* Full name: Iván Kaleb Ramírez Torres
* NAO ID: 3357
* Date: September 30, 2025
* Name of the pathway in which you are enrolled: Bécalos Techno Ready
* Title of the Challenge: **Challenge 3 – Server and Database Commands**

# Tracking Tables

## Table 1 – Requirements list

|  |  |
| --- | --- |
| Sprint | Requirements |
| Sprint 1:  **Research and document the Google Scholar API, producing a technical report and creating a GitHub repository to manage the project.** | 1. Constructing a Backlog for Challenge 2. 2. Build a Roadmap integrating all the requirements, dates, participants and Sprints for Challenge 2. 3. Document:  * Document endpoints (API URLs) * Describe authentication methods * List query parameters * Explain response formats * Detail usage limits * Provide code examples.  1. Create repository  * Add README.md (purpose, functionalities, relevance). * Upload technical report. * Configure access for Digital NAO team. |
| Sprint 2:  **Develop Java code to perform GET requests to the Google Scholar Author API using the MVC (Model-View-Controller) design pattern** | - Design model to represent author information  - Implement view to display author search results  - Perform GET requests with HttpClient - Process API responses - Handle errors/exceptions - Update view  - Combine model, view, controller - Test with sample author searches  - Push MVC code - Update README.md with Sprint 2 deliverables - Ensure repo permissions |
| Sprint 3:  **Integrate the data retrieved from the API into a database, structuring the database appropriately and handling pagination and API errors.** | - Choose DBMS (MySQL/PostgreSQL/SQLite) - Design schema with table articles (id, title, authors, pub\_date, abstract, link, keywords, cited\_by) - Integrate API data for 2 researchers & 3 articles each - Allow IDE connection wizards - Store API responses in DB  - Implement robust error handling (network, API, DB) - Respect API usage restrictions  - Push DB schema + integration code - Update README.md with Sprint 3 deliverables - Configure repo permissions |
| Final Project:  **Document Analysis & Results for**  **Challenge 3** | Make a video presentation explaning Analysis & Result of the Challenge 2 |

## Table 2: Prioritize list – Challenge 2

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Stages (Steps) | Time Estimation | Deliverables |
| Technical Report on Google Scholar API | - Document endpoints (API URLs) - Describe authentication methods - List query parameters - Explain response formats - Detail usage limits - Provide code examples | 3h | Technical report (document) |
| GitHub Repo Setup (Initial) | - Create repository - Add README.md (purpose, functionalities, relevance) - Upload technical report - Configure access for Digital NAO team | 4h | Public GitHub repo with initial documentation |
| Java MVC – Data Model | - Design model to represent author information | 2h | Java model class for Author |
| Java MVC – View | - Implement view to display author search results | 6h | Java view (console/GUI) |
| Java MVC – Controller | - Perform GET requests with HttpClient - Process API responses - Handle errors/exceptions - Update view | 2h | Java controller with working API connection |
| Java MVC – Integration & Testing | |  | | --- | |  |  |  | | --- | | - Combine model, view, controller - Test with sample author searches | | 6h | Functional Java MVC app |
| GitHub Repo Update (MVC) | - Push MVC code - Update README.md with Sprint 2 deliverables - Ensure repo permissions | 3h | Updated GitHub repo with MVC code |
| Database Setup | - Choose DBMS (MySQL/PostgreSQL/SQLite) - Design schema with table articles (id, title, authors, pub\_date, abstract, link, keywords, cited\_by) | 8h | Database schema |
| Database Integration | - Integrate API data for 2 researchers & 3 articles each - Allow IDE connection wizards - Store API responses in DB | 5h | Functional DB with test data |
| Error Handling & API Limits | - Implement robust error handling (network, API, DB) - Respect API usage restrictions | 3h | Stable DB integration with error management |
| GitHub Repo Update (DB) | - Push DB schema + integration code - Update README.md with Sprint 3 deliverables - Configure repo permissions | 2h | Final GitHub repo with DB integration |
| Make a video presentation explaining analysis & results | 1. Prepare script 2. Record video 3. Edit final file | 6h | Video presentation file |

As the User Stories was an exercise already made in Challenge 1, All this backlog was made according to Challenge 3 requirements for All 3 Sprints and Final Project.