

- Full name: Iván Kaleb Ramírez Torres
- NAO ID: 3357
- Date: October 4th, 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
<p>Sprint 1:</p> <p>Research and document the Google Scholar API, producing a technical report and creating a GitHub repository to manage the project.</p>	<ol style="list-style-type: none">1. Constructing a Backlog for Challenge 2.2. Build a Roadmap integrating all the requirements, dates, participants and Sprints for Challenge 2.3. Document:<ul style="list-style-type: none">- Document endpoints (API URLs)- Describe authentication methods- List query parameters- Explain response formats- Detail usage limits- Provide code examples.4. Create repository<ul style="list-style-type: none">- Add README.md (purpose, functionalities, relevance).- Upload technical report.- Configure access for Digital NAO team.
<p>Sprint 2:</p> <p>Develop Java code to perform GET requests to the Google Scholar Author API using the MVC (Model-View-Controller) design pattern</p>	<ul style="list-style-type: none">- 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

<p>Sprint 3:</p> <p>Integrate the data retrieved from the API into a database, structuring the database appropriately and handling pagination and API errors.</p>	<ul style="list-style-type: none"> - 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
<p>Final Project:</p> <p>Document Analysis & Results for Challenge 3</p>	<p>Make a video presentation explaining Analysis & Result of the Challenge 2</p>

Table 2: Prioritize list – Challenge 2

Requirements	Stages (Steps)	Time Estimation	Deliverables
Technical Report on Google Scholar API	<ul style="list-style-type: none"> - 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)	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Design model to represent author information 	2h	Java model class for Author
Java MVC – View	<ul style="list-style-type: none"> - Implement view to display author search results 	6h	Java view (console/GUI)
Java MVC – Controller	<ul style="list-style-type: none"> - Perform GET requests with HttpClient - Process API responses - Handle errors/exceptions - Update view 	2h	Java controller with working API connection
Java MVC – Integration & Testing	<ul style="list-style-type: none"> - Combine model, view, controller - Test with sample author searches 	6h	Functional Java MVC app
GitHub Repo Update (MVC)	<ul style="list-style-type: none"> - Push MVC code - Update README.md with Sprint 2 deliverables - Ensure repo permissions 	3h	Updated GitHub repo with MVC code
Database Setup	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Integrate API data for 2 researchers & 3 articles each 	5h	Functional DB with test data

	<ul style="list-style-type: none"> - Allow IDE connection wizards - Store API responses in DB 		
Error Handling & API Limits	<ul style="list-style-type: none"> - Implement robust error handling (network, API, DB) - Respect API usage restrictions 	3h	Stable DB integration with error management
GitHub Repo Update (DB)	<ul style="list-style-type: none"> - 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	<ol style="list-style-type: none"> 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.