MEDAILAB

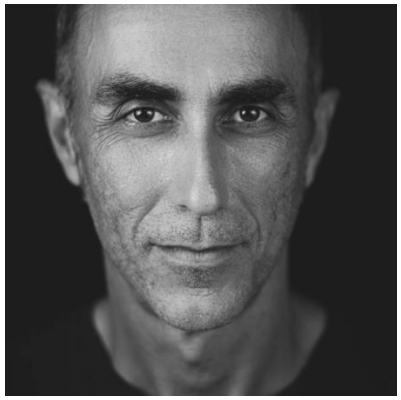
Using a Large Data Model Explorer to Maintain a Healthcare Information System

Rubén Martínez Martínez, **Francisco Javier Bermúdez Ruiz** <fjavier@um.es>
Manuel Campos Martínez, José Manuel Juárez Herrero

28th OCTOBER 2025

Medical AI Laboratory (MEDAILAB)
University of Murcia

Francisco Javier Bermúdez Ruiz, Ph.D.



Dissertation: *An approach for model data reengineering*

"Expertise in **Model-Driven Engineering, Data Modelling** and **Data Reengineering**"

Some papers:

- Developing a Model-Driven Reenginering Approach for Migrating PL/SQL Triggers to Java: A Practical Experience (Journal of Systems and Software). DOI: 10.1016/j.jss.2019.01.068
- On the application of Model-Driven Engineering in Data Reengineering (Information Systems). DOI: 10.1016/j.is.2017.10.004
- A tool to support the definition and enactment of model-driven migration processes (Journal of Systems and Software). DOI: 10.1016/j.jss.2017.03.009

MED AI LAB

MedAI Lab

Research in the field of **Medical Informatics** and the use of **Artificial Intelligence**

Main areas of work: **clinical information systems, medical decision support tools, software design for hospital management,** and **clinical data analysis**.

Agenda

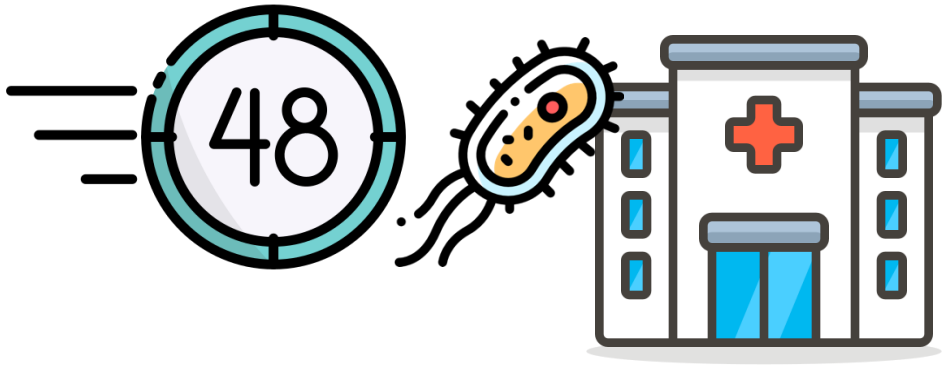
- Introduction
- Related Work
- Proposed Solution
  - Functionalities
  - Architecture
  - JPA Extraction and Storing
  - Interactive Model Visualization
  - User Experience Enhancements
- Evaluation
- Conclusion and Future Works

Introduction (I)

Clinical Context: Hospital-Acquired Infections

Hospital-acquired infections (HAIs), also known as **nosocomial infections**, are a major public health concern.

They appear 48 hours after hospital admission and are often caused by **multidrug-resistant bacteria**.



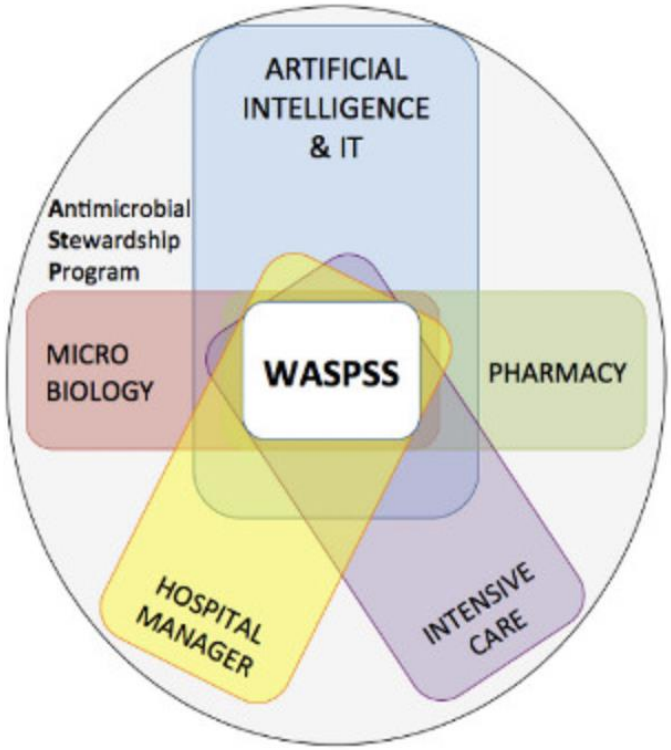
Antibiotic misuse contributes to resistance, limiting therapeutic options and increasing morbidity, mortality, and hospital costs.

Introduction (II)
The WASPSS Tool

WASPSS tool (Wise Antimicrobial Stewardship Program Support System) was developed to support **antimicrobial stewardship programs**.

Created by the **University of Murcia** and **Getafe University Hospital**, first implementation in 2015.

Since then, it has expanded to hospitals in several Spanish regions, including the **Basque Country** and **Murcia Health Service**.



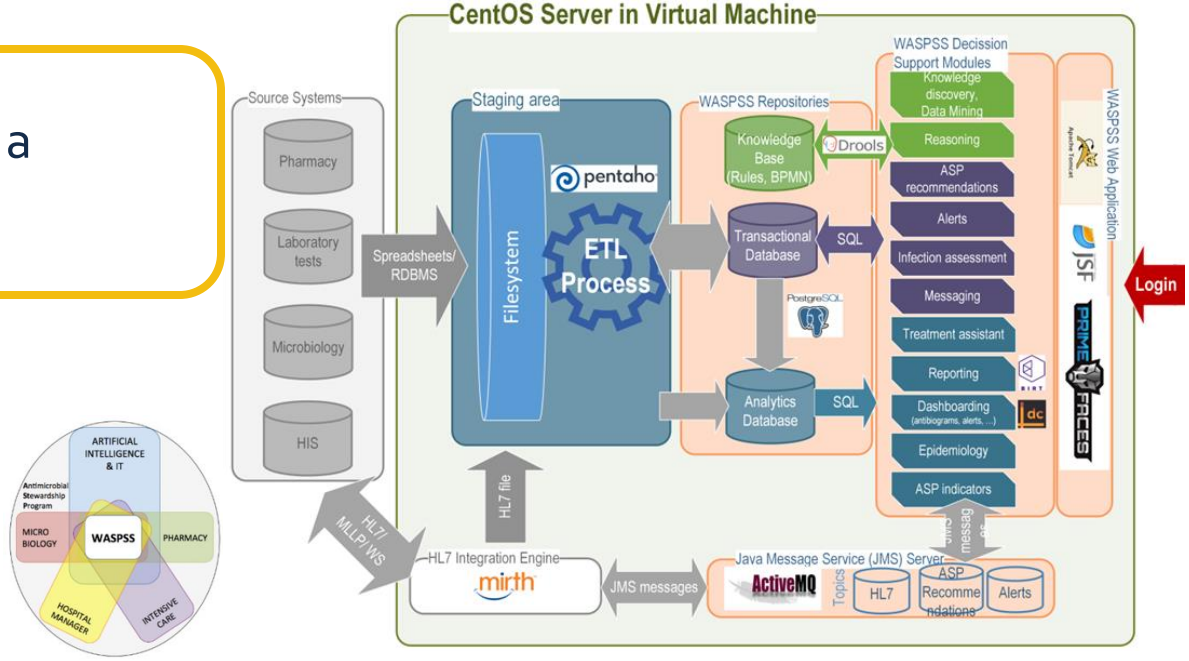
Introduction (III)

WASPSS: System Architecture and Technologies

WASPSS is a **web application** built using **Java technologies** and the basic **Model-View-Controller** pattern.

Integration with hospital systems via **HL7** and uses **Drools** for clinical rule processing and decision support.

Persistence relies on **JPA** and a **PostgreSQL database**.



Introduction (IV)
Technical Challenges

After more than **12 years of development**, the codebase has **grown** considerably, but the **documentation** has **not** kept pace.

The system now includes **over 1,000 Java classes**, **200 JPA entities**, and **166 database tables** → maintenance difficult.

The **lack** of updated **documentation complicates** onboarding, bug fixing, and knowledge transfer among **developers**.

Existing market tools **fail** to visualize such **large-scale data models**
→ diagrams become messy and unreadable

Introduction (V)

Proposed Solution and Contributions

The **domain model** is the **core** of the software, representing the essential business knowledge, rules, and processes.

This work proposes a **methodology and tool** for **exploring large**, **poorly documented data models** in healthcare systems.

- Tool provides **progressive visualization** of model (entities, attributes, and relationships), **propagating inheritance** information for clarity.
- Developers can **incrementally explore** and **understand** the model through **customizable** graph-based visualizations.

Introduction (VI)
Proposed Solution and Contributions

Research is guided by questions about: **improving comprehension**, **required functionalities**, and **perceived usability** of such tools.

The main contributions:

The **exploration methodology**
The **supporting visualization tool**
The **validation by end users**

The tool for improving:

Maintainability
Developer onboarding
Software quality

…in complex **healthcare systems**

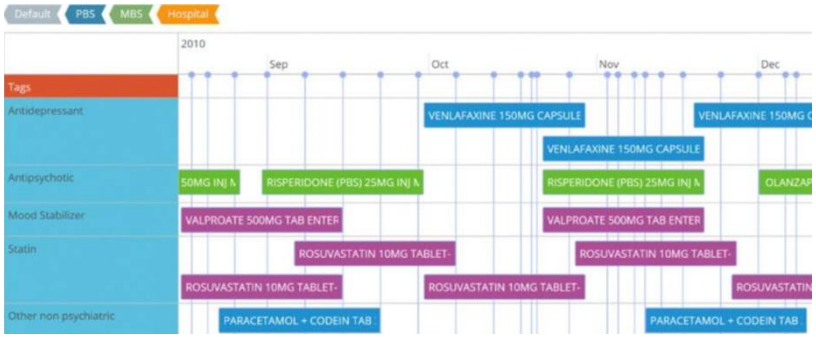
Related Work (I)

Interactive Visualization in Healthcare Systems

Prior research explored **interactive visualization** to manage complex healthcare systems.

Owlready enabled **incremental exploration** and **classification** of biomedical ontologies.

Health Timeline improved understanding of patient records through **progressive visualization**.



Other tools highlighted the value of **dynamic filtering** and **navigation** in large datasets.

These works support **user-centered and incremental visualization** → extended in our **JPA model visualizer** for large healthcare data models.

Related Work (II)

Visualization Tools for JPA Models

Large **JPA models** in Java are difficult to interpret directly from code.

Tools like **Eclipse Dali**, **Hibernate Tools**, **Jeddict**, **IntelliJ**, and **JPA Buddy** offer partial visualization support.

Generic tools (e.g., **DBeaver**, **SchemaSpy**) show database diagrams but ignore JPA details.

- None support **incremental exploration** or **inheritance propagation**.
- Most lack **partial view saving** and **interactive editing**, gaps our tool addresses.

Proposed Solution (I)

Support tool that offers developers (a new methodology) that **progressively and interactively explore** an application data model.

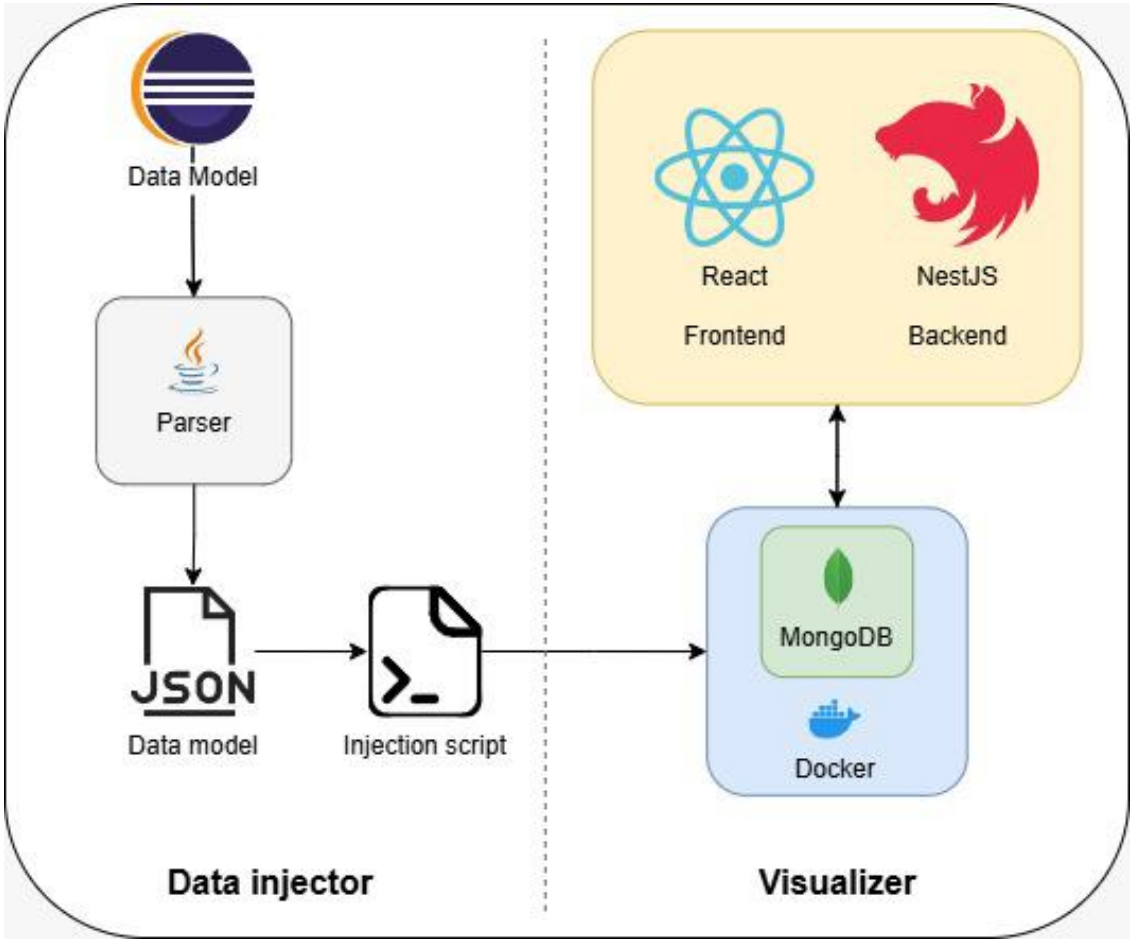
- **Dynamic visualization** → adapting to the entities displayed on screen.
- **Inheritance behavior** support → subclasses automatically inherit attributes and relationships from parent classes.
- **Own and inherited information** available → shown depending on what the user explores.
- **UML-like graphical diagram** → represent entities, attributes, and relationships.
- To improve clarity, **methods are omitted**, focusing only on the **structural aspects** of the model.

Proposed Solution (II) – Functionalities

Progressive and interactive exploration of models: entities, relationships, and inheritance.

- Users can **add or remove entities dynamically**, updating inherited information in real time.
- **Saving and loading partial model views** for focused analysis.
- Entities can be **tagged and filtered** to display specific subsets of the model.
- Its **modular → domain-independent design** enables reuse beyond WASPSS, in other medical systems.

Proposed Solution (III) – Architecture



Proposed Solution (IV) -Model Extraction and JSON Generation

The solution addresses the need to **visually represent complex data models**.

A **parser** analyzes the project source code to extract **JPA entities, inheritance, and relationships**.

The extracted data is stored as **JSON files**.

Using JSON as an **intermediate format** makes the tool **independent from WASPSS** and **reusable** in other systems.

This approach ensures **flexibility, scalability, and reusability** across different domains.

Proposed Solution (V) – Database Integration and Visualization

The **JSON model** is imported into a **MongoDB database**.

MongoDB → **native handling of JSON documents**.

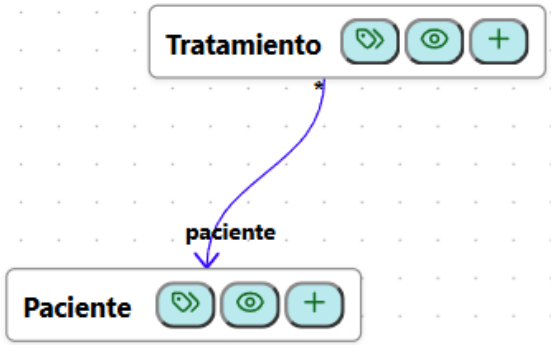
The database stores both the **data model** and the **interactive visualization information** (views and tags).

The **persistence core** of the system → ensuring consistency during user interactions.

Proposed Solution (VI) – Interactive Model Visualization

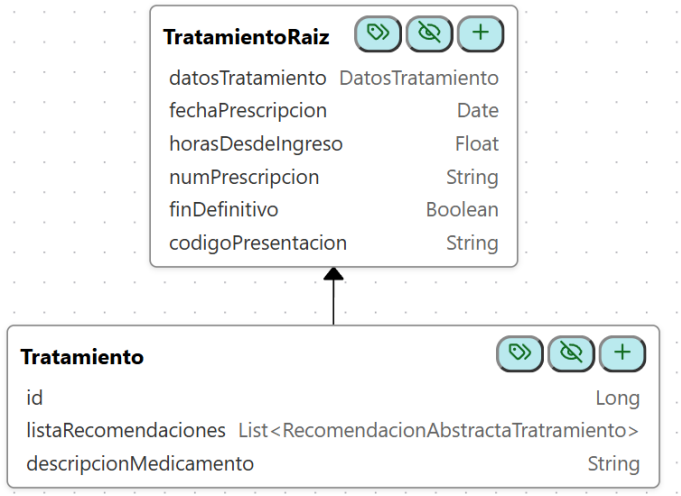
Core of the solution → a **graphical interface** for **interactive exploration** of JPA entities.

Entities are displayed as **nodes**, and relationships appear as **directed edges** in a dynamic diagram.



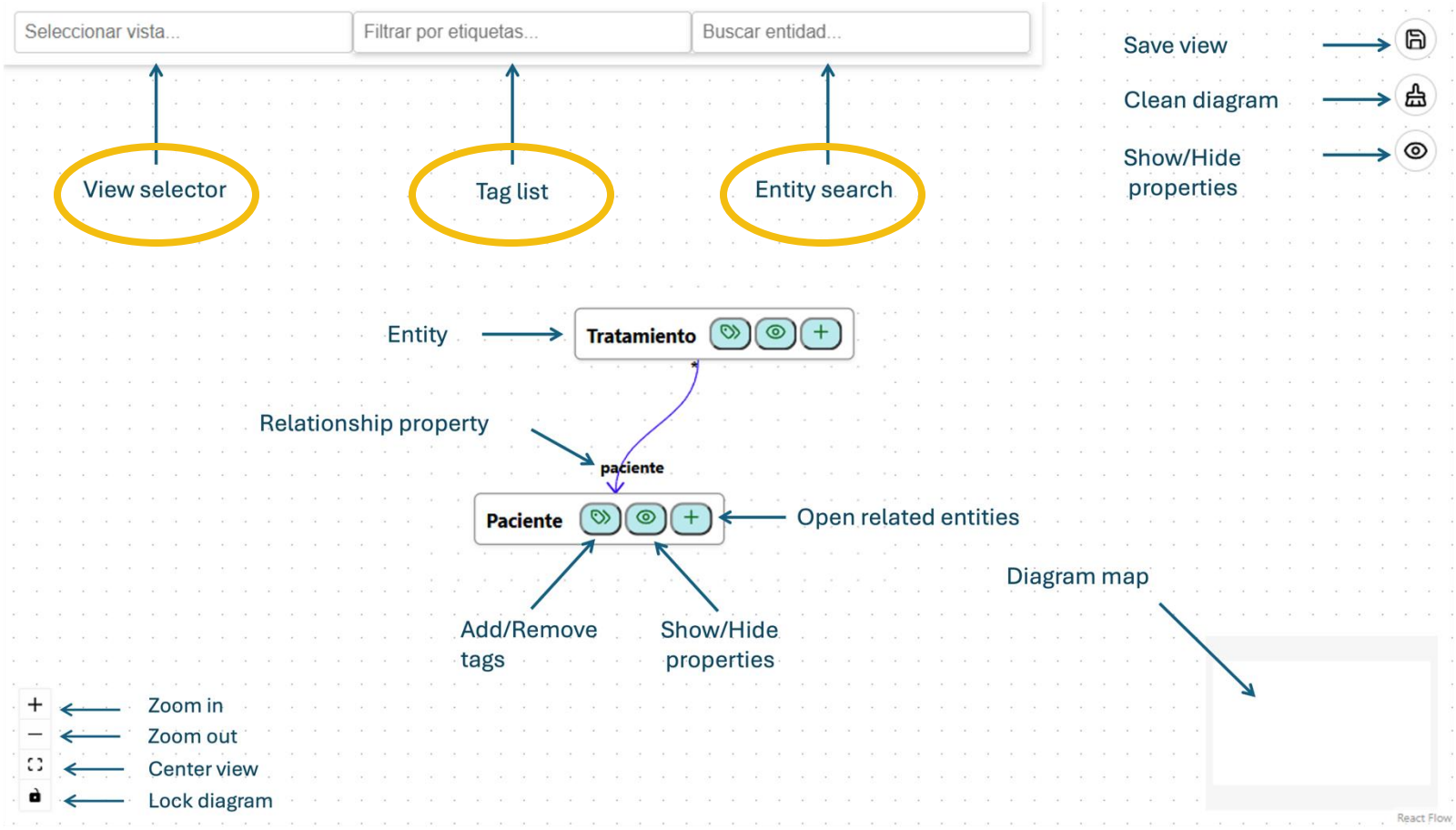
Users can **save partial views**, **tag entities**, and **filter** the model for easier navigation.

The visualizer automatically manages **inheritance and relationship consistency** in the diagram.



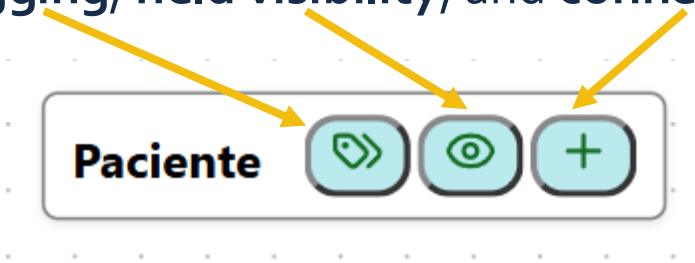
Proposed Solution (VII) – Interactive Model Visualization

View Selector, **Tag List**, and **Entity Search** to manage and locate entities.



Proposed Solution (VIII) – Interface Controls and Node Features

The workspace shows entities as rectangles with buttons for: **tagging**, **field visibility**, and **connections**.



Tag Button → manage tags and sync them with the backend.

Visibility Button → show/hide fields in the entity.

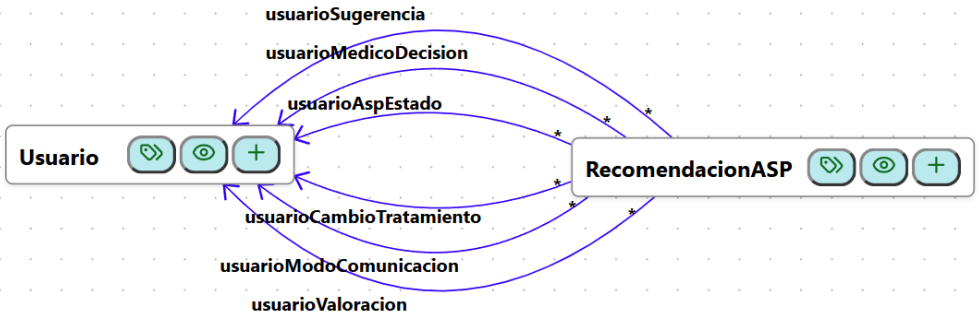
Connection Button → enables adding related entities—visible, inherited, or ancestor—allowing **progressive model discovery**.

Proposed Solution (IX) – User Experience Enhancements

Auto-focus in the **tag creation modal** for faster input.

Dropdown scrolling does not affect canvas zoom.

Relationships (simple, multiple, reflexive) are rendered with **clear labels and cardinalities**, using **custom algorithms**.

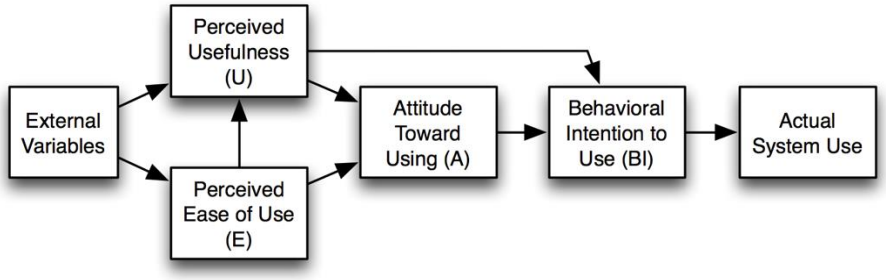


Partial views can be saved, preserving entity positions for consistent layout.

Tag-based filtering instantly retrieves entities with specific tags.

Tags can be **assigned or created** dynamically via node icons, with validation to **prevent duplicates**.

Evaluation (I)



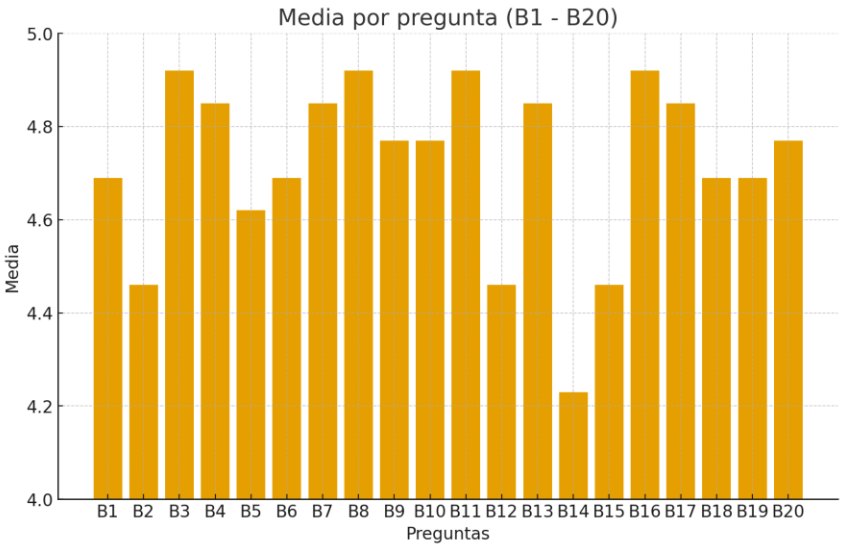
Tool validated using the **Technology Acceptance Model (TAM).**

TAM measures **perceived usefulness** and **perceived ease of use.**

Participants completed a **practical exercise** and a **questionnaire** (20 questions) covering: demographics, ease of use, usefulness, attitude, and intention to use.

13 faculty members and staff from the **University of Murcia** participated.

Questions	Mean	Stand. Dev.	Questions	Mean	Stand. Dev.
B1	4.69	0.48	B11	4.92	0.28
B2	4.46	0.52	B12	4.46	0.88
B3	4.92	0.28	B13	4.85	0.55
B4	4.85	0.38	B14	4.23	1.09
B5	4.62	0.51	B15	4.46	0.66
B6	4.69	0.63	B16	4.92	0.28
B7	4.85	0.38	B17	4.85	0.38
B8	4.92	0.28	B18	4.69	0.63
B9	4.77	0.60	B19	4.69	0.48
B10	4.77	0.44	B20	4.77	0.44



Evaluation (II)

Perceived Ease of Use: mean score 4.75 – users found the visualizer **easy and intuitive**.

Perceived Usefulness: mean score 4.66 – participants viewed the tool as **functional and valuable**.

Attitude Toward Use: mean score 4.77 – users believed using the visualizer is a **good idea**.

Intention to Use: mean score 4.73 – strong willingness to **use in real-world contexts**.

Overall, the visualizer shows **high levels of user acceptance**.

Conclusions and Future Works

Presented a **methodology and tool** for exploring and maintaining **large, poorly documented healthcare data models**.

Applied to **WASPSS**, enabling **incremental and inheritance-aware visualization** of complex JPA domain models.

Supports **interactive exploration**, **custom views**, and **tag-based filtering**, improving **comprehension and maintainability**.

Modular architecture ensures reusability in other **object-oriented medical systems**.

User evaluation (TAM) confirms the tool is **intuitive and effective** in real-world use.

Future work: add **collaborative features**, **model editing**, and **integration with automated documentation**, plus broader evaluation including industry professionals.


THANK
YOU