**Universidad Distrital Francisco José de Caldas**

David Santiago Buitrago C 20221020085
Carlos Andres Pardo Angel 20231020041

# First Project Delivery

**Bogotá Abril 2024**

# Description of the project

## Overview

The project aims to develop a mobile application similar to Mercado Libre, a popular e-commerce platform. This app will offer a marketplace where users can buy and sell a wide range of products. One of the key focuses of this project will be the design and implementation of a robust and scalable database to handle large volumes of data effectively.

## Stakeholders

- Primary Stakeholders

Users: Individuals who will buy and sell products on the platform.
Developers: The team responsible for designing, developing, and maintaining the app.
Business Owners: The individuals or company funding the project and expecting a return on investment.

- Secondary Stakeholders

Suppliers: Entities providing products to be sold on the platform.
Payment Gateways: Services facilitating transactions between buyers and sellers.
Regulatory Bodies: Organizations overseeing e-commerce regulations and standards.

## Business Model

1. Revenue Streams
    a. Transaction Fees: Charging a small percentage of each transaction made through the platform.
    b. Premium Listings: Offering paid options for sellers to highlight their products.
    c. Advertising: Displaying targeted ads to users based on their browsing and purchasing history.
2. Value Proposition
    a. Convenience: A user-friendly platform that simplifies buying and selling processes.
    b. Variety: A wide range of products and categories available in one place.
    c. Security: Secure payment gateways and buyer protection policies to ensure safe transactions.
3. Market Strategy
    a. Target Audience: Initially focusing on a specific demographic or geographic region before expanding globally.
    b. Partnerships: Collaborating with suppliers, delivery services, and other e-commerce platforms to enhance the platform's offerings.

## Tools to Use

Development Tools
Programming Languages: Python
Frameworks: Django or Flask for backend development, React Native for cross-platform mobile development.
Version Control: Git and GitHub for code management and collaboration.
Database Tools
Database Management System: PostgreSQL or MongoDB for storing product listings, user profiles, transaction history, etc.
Data Warehousing: Amazon Redshift or Google BigQuery for analytics and reporting.

## user stories

1. As a registered user, I want to be able to log in to the platform to access my account and manage it.

2. As a registered user, I want to be able to register on the platform by providing my personal and contact information to create a new account.

3. As a user, I want to be able to search for products by categories, keywords, or specific features to quickly and efficiently find what I need.

4. As a user, I want to see a list of products with images, detailed descriptions, prices, and ratings to make an informed decision about my purchase.

5. As a user, I want to be able to add products to my shopping cart and then proceed to the checkout process to complete the purchase.

7. As a site administrator, I want to be able to add, edit, and delete products from the catalog to keep the information up-to-date and accurate.

8. As a site administrator, I want to be able to manage user accounts, including the ability to approve registrations, reset passwords, and manage permissions.

9. As a site administrator, I want to be able to view sales reports, performance metrics, and inventory data to make informed business decisions.

10. As a user, I want to be able to leave reviews and ratings on the products I have purchased to help other users in their purchasing decisions.

1. **Logging In for Registered Users:**
    - *Why:* This feature enables registered users to access their personalized accounts securely. By logging in, users can manage their profile information, view order history, and access other personalized features.

2. **User Registration:**
    - *Why:* Registration functionality allows new users to create accounts on the platform, providing essential personal and contact information. This ensures that users can engage with the platform's features fully, including making purchases, leaving reviews, and receiving personalized recommendations.
3. **Product Search by Categories, Keywords, or Features:**
    - *Why:* Users often have specific preferences when searching for products. Allowing them to search by categories, keywords, or features enhances the user experience by providing tailored search results, making it easier for users to find relevant products quickly and efficiently.
4. **Product Listing with Detailed Information:**
    - *Why:* Providing detailed product listings with images, descriptions, prices, and ratings empowers users to make informed purchasing decisions. Access to comprehensive product information enhances user trust and confidence in the platform, leading to increased sales and customer satisfaction.
5. **Adding Products to Shopping Cart and Checkout Process:**
    - *Why:* This functionality streamlines the purchasing process for users, allowing them to add desired products to their shopping carts and proceed seamlessly to checkout. Simplifying the buying process enhances user satisfaction and encourages conversion rates, ultimately driving revenue for the platform.
6. **Product Management for Site Administrators:**
    - *Why:* Site administrators need the ability to add, edit, and delete products from the catalog to ensure that the platform's product offerings remain up-to-date and accurate. This ensures a positive user experience by providing users with access to the latest products and removing outdated or irrelevant items.
7. **User Account Management for Site Administrators:**
    - *Why:* Site administrators require tools to manage user accounts effectively, including approving registrations, resetting passwords, and managing permissions. This functionality ensures user security, compliance with regulations, and the smooth operation of the platform.
8. **Viewing Sales Reports, Performance Metrics, and Inventory Data:**
    - *Why:* Site administrators need access to comprehensive data on sales, performance metrics, and inventory to make informed business decisions. Analyzing this data allows administrators to identify trends, optimize inventory management, and strategize for future growth.
9. **Leaving Reviews and Ratings on Purchased Products:**
    - *Why:* Allowing users to leave reviews and ratings on purchased products promotes user engagement and community interaction. User-generated content serves as valuable social proof for other potential buyers, influencing their purchasing decisions and fostering a sense of trust and transparency on the platform.

3. CONCEPTUAL MODEL

STEP 1. Define entities

E1. USUARIO
E2 PRODUCTOS
E3 PEDIDOS
E4 CATEGORÍAS
E5 CARRITO
E6 HISTORIAL
E7 CALIFICACIONES
E8 COMENTARIOS
E9 PAÍS
E10 CUENTA BANCARIA

STEP 2.

E1. USER

- User ID
- Username
- Email
- Address
- Password (hashed)
- Postal Address
- Phone Number
- Registration Date
- Status

E2. PRODUCTS

- Product ID
- Product Name
- Product Description
- Price
- Category
- Manufacturer/Brand
- Stock/Availability
- Creation Date

E3. ORDERS

- Order ID
- User ID who placed the order

- List of Products
- Order Date and Tim
- e Order Status (pending, processing, shipped, delivered, etc.)
- Shipping Address
- Payment
- Method Used
- Order Total

E4. CATEGORIES

- Category ID
- Category Name
- Category Description

E5. SHOPPING CART

- Cart ID (unique identifier of the cart)
- User ID (to associate the cart with a specific user)
- List of products in the cart
- Cart Creation Date and Time
- Cart Status (active, abandoned, purchased, etc.)

E6. HISTORY

- Order ID (unique identifier of the purchase)
- User ID (to associate the purchase with a specific user)
- Purchase Date and Tim
- e Payment Method Used
- Purchase Total
- Purchase Status (completed, canceled, returned, etc.)

E7. RATINGS

- Rating ID
- User ID who made the rating
- Rated Product ID
- Rating (e.g., 1-5 stars)

E8. COMMENTS

- Comment ID
- User ID who made the comment
- Commented Product ID
- Comment Content
- Comment Date and Time

E9. COUNTRY

- Country Code
- Name

E10. BANK ACCOUNT

- Unique ID
- Bank Name
- Account Number

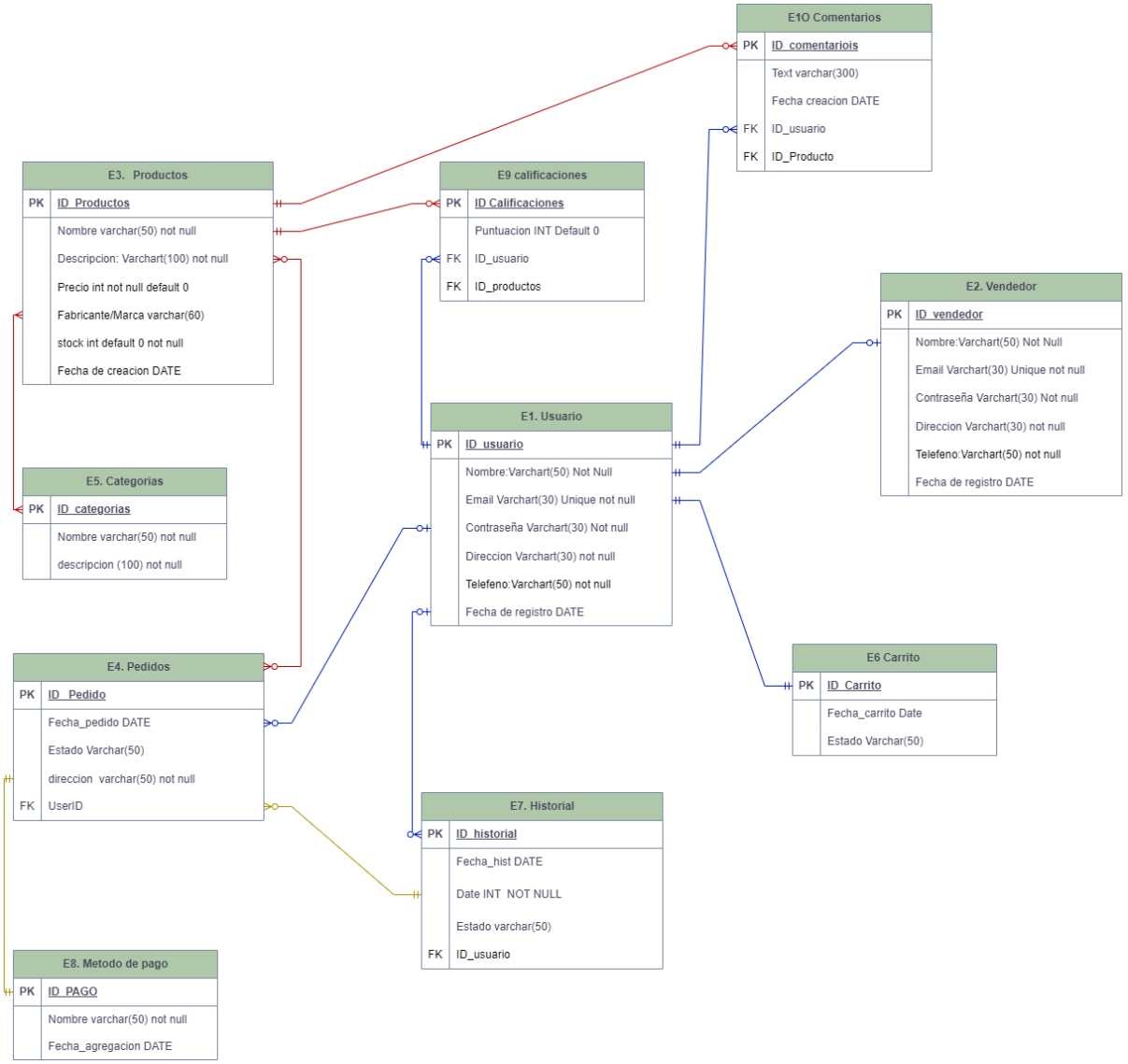
Step 3.


step 4
E1← →E2
E1←一□ E5
E1 ← →E6
E1← 一□E7
E1← 一□ E8
E1← 一□ E9

E3□一□ E4
E3□一□ E5
E3 ←□E8
E3 ←□E9

E4□→ E7
leyenda:
←→ uno a
□□ muchos

**Step 5**




**step 6 romper relaciones muchos a muchos**

E3□—□ E4

ENTITY : Product list

FK Id_product

FK ID_ORDER

E3□—□ E5

ENTIDAD: product_category

FK ID_productS

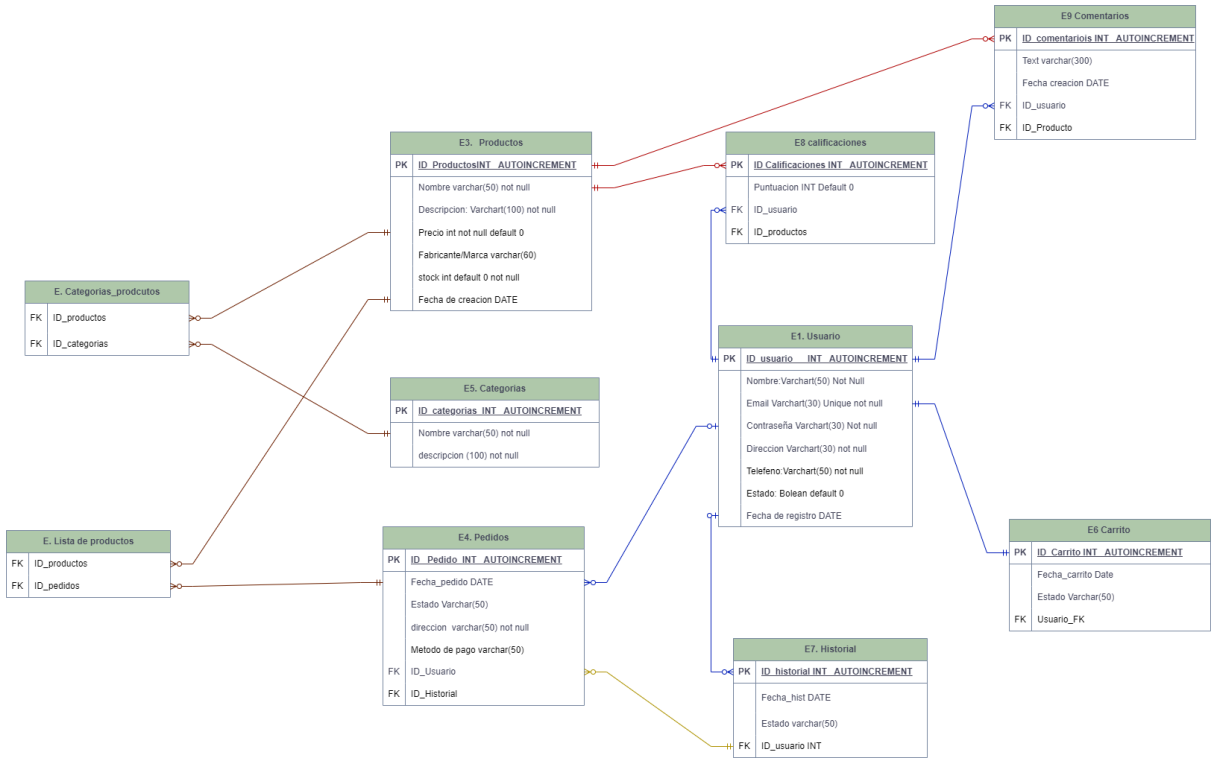
FK ID_categorYS

y relation onr to one

E4 Y E8

By doing a better analysis, we can conclude that the entity e4 (orders) can contain the entity e8 (payment method).

Step7 -step 8-step 9




3. RELATIONAL ALGEBRA

1. Buscar un usuario específico por su nombre de usuario y email:
   σ (UserID = 'user_id' ∧ Email = 'email') (User)
2. Display relevant data for a product with a specific name and within a price range:
   π (product_name, description, price, stock, seller) (σ (price >= min_price ∧ price <= max_price) ( σ (product_name = 'specific_name') (product) ))
3. Display relevant data of a product through its category:
   π (product_name, description, price, stock) (Products ⋈σ (Category_Name = 'category_name') (Categories))
4. View the date and status of the purchase history of a user:
   π (buy_date, status) (ShoppingHistory)

5. View all orders of a seller by their shipping status:
   π (date, shipping_address, status, delivery_date, method) ( σ (status="entered_status") (Shipping) ⋈ (Seller ⋈ Order))
6. Display the shopping cart list of a specific user:
   π (date_added, quantity, name, price) ((Product) ⋈ ((Product_ShoppingCart) ⋈ (User ⋈ ShoppingCart)))

7. Display comments for a product:
   π (name, text, date) (Comments ⋈ σ (product_name="product_name") (Product))

**5.**

**Strategy to Generate a Dummy Database:**

- **Generate Random Data:**
  - Use programming languages like Python to generate random data for each entity. Libraries like Faker can be helpful for creating realistic names, addresses, emails, etc.
  - Maintain Relationships:Ensure that the generated data maintains the relationships between entities. For example, a product should have a valid category and be associated with a real vendor.

- **Data Consistency:**
  - Validate the generated data to ensure consistency. For instance, the stock of a product should not be negative, and the price should be realistic.
- Scale the Data:
  - Depending on your testing requirements, scale the data to create a large dataset. This will help you test the performance of your queries under different load conditions.
- Data Injection:
  - Use database scripts or ORM (Object-Relational Mapping) tools to inject the generated data into the database.
- GitHub: GitHub hosts a variety of open-source projects that may include sample datasets for testing purposes. You can search for repositories related to your domain and see if they provide any sample data.
- Yelp Open Datasets: This dataset comprises a portion of Yelp's business, reviews, and user data. It was originally compiled for the Yelp Dataset Challenge, providing students with an opportunity to delve into Yelp's data for research or analysis and to share their findings. In the latest dataset, you'll find details about businesses located in 8 metropolitan areas across the United States and Canada.