balance, register transaction) must be performed in less than 1 second in 95% of the cases.

2. Scalability
   - The system must be able to handle millions of users and concurrent transactions, thus it must support horizontal and vertical partitioning of databases.

3. Availability
   - The system must be available 24/7 and must be designed with data backup (replication) and failure recovery.

4. Security
   - All sensitive information must be stored encrypted (e.g., hashed passwords, access tokens).
   - Transactions must include associated authentication (e.g., temporary token or session ID).
   - There must be role-based access control (user, system administrator, auditor, etc.).

5. Data Integrity
   - Transactions must comply with atomicity, consistency, isolation, and durability (ACID).
   - No transaction should be allowed to remain in an intermediate state.

6. Audit and Traceability
   - It must be possible to reconstruct the complete history of each user's operations for legal or security purposes.

7. Maintainability
   - The database structure must be normalized and documented to facilitate understanding and maintenance.

USER STORIES

US-01

Title: Account registration and creation	**Priority**: High	**Estimate (1-5)**: 2 points
User Story: As a user, I want to register and create a Nequi account so that I can manage my finances without complications.		
Acceptance Criteria: Given the user downloads the Nequi app, When they complete the registration form and accept the terms, Then the system creates an account and notifies successful registration.		

US-02

Title: Balance Top-Up	**Priority**: High	**Estimate (1-5)**: 2 points
User Story: As a user, I want to top up my balance so that I have funds available for transactions.		
Acceptance Criteria: Given the user has an active account, When they select the top-up option and provide amount and method, Then the system processes and confirms the transaction.		

US-03

Title: Send & Receive Money	**Priority**: High	**Estimate (1-5)**: 3 points
User Story: As a user, I want to send and receive money so that I can transfer funds quickly and securely.		
Acceptance Criteria: Given the user has balance, When they enter recipient details and confirm, Then the system processes and notifies both parties.		

US-04

Title: Payments in Stores	**Priority**: Medium	**Estimate (1-5)**: 3 points
User Story: As a user, I want to pay at stores so that I can shop without cash or physical cards.		
Acceptance Criteria: Given the store accepts Nequi, When I scan or enter the merchant code, Then the system deducts the amount and notifies both parties.		

US-05

Title: Expense & Savings Management	**Priority**: Medium	**Estimate (1-5)**: 3 points
User Story: As a user, I want to manage expenses and savings so that I can control my finances and reach goals.		
Acceptance Criteria: Given the user wants a savings goal, When they provide name, amount and deadline, Then the system creates the goal and tracks progress.		

US-06

Title: Credit Application	**Priority**: Medium	**Estimate (1-5)**: 4 points
User Story: As a user, I want to apply for a loan so that I can get financing under favorable conditions.		
Acceptance Criteria: Given the user meets the requirements, When they apply for a loan and enter the amount and term, Then the system evaluates and disburses funds if approved.		

US-07

Title: Mobile Top-Up & Bill Payments	**Priority**: High	**Estimate (1-5)**: 3 points
User Story: As a user, I want to pay services and top up from one platform so that I can manage everything easily.		
Acceptance Criteria: Given the user wants to pay a bill, When they select the service, enter details, and confirm, Then the system processes and notifies successful payment.		

US-08

Title: Security and Support	**Priority**: High	**Estimate (1-5)**: 5 points
User Story: As a user, I want my transactions to be protected and have support when needed so that I feel secure.		
Acceptance Criteria: Given the user logs in from a new device, When the system asks for biometric or code verification, Then the user authenticates and securely accesses their account.		

Initial Database Architecture

1. Components:

    1. **User and Account Management**
    Responsible for storing and managing user personal information and account details, including balances, limits, types, and account status.
    2. **Financial Transaction Management**
    Records and monitors all transactions carried out by users, such as transfers, payments, recharges, and withdrawals, enabling detailed tracking of financial flows.
    3. **Loan Management**
    Handles credit applications, approvals, terms, and monitoring of loans granted to users, such as the "Crédito Propulsor" and "Crédito Salvavidas".
    4. **Authentication and Security Management**
    Controls access mechanisms to the system, including the use of PIN, biometrics, and device verification, ensuring data integrity and confidentiality.
    5. **Support and Communication Management**
    Manages interactions between users and the support area, including inquiries, reports, responses, and the customer communication history.

2. Entities:

    1. **User and Account Management**
        a. User
        b. Account
        c. Device

    2. **Financial Transaction Management**
        a. Movement
        b. Transfer
        c. BillPayment
        d. TopUp
        e. Withdrawal

    3. **Loan Management**
        a. Credit

4. **Authentication and Security Management**
    a. Authentication
5. **Support and Communication Management**
    a. Support

3. Attributes per Entity:

    a. **User**
        - ➔ user_id (PK)
        - ➔ full_name
        - ➔ ID_document
        - ➔ phone
        - ➔ email
        - ➔ country_of_residence
        - ➔ registration_date
        - ➔ status

    b. **Account**
        - ➔ account_id (PK)
        - ➔ user_id (FK)
        - ➔ type (low amount, savings)
        - ➔ balance
        - ➔ monthly_limit
        - ➔ status
        - ➔ creation_date
        - ➔ exempt_4x1000

    c. **Device**
        - ➔ device_id (PK)
        - ➔ user_id (FK)
        - ➔ registration_date

    d. **Movement**
        - ➔ movement_id (PK)
        - ➔ source_account_id (FK)
        - ➔ amount
        - ➔ date_time
        - ➔ status

- type
- reference

e. **Transfer**

- movement_id (PK, FK)
- destination_account_id (FK)

f. **Withdrawal**

- movement_id (PK, FK)
- channel

g. **Recharge**

- movement_id (PK, FK)
- operator
- recharge_number

h. **BillPayment**

- transaction_id (PK, FK)
- service_company
- reference_number

i. **Credit (Loan)**

- transaction_id (PK, FK)
- credit_type (lifeline, booster)
- interest
- term
- due_date
- credit_status

j. **Authentication**

- user_id (PK)
- PIN
- biometrics
- current_device

k. **Entity: Support**

- ticket_id (PK)
- user_id (FK)

- ➔ type (query, report, response)
- ➔ description
- ➔ date
- ➔ status

4. Relationships

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*
a		✓	✓							✓	✓
b	✓			✓							
c	✓										
d		✓			✓	✓	✓	✓	✓		
e				✓							
f				✓							
g				✓							
h				✓							
i				✓							
j	✓										
k	✓										

5. Relationships types

1. User (a) ↔ Account (b): (1:1)
   A user can have multiple accounts, but each account belongs to only one user.

2. User (a) ↔ Device (c): (1:N)
   A user can register multiple devices, but each device is associated with only one user.
3. Account (b) ↔ Movement (d): (1:N)
   An account can have multiple transactions, but each transaction is associated with only one account.

4. Movement (d) ↔ Transfer (e): (1:1)
   Each transfer is a specific type of transaction.

5. Movement (d) ↔ Withdrawal (f): (1:1)
   Each withdrawal is a specific type of transaction.

6. Movement (d) ↔ Recharge (g): (1:1)
   Each top-up is a specific type of transaction.

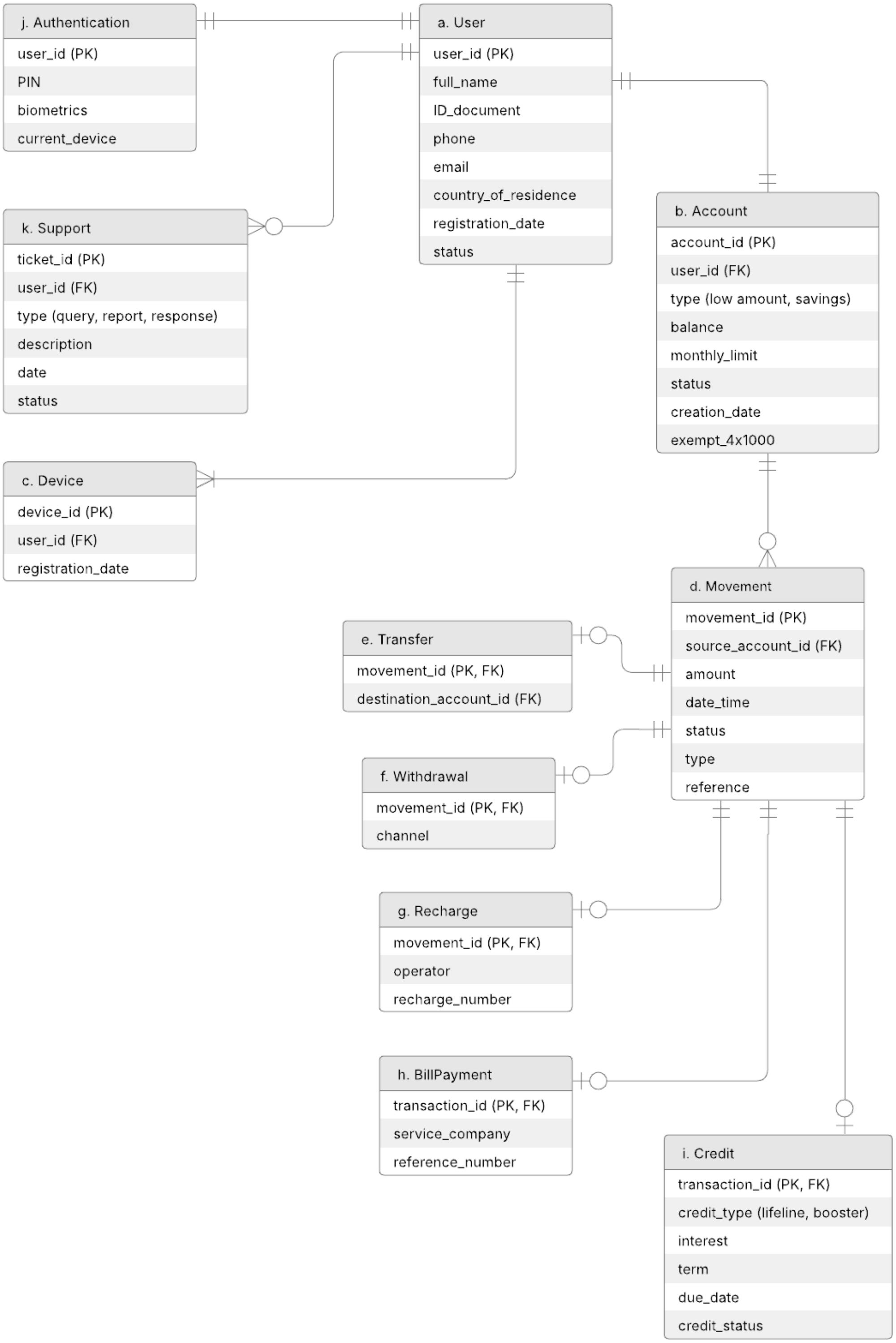
7. Movement (d) ↔ BillPayment (h): (1:1)
   Each bill payment is a specific type of transaction.

8. Movement (d) ↔ Credit (i): (1:1)
   Each loan is a specific type of transaction.

9. User (a) ↔ Authentication (j): (1:1)
   Each user has a unique authentication record.

10. User (a) ↔ Support (k): (1:N)
    A user can have multiple support tickets, but each ticket is associated with only one user.

6. First Entity-Relationship Model Draw


j. Authentication
user_id (PK)
PIN
biometrics
current_device
a. User
user_id (PK)
full_name
ID_document
phone
email
country_of_residence
registration_date
status
k. Support
ticket_id (PK)
user_id (FK)
type (query, report, response)
description
date
status
b. Account
account_id (PK)
user_id (FK)
type (low amount, savings)
balance
monthly_limit
status
creation_date
exempt_4x1000
c. Device
device_id (PK)
user_id (FK)
registration_date
d. Movement
movement_id (PK)
source_account_id (FK)
amount
date_time
status
type
reference
e. Transfer
movement_id (PK, FK)
destination_account_id (FK)
f. Withdrawal
movement_id (PK, FK)
channel
g. Recharge
movement_id (PK, FK)
operator
recharge_number
h. BillPayment
transaction_id (PK, FK)
service_company
reference_number
i. Credit
transaction_id (PK, FK)
credit_type (lifeline, booster)
interest
term
due_date
credit_status