Final Proyect Databases II

Cuellar B. Paola A.
Ebratt S. Johan D.
Universidad Distrital Francisco José de Caldas

Semester 2025-I

1 Business Canvas Model

The Canva Model is a simple methodology that defines the business strategy and the target audience. Its objective is to provide a clear understanding of the type of business to be created and introduced in the market, who it is aimed at, how it will be sold and how income will be obtained.

Having this clear, we want to develop a social network, UDshare, which allows to connect users with their followers in a safe way, the followers will be other users, public figures and companies. In UDshare it will be possible to make publications with multimedia content and users will be able to interact with them, with comments and reactions in addition users will be able to follow their publications.

KEY PARTNERS	KEY ACTIVITIES	KEY RESOURCE	VALUE PROPOSITIONS	CHANNELS	CUSTOMER SEGMENTS
Advertising Companies Companies that want to advertise their resources, representing a source of income and attracting users interested in their offerings.	**User Identity Verification** Implementation of secure identity verification using official documents based on the user's country, ensuring full compliance with data privacy regulations.	**Human Talent** Skilled developers and DevOps professionals proficient in modern technologies, responsible for continuous development, deployment, and maintenance of the platform.	We offer a personalized social media platform that connects users with shared interests through intelligent recommendations and secure identity verification. Our propuse fosters meaningful interactions between users and companies and continuously improves the experience using data analytics and business intelligence.	**Core Platform** The web applications serve as the main interface for users to interact with the platform. Through these channels, users can publish content, engage with others.	**General Audience** Our platform is designed for a broad audience, targeting anyone over the age of 15 who seeks to connect with others through shared interests, trends, and personalized content.
Other Social Media Platforms Interconnectivity with other platforms helps us attract new users and increase reach.	**Interest-Based Interaction Channels** Development of official interest spaces where users and companies can engage around shared topics and opportunities.	**Security Frameworks** Resources and technologies to guarantee secure user authentication, identity verification, and privacy compliance.	**CUSTOMER RELATIONSHIPS** Direct relationships with users to foster trust and long-term engagement. Immediate and effective response to account-related issues to ensure a safe and smooth user experience. 24/7 support availability through chat and email for quick resolution of concerns. Feedback and interaction channels that allow users to report issues, suggest improvements, and feel heard. Customer priority support, especially for verified accounts and business partners.	**Distribution and Growth** We rely on global distribution alliances with key partners to expand our reach efficiently. In addition, integration with other social media platforms allows users to share content and attract new audiences organically.	**Users** They are the primary customers who engage daily with the platform, posting content, following others, and consuming media. Their needs revolve around usability, content relevance, privacy, and connection.
Data Analysis Companies Partners with whom we exchange information to improve recommendations and provide valuable insights for their analysis.	**Recommendation Engine Management** Continuous refinement of algorithms that personalize user feeds based on behavior and preferences.	**Data Storage and Processing Systems** Data centers and warehouses that ensure efficient handling of big data, supporting ingestion and retrieval at scale.		**Support and Communication** To ensure a responsive and user-centric experience, we offer immediate support through online communication channels such as in-app chat and email. We also provide topic-based channels where users and companies can interact around shared interests.	**Companies** Businesses and service providers use the platform to advertise, engage with targeted audiences, and participate in interest-based channels. They seek visibility, analytics, and effective communication with users.
Moderation and NLP API Providers External services that help detect hate speech, spam, and inappropriate content automatically.	**Data Ingestion & Analytics** Constant ingestion of user interactions for real-time analytics and improved recommendations.	**Business Intelligence & Analytics Tools** Systems and expertise for analyzing user behavior and market trends to refine recommendations and target content.			**Personages** These are socially influential users, public figures, thought leaders, or content creator, who use the platform to amplify their reach and shape public opinion. Their needs include audience analytics, content promotion, and identity verification.
Cloud Infrastructure Providers Enable scalable, high-availability hosting and distributed data storage.		**Cloud Infrastructure** Scalable and reliable cloud services supporting multi-location access, fast queries, and high availability.			**Members** Paying users who subscribe to premium features such as enhanced visibility, analytics, post boosts, or exclusive content. They expect added value and personalization for their investment.

COST STRUCTURE	REVENUE STREAMS
Data analysis and storage Major investments go toward processing and storing vast amounts of user data to deliver personalized experiences and accurate recommendations	**Membership subscriptions** Users pay for access to premium features such as exclusive content, performance analytics, and improved visibility on the platform.
Technology and network infrastructure Costs include cloud services, high-speed connectivity, and computing power to ensure scalability, fast response times, and user interaction.	**Advertising** Companies and public figures invest in ad placements to reach specific audiences, increasing brand exposure and engagement.
Server maintenance Ongoing expenses involve maintaining server uptime, applying security patches, system monitoring, and ensuring platform availability 24/7.	**Data insights services** Businesses pay for anonymized, aggregated analytics that help them understand trends and improve their strategies, respecting privacy standards.

Figure 1: Business Canvas Model Source: Authors

2 User Stories

The following user stories describe the functional and non-functional requirements of the system from the end user's perspective. Each story is written in a format that captures a specific need and the value it brings to the user within the social network being developed.

The purpose of documenting these stories is to facilitate a shared understanding among the development team and stakeholders about what is expected from the system. These stories form the foundation of the product backlog and guide the iterative and incremental development of the solution.

Each user story has been assigned a priority (High, Medium, or Low), which reflects its impact on system functionality and the value it delivers. Additionally, an effort estimation is included for each story, expressed in story points. These points do not represent a direct measure of time (e.g., hours or days), but rather a relative measure of complexity, uncertainty, and the amount of work required to implement the story.

The estimation technique used was Scrum Poker or Planning Poker, a collaborative practice in agile methodologies where each development team member independently assigns effort points to a story. The differences in estimations are then discussed until a

group consensus is reached, promoting a shared understanding of the scope and encouraging active participation from all team members.

Below is the set of prioritized and estimated user stories for this phase of the project.

ID	**UH-1: User Registration**
Priority	High
Estimate	8
User Story	As a user, I want to register on the platform by providing my basic information so that I can access all the features of the social network.
Acceptance Criteria	• Given a visitor wants to create an account on the platform • When they submit the registration form with their name, email, password, and date of birth • Then the system must validate: – The email has a valid format and is unique – The password meets security requirements (min. 8 characters, including uppercase, numbers, special characters) – All required fields are completed – If all validations pass, create the user account and grant access

ID	**UH-2: User Login**
Priority	High
Estimate	5
User Story	As a user, I want to log in using my email and password or a third-party account so that I can securely access the social network.
Acceptance Criteria	• **Successful Login:** Given a registered user has valid credentials or linked third-party account. When they log in, then authenticate and generate a JWT token with expiration. • **Invalid Credentials:** Given incorrect credentials. When they attempt to log in, then show generic error (no email existence hint). • **Account Lockout:** If failed attempts exceed threshold, temporarily lock account.

ID	UH-3: Secure Authentication Implementation
Priority	High
Estimate	8
User Story	As a user, I want to log in securely using a token-based authentication method so that I can access the platform safely.
Acceptance Criteria	• Token includes digital signature and expiration • Third-party login issues secure tokens under same standards • Tokens follow platform-wide security policies

ID	UH-4: View Profile
Priority	High
Estimate	8
User Story	As a user, I want to be able to view my profile so that I can see how my account appears to others.
Acceptance Criteria	• Profile shows name, biography, profile picture, and cover photo • Biography shown in full if near character limit • Sensitive data like email shown securely to owner only

ID	UH-5: Profile Editing
Priority	Medium
Estimate	5
User Story	As a user, I want to be able to edit my profile to personalize my account on the social network.
Acceptance Criteria	• Update name, biography, picture, and cover photo • Notify user on successful update • Prevent save if biography exceeds character limit • Store history of changes to sensitive data (email/username)

ID	UH-6: Posting
Priority	High
Estimate	8
User Story	As a user, I want to write and post up to 500 characters to share ideas and updates.
Acceptance Criteria	• Allow text input up to 500 characters • Prevent further input or show error if limit exceeded • Save and display post in timeline upon publishing

ID	UH-7: Add Multimedia to Post
Priority	Medium
Estimate	8
User Story	As a user, I want to create posts with text and media to share updates with my followers.
Acceptance Criteria	• Attach media to posts optionally • Publish post with media to timeline

ID	UH-8: Follow Other Users
Priority	High
Estimate	5
User Story	As a user, I want to follow others to see their posts and stay updated.
Acceptance Criteria	• Add users to "Following" list • Show their posts in user's timeline • Display followers/following count

ID	UH-9: Unfollow Users
Priority	Medium
Estimate	5
User Story	As a user, I want to follow and unfollow users to personalize my content feed.
Acceptance Criteria	• Toggle follow/unfollow on user profile • Feed updates accordingly

ID	UH-10: Interact with Posts
Priority	Medium
Estimate	5
User Story	As a user, I want to interact with posts and find them easily later.
Acceptance Criteria	• Allow like/favorite interaction • Highlight interaction visually • List interacted posts in user profile

ID	UH-11: Comment on Posts
Priority	High
Estimate	8
User Story	As a user, I want to comment on posts so that I can engage in conversations.
Acceptance Criteria	• Add comment below the post • Notify original post author

ID	UH-12: Share Post
Priority	Low
Estimate	3
User Story	As a user, I want to share posts so that they can be seen by my followers.
Acceptance Criteria	• Re-publish post to sharer's profile timeline • Visibility according to sharer's privacy setting

ID	UH-13: Edit a Post
Priority	Medium
Estimate	5
User Story	As a user, I want to edit a post after publishing it to correct errors or add information.
Acceptance Criteria	• Edit post content in editable text field • Save changes • Show "edited" label on updated posts

ID	UH-14: Configure Post Privacy
Priority	Medium
Estimate	8
User Story	As a user, I want to configure post privacy to control who sees my content.
Acceptance Criteria	• Set privacy level (Public, Followers, Private) when posting • Enforce visibility based on selected privacy

ID	UH-15: Reported Post
Priority	Medium
Estimate	5
User Story	As an admin, I want to review reported content to moderate and enforce standards.
Acceptance Criteria	• View report details from moderation dashboard • Allow actions like delete, ignore, or ban

ID	UH-16: Reported Users
Priority	Medium
Estimate	5
User Story	As an admin, I want to suspend users who violate rules to ensure platform safety.
Acceptance Criteria	• Review user violation history • Suspend account and notify the user

ID	UH-17: Verification Process
Priority	High
Estimate	8
User Story	As an admin, I want to manage verification requests to ensure credibility for public accounts.
Acceptance Criteria	• Approve or reject verification request • Reflect decision on user profile

ID	UH-18: Boost Post
Priority	High
Estimate	5
User Story	As a member, I want to boost my posts so they reach a larger audience.
Acceptance Criteria	• Post is prioritized in timelines • Post is labeled as "promoted" • Requires active premium subscription

ID	UH-19: Advanced Analytics to Members
Priority	Low
Estimate	5
User Story	As a member, I want access to advanced analytics to evaluate my content's performance.
Acceptance Criteria	• Display metrics such as impressions, engagement rate, and demographics • Accessible from profile for premium users

3 System-Level Analysis and Requirement Justification

The proposed platform is a social networking application intended to support user-generated content, profile management, and personalized feeds, similar to existing platforms such as X (formerly Twitter). Users will be able to interact through posts, comments, likes, and shares. Key technical constraints include the need for scalability to support high user volumes, rapid data delivery for a seamless experience, and robust moderation to ensure community safety.

This system is expected to handle thousands of concurrent users and deliver content within strict performance limits. For example, loading a timeline or executing a search query must occur within 100 milliseconds under typical conditions. This benchmark is informed by user expectations based on industry standards from leading platforms, where high responsiveness directly affects user retention. Additionally, user interactions like posting or following must update system state with minimal latency and ensure consistent delivery across geographically distributed data centers.

Security and privacy are essential due to the collection and storage of sensitive user information. Thus, the system must implement token-based authentication, enforce encryption, and comply with international regulations such as GDPR.

These factors inform the following functional and non-functional requirements.

Functional Requirements Specification

Requirement ID	**FR-001: User Registration**
Version	1
Date	28/April/2025
Dependency	• Database • Authentication service
Description	The system must allow a new user to register using an email or an external authentication provider (OAuth).
Specific Data	• Username • Email address • Password or external auth token
Priority	High
Status	In development
Comments	The system must validate that the email is not already in use.

Requirement ID	**FR-002: Create Post**
Version	2
Date	20/May/2025
Dependency	• Database • Media service
Description	The system must allow users to create and publish messages with text, images, or videos.
Specific Data	• Post content (text) • Media files (optional) • Timestamp of publication
Priority	High
Status	In development
Comments	Content filtering and quality checks should be applied before publishing.

Requirement ID	FR-003: Follow/Unfollow Users
Version	2
Date	20/May/2025
Dependency	• Database • Notification service
Description	The system must allow users to follow or unfollow other users, modifying their personalized feed accordingly.
Specific Data	• Follower user ID • Followed user ID • Timestamp of the action
Priority	High
Status	In development
Comments	A notification should be sent if the followed user has this setting enabled.

Requirement ID	FR-004: Like and Repost Content
Version	1
Date	28/April/2025
Dependency	• Database • Notification system
Description	The system must allow users to like or repost content created by others.
Specific Data	• User ID • Post ID • Type of interaction (like or repost) • Timestamp
Priority	High
Status	In development
Comments	Notifications should be generated for the content creator.

Requirement ID	FR-005: Comment on Posts
Version	1
Date	28/April/2025
Dependency	• Database • Moderation system
Description	The system must allow users to write and publish comments on existing posts.
Specific Data	• Comment text • Post ID • Commenting user ID • Timestamp
Priority	High
Status	In development
Comments	Comments must pass a content validation process before being displayed.

Requirement ID	FR-006: Report Content
Version	2
Date	20/May/2025
Dependency	• Database • Moderation engine
Description	The system must allow users to report posts they consider inappropriate, offensive, or harmful.
Specific Data	• Report reason • Post ID • Reporting user ID • Timestamp
Priority	High
Status	In development

Comments	Reports should trigger a review process by moderators.

Non-Functional Requirements Specification

Requirement ID	**NFR-001: Performance for Search**
Version	1
Date	28/April/2025
Dependency	• Database • Caching engine
Description	The system must ensure that search queries and timeline loading are executed in under 100 milliseconds.
Specific Data	• Search engine response time • Timeline fetch latency
Priority	High
Status	Planned
Comments	Response time must be validated under peak load conditions.

Requirement ID	**NFR-002: Scalability**
Version	3
Date	15/June/2025
Dependency	• Cloud infrastructure • Load balancer
Description	The system must support elastic scaling to handle traffic surges during peak activity periods.
Specific Data	• Auto-scaling policies • Concurrent user support
Priority	High
Status	Planned
Comments	Stress testing should be performed before deployment.

Requirement ID	NFR-003: High Availability
Version	1
Date	28/April/2025
Dependency	• Infrastructure as Code • Group of geographically distributed servers
Description	The system must guarantee 99.9% uptime with regional redundancy to minimize service disruption.
Specific Data	• Regional server replicas • Downtime tolerance
Priority	High
Status	Planned
Comments	Use of failover systems and availability zones is required.

Requirement ID	NFR-004: Security Standards
Version	2
Date	20/May/2025
Dependency	• OAuth provider • Encryption layer
Description	The platform must enforce encrypted data storage and secure authentication via OAuth, with API rate limiting.
Specific Data	• Encryption algorithm • Auth flow compliance • API rate thresholds
Priority	High
Status	Planned
Comments	Must comply with GDPR and platform privacy standards.

Requirement ID	NFR-005: Real-Time Data Ingestion
Version	2
Date	20/May/2025

Dependency	• OAuth provider • Encryption layer
Description	The platform must enforce encrypted data storage and secure authentication via OAuth, with API rate limiting.
Specific Data	• Encryption algorithm • Auth flow compliance • API rate thresholds
Priority	High
Status	Planned
Comments	Must comply with GDPR and platform privacy standards.

Requirement ID	**NFR-006: Multi-Region Access**
Version	3
Date	15/June/2025
Dependency	• Global CDN • Data replication system
Description	User data must be accessible from different regions with low latency, using synchronized replicas.
Specific Data	• Regional replication delay • Access latency
Priority	High
Status	Planned
Comments	Ensure consistency and data integrity across replicas.

Requirement ID	**NFR-007: Real-Time Data Ingestion**
Version	3
Date	15/June/2025

Dependency	• ML engine • Data pipeline
Description	The system must implement content-based and collaborative filtering algorithms for user feed personalization.
Specific Data	• User interaction data • Embedding vectors
Priority	Medium
Status	Planned
Comments	Must comply with GDPR and platform privacy standards.

Requirement ID	**NFR-008: Business Intelligence Module**
Version	1
Date	28/April/2025
Dependency	• Data warehouse • BI tool (Superset/Power BI)
Description	The system must integrate a BI module for generating insights on user engagement, content performance, and growth.
Specific Data	• User metrics • Post interactions • Regional trends
Priority	Medium
Status	Planned
Comments	Dashboards must be accessible to managers and admins.

4 Entity-Relationship Model

This model is a fundamental tool in the design of databases, because it allows to represent in a clear and structured way the entities of the system, their attributes and the relationships between them. Its use facilitates the understanding of the problem domain for both developers and users, allows the detection of inconsistencies or redundancies at

an early stage of development, and serves as a basis for the construction of the logical and physical model in a database management system. In addition, the MER allows us to visualize data integrity and consistency by defining primary and foreign keys and constraints, improves system documentation and facilitates long-term maintenance and evolution.

Figure: Entity-Relationship Model

5 Architecture Diagram

The following diagram presents the architecture of the social network currently under development. This diagram illustrates the main components of the system, as well as the

interactions between them, providing an overview of its structure and functionality.

Figure 2: Architecture Diagram

6 Information Requirements

The following section provides a list of the information related to the system requirements. This compilation details the essential functionalities and constraints identified during the analysis phase.

Requirement	**Detailed Description**	**Target Users**	**Related UH**
Personalized post feed	Retrieve recent posts from followed users, prioritized by relevance, date, or interactions.	Members	UH-1, UH-2, UH-11
User profile details	Display name, profile picture, biography, linked social accounts, and activity stats.	Members, companies	UH-4, UH-5
Posts by user	List all posts made by a specific user, including date, content, multimedia, and reactions.	Members, analysts	UH-1, UH-2, UH-6
Post comments	Retrieve comments on a post in chronological order.	All users	UH-11
Reactions per post	Obtain the number and type of reactions (likes, hearts, etc.) for a specific post.	All users	UH-10
Followers and following	Retrieve follower/following lists, with filters by name or user type.	Members	UH-2
User and post search	Search users by name or nickname, and posts by keyword, hashtag, or topic.	All users	UH-12

Requirement	Detailed Description	Target Users	Related UH
Reported content	Show details of reported users or posts for review by administrators.	Administrators	UH-15, UH-16, UH-7
Premium user statistics	Retrieve performance data of posts: views, engagement, follower growth, reach.	Premium members	UH-19
Multimedia linked to posts	List uploaded images/videos with technical details and post associations.	Content creators	UH-12, UH-7
Post edit history	View previous versions of a post with timestamps for auditing or recovery.	Admins, advanced users	UH-6
Recent user activity	Log of recent actions: posts, comments, reactions, new/unfollowed users.	Account owners	UH-1, UH-19
Queries by user type or role	Retrieve filtered user lists by role (person, company, admin) or attributes (verified, influencer, active).	Admins, companies	UH-1, UH-16
Posts by visibility level	List posts by visibility: public, private, followers only, with proper access controls.	Members, admins	UH-6
Promoted or featured posts	Show timeline-prioritized posts due to being boosted or marked as relevant.	All users	UH-19
Hashtag tracking and search	Retrieve the most used hashtags (top N), allow search by hashtags, and show latest posts associated with a given hashtag.	All users, analysts	UH-12, UH-1

7 Query Proposals

This section presents a set of query proposals designed to demonstrate how key information can be retrieved from the system using both SQL and NoSQL approaches. The SQL queries are based on structured relational models, while the NoSQL examples utilize document-oriented paradigms, reflecting the flexibility and scalability of modern database solutions.

7.0.1 SQL Queries

- **Retrieve public user profile**

```
SELECT name, nickname, email, profile_image
FROM user
WHERE id = 101;
```

- **List followers of a user**

```
SELECT u.nickname
FROM follower f
JOIN user u ON f.fk_follower_user = u.id
WHERE f.fk_followed_user = 101;
```

- **Top users by number of posts**

```
SELECT u.nickname, COUNT(p.id) AS post_count
FROM user u
JOIN post p ON p.fk_user_id = u.id
GROUP BY u.nickname
ORDER BY post_count DESC
LIMIT 10;
```

- **Retrieve comments of a post**

```
SELECT u.nickname, c.text_comment, c.date_create
FROM comment c
JOIN user u ON c.fk_user_comment_id = u.id
WHERE c.fk_post_id = 501
ORDER BY c.date_create ASC;
```

- **Reactions per type for a post**

```
SELECT cr.label, COUNT(r.id) AS total
FROM reaction r
JOIN cat_reaction cr ON r.fk_reaction_id = cr.id
WHERE r.fk_post_id = 501
GROUP BY cr.label;
```

- **Recent posts by a user**

```
SELECT text_post, date_post
FROM post
WHERE fk_user_id = 101
ORDER BY date_post DESC
LIMIT 5;
```

- **Posts with attached multimedia**

```
SELECT p.id, p.text_post, a.name AS file_name, a.format
FROM post p
JOIN attached_multimedia a ON a.id = p.id;
```

- **Edit history of a post**

```
SELECT hp.text_version, hp.date_modificate
FROM historical_post hp
WHERE hp.id = 501;
```

- **Top users by number of reactions**

```
SELECT u.nickname, COUNT(r.id) AS total_reactions
FROM reaction r
JOIN user u ON r.fk_user_react_id = u.id
GROUP BY u.nickname
ORDER BY total_reactions DESC
LIMIT 10;
```

- **Private posts of a user**

```
SELECT id, text_post
FROM post
WHERE fk_user_id = 101 AND visibility_type = 'private';
```

- **Users with most followers**

```
SELECT u.nickname, COUNT(f.fk_follower_user) AS total_followers
FROM user u
JOIN follower f ON u.id = f.fk_followed_user
GROUP BY u.nickname
ORDER BY total_followers DESC
LIMIT 10;
```

- **Most reported users**

```
SELECT u.nickname, COUNT(r.id) AS report_count
FROM report_user r
JOIN user u ON r.fk_report_to_user = u.id
GROUP BY u.nickname
ORDER BY report_count DESC
LIMIT 10;
```

7.0.2 NoSQL Queries

- **Query recent hashtags used (Redis sorted set)**

```
ZREVRANGE hashtags 0 9 WITHSCORES
```

- **Get latest posts by hashtag (Redis list)**

```
LRANGE hashtag:tech:posts 0 9
```

8 Concurrency Analysis

Scenarios of Concurrent Access

- **Lost Updates during Posting:** When multiple users post at the same time, the application may miss updates or display inconsistent timelines because concurrent operations are not properly synchronized.
  Solution: Use pessimistic locking or versioning to prevent overwrites and ensure timeline consistency.

- **Race Conditions on Reactions:** When many users react to a post at the same time, the application may display an incorrect reaction count because a race condition occurs between the processes updating the counter.
  Solution: Apply atomic operations, increment locks, or optimistic concurrency control with conflict detection.

- **Concurrent Profile Edits:** When a user edits his or her profile while another process is also modifying it, the application may overwrite information or lose updates because data locks or versions are not handled correctly.
  Solution: Implement version control, or use SELECT FOR UPDATE.

- **Insert Contention on Comments:** When many users comment on the same publication at the same time, the application may experience locks or congestion in the insertions because there is contention in the concurrent access to the database.
  Solution: Use row-level locks and batch inserts, or scale horizontally with partitioned tables.

- **Dirty Reads during Moderation:** When a user reports a post while it is being moderated, the application may show dirty reads because the unconfirmed information is accessed before the end of the moderation transaction.
  Solution: Set the isolation level to Read Committed or higher Repeatable Read, Serializable.

- **Phantom Reads on Concurrent Search:** When search filters are applied while inserting new data, the application may show incomplete or inconsistent results because the snapshot of the data is not controlled under an appropriate level of isolation.
  Solution: Use the Serializable isolation level or predicate locking to avoid phantom reads.

9 Distributed Database Design

To ensure scalability, high availability and low latency in various geographical regions, the system adopts a hybrid distributed architecture combining SQL and NoSQL technologies.

Architecture Overview

A hybrid approach is adopted using PostgreSQL for structured data and MongoDB/Redis for flexible, high-performance access.

Distributed Model

UDshare takes a hybrid approach using PostgreSQL for structured and relational data, and MongoDB/Redis for flexible, high-performance access to semi-structured data such as trending hashtags, user activity and timelines. The architecture will be distributed in three major geographic regions:

- **Region A (Colombia):** Main region for Latin America.
- **Region B (USA):** Central hub for data routing and replication.
- **Region C (Europe):**Ensures data localization compliance and low latency access for European users.

Figure 3: Distributed Diagram DB

Mechanisms

- **PostgreSQL Sharding by Region:** Each region stores structured user-related data in PostgreSQL. Region sharding reduces write contention and localizes data access.

- **MongoDB Replication Sets:** MongoDB is used for semi-structured documents (entries, comments). Replica sets guarantee high availability and support read scaling on a per-region basis.
- **Redis Hash Slots:** Redis manages high-speed in-memory data, such as trending hashtags and session tokens, distributed by partitioning hash slots.
- **Replication between regions:** Bi-directional replication synchronizes critical data sets across SQL and NoSQL nodes to ensure end-to-end consistency and disaster recovery readiness.

This hybrid architecture, segmented by region, is justified by the key needs of the project: the handling of large volumes of data such as millions of posts, reactions and interactions in real time; the global reach with users located in Latin America, North America and Europe; the delivery of content in real time, where Redis allows to quickly update the most relevant content; and the integrity of the data, guaranteed by PostgreSQL through strong consistency mechanisms for the management of user accounts and relationships.

10 Performance Improvement Strategies

To improve system performance and minimize latency in distributed regions, the following strategies will be applied:

Strategy 1: Horizontal Partitioning (Sharding)

Messages and other large-scale content are horizontally partitioned (sharding) by region or by user ID range. This means that data is spread across multiple database instances, which reduces the load on a single node.

Advantages:

- Increases performance by allowing parallel writes and reads on independent partitions.
- Reduces contention and bottlenecks in high traffic regions.

Challenges / Trade-offs:

- Queries between shards become more complex and may require aggregation mechanisms.
- Difficult to maintain referential integrity between fragments.

Strategy 2: Replication for High Availability

A combination of PostgreSQL master-slave replication and MongoDB replica sets ensures that data is continuously synchronized and redundantly stored.

Advantages:

- Enables failover in case of node failure.
- Distributes read queries to replicas, improving read performance.

Challenges / Trade-offs:

- Replication lag may cause slightly stale reads in asynchronous replicas.
- Increases storage requirements and operational complexity.

Strategy 3: Parallel Query Execution

Enable parallel sequential scans and joins in PostgreSQL for analytical queries, and use Redis Streams to process real-time events (e.g., trending hashtags, live notifications) concurrently.

Advantages:

- Maximizes CPU utilization and reduces response time for large queries.
- Real-time analytics and dashboards are supported with minimal delay.

Challenges / Trade-offs:

- Requires tuning PostgreSQL parameters and understanding workload patterns.
- Real-time stream processing may require back-pressure management and scaling policies.

References

1. MongoDB Inc., "Database Sharding: Concepts and Examples". MongoDB Developer Resources, 2020. [Online]. Available: `https://www.mongodb.com/resources/products/capabilities/database-sharding-explained#:~:text=Sharding%20is%20a%20method%20for,See%20more%20on%20the%20basics`
2. R. Ragul and A. P. Rajan, "Efficient Horizontal Scaling of Databases using Data Sharding Technique". International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 5, pp. 590–593, 2020. [Online]. Available: `https://www.ijitee.org/wp-content/uploads/papers/v9i5/E2418039520.pdf#:~:text=1,unique%20from%20other%20partition%20tables`
3. Redis Project, "Redis Cluster Specification". Redis Documentation, sec. Hash Slot, 2019. [Online]. Available: `https://redis.io/docs/latest/operate/oss_and_stack/reference/cluster-spec/#:~:text=Each%20master%20node%20in%20a,however%20the%20serving`

4. PostgreSQL Global Development Group, "Parallel Query in PostgreSQL – Documentation v17". PostgreSQL Manual, ch. 15, 2023. [Online]. Available: https://www.postgresql.org/docs/current/parallel-query.html#:~:text=PostgreSQL%20can%20devise%20query%20plans,This%20chapter%20explains%20some