

## Título

### IR-Board: Requirements management platform

## Resumen

Se pretende crear una herramienta que permita dar soporte a los procesos involucrados en la gestión de requisitos: gestión de fuentes e interesados, obtención y refinado de requisitos permitiendo la comprobación de los atributos de calidad, y dando soporte a distintas estándares y normas (guías del IEEE 830 y el estándar de la ISO 29148) Se incorporarán facilidades para incluir información de técnicas estáticas y dinámicas (como casos de uso, diagramas de flujo y tablas de decisión) y la aplicación de metodologías ágiles como las historias de usuario.

## Decisiones arquitectónicas

- Debido a petición de los tutores, se decidió un sistema web sobre standalone.

Se elige React con Material UI debido a su conocimiento previo, además de una gran versatilidad y enfoque destinado a portabilidad y adaptabilidad. Se prefiere sobre la madurez de otros frameworks como Angular, o sobre bibliotecas de componentes como Ant Design o TailwindCss debido a dicho enfoque y simplificación del desarrollo.

- Se elige SpringBoot sobre Express.js por el entorno más seguro por defecto, y valorando que ambos tienen una buena integración con la tecnología de frontend elegida.
- A fin de facilitar su implantación en otros sistemas, se decide separar el frontend y backend en proyectos separados para facilitar su adaptación a otros entornos.
- Arquitectura web simulando trabajo colaborativo similar al word con los mutex a nivel de párrafo = requisito
- La asociación entre elementos a fin de marcar como pendientes de revisar debe ser bidireccional, y de forma que el patrón observer debe estar unido para cada entidad. Para evitar bucles infinitos, se puede hacer una revisión por separado de una modificación, que a efectos, aplica el mutex de la misma manera, y o bien valida o modifica aquello marcado. Se ha decidido ante la visión que, aunque se pueda entrar en un bucle de modificar todo, forma parte del proceso de refinado, y es en esencia, una buena práctica.
- Arquitectura de microservicios – Traefik - Gateway (ya tengo una mínima experiencia con este de ASR) – React - frontend – Ory Kratos - Identification & Sessions – Ory Oathkeeper - Enforcement – springboot - backend con el RMS – PostgreSQL - BD Identidad para Ory Kratos – PostgreSQL - BD del RMS – Loki - Logging centralizado – Promtail - Agente que recolecta los logs para Loki – Grafana - Dashboard para métricas y logs de Loki
- Consideraciones de la arquitectura a indicar que valoré pero descarté  
Ory Hydra - (unneeded because I'm not an identity provider) Ory Keto - Authorization (unneeded because the role groups are simple) gateways Kong y Typhoon.
- Elegí microservicios por una mayor escalabilidad e integración más sencilla con servicios existentes.
- Elegí hacer uso de sistemas ya probados de seguridad modulares, como Ory, así como la gateway de traefik, debido a la complejidad, trabajo añadido y probablemente menor eficiencia que tendría crearlos por mi cuenta. Asimismo, evitarlo me asegura un proyecto más sólido, con más facilidad de añadir nuevas funcionalidades, más seguro, y más eficiente.

- Elección de postgres como base de datos relacional debido a que el dominio es estable, se trata de requisitos y datos a guardar considerados estándar, y debido a su integración preferida por Ory. Además, permite usar columnas JSONB, que ofrecen la flexibilidad de bases de datos NoSQL para tener atributos personalizados. Por último, tiene una buena integración con un driver de mucha madurez con springboot.
- Elegí separar en dos redes la arquitectura para dar una mayor seguridad a la arquitectura siguiendo el principio de menor privilegio.
- 2026/01/31 I decided that when flagging an element, it will have a `is_suspect` boolean value, and a `impact_summary` field. Whenever a new flagging occurs, it will simply check the flag and set it if it is false, and append why to the `impact_summary`. Reviewing it removes everything in its `impact_summary` field.
- 2026/01/31 I decided that to ensure natural ordering for a better UX, the identifiers are calculated on the frontend virtually. To ensure a way to order, the entities will have both a UUID and a `sort_order` field, with float values. If an element were to be placed between element 1 and 2, its `sort_order` would be its midpoint: 1.5. To ensure a problem would not happen, if the difference between two numbers is less than a constant, a re-normalization would occur, iterating over all elements and setting their `sort_order` to their integer index.
- 2026/02/02 I decided that the project, before a baseline is saved, will allow reordering of requirements, which will make requirements have a dynamic semantic ID (FR-UM-1.5 -> functional requirement user management 1.5). This ensures a better UX until it is reasonable that the project is past its inception, and then will lock the semantic IDs once the first baseline is set. Any semantic ID used before, even if deactivated, won't be used again, ensuring compliance with ISO 29148 and its traceability.
- 2026/02/02 I decided that in order to comply with ISO 29148, the system can't allow any user to delete anything, but rather deactivate it.
- 2026/02/05 Along with my tutors, we decided to always allow reordering of entities, with the corresponding modification of the entity's dynamic id. This was done to prioritize UX. To comply with ISO 29148's traceability, it will be implemented via the internal UUID; To ensure users can follow this traceability, the system will show both the dynamically generated id and the internal UUID of the entities, and the system will allow to search by UUID.
- 2026/02/05 Along with my tutors, we decided to add a set of states for all entities of a project, as suggested by the lifecycle of an element on ISO 29148. The flagging as pending a review will be absorbed into this system.
- 2026/02/05 Along with my tutors, we decided that for now, all documents linked to a project are global and accessible by those linked to said project. Access control for these elements falls beyond the scope of this project.
- 2026/02/05 Along with my tutors, we decided that the linking will be first via a context menu, and if the project advances at a sufficiently rapid pace, implement a markdown style linking. The linking between elements will be unidirectional after all, as linking a requirement to another does not imply that the observed requirement is affected by the observer, and similarly with linking to a stakeholder or to a document, the recipient of the observation is most of the times, unlikely to change, and when it does is when one would be interested in being notified.

- 2026/02/06 requirement engineer will have access via linking with read or write access to each functionality on a project, not level access. This is done to avoid breaking the workflow due to a req. engineer needing to modify a parent and not being able to do so.

–2026/02/06 las dependencias a usar en el backend de springboot son Lombok, Spring Web, Spring Boot DevTools, H2 Database, Spring Data JPA, PostgreSQL Driver, y Validation.

## Functional requirements of the system

### project management

- The system must allow an admin to create a project
  - The following are required:
    - Project name
    - Description
    - Owner
- The system must allow an admin to deactivate a project
  - The system must put the project on read only mode for those linked to it
- The system must allow an admin to reactivate a deactivated project
- The system must allow an admin to modify an active project
  - The system must forbid other users from modifying the entity
    - The system must release automatically the entity upon saving
    - The system must release automatically the entity after a predetermined timeout period
    - The system must release automatically the entity if the user editing it modifies another entity
    - The system must only accept changes to the entity from the user who holds the entity
  - The system must display for other users who is modifying the entity
- The system must allow to link users with a project
  - The system must allow an admin to link users to a project as project manager
  - The system must allow an admin or project manager linked to the project to link users to a functionality on said project as a stakeholder user
  - The system must allow a project manager to link users to one or more functionalities of a project as requirement engineers.
- The system must allow access to the project description/dashboard to users linked to it or a functionality of it.
  - The system must show the total split of requirements by their states (pie chart)
  - The system must show the different functionalities of the project
- The system must allow a project manager to add a functionality to a project
  - A functionality needs a name and unique set of letters for its dynamic identifier.
  - The system must automatically attempt to get the letters for the dynamic identifier from the name
    - The system must take the first letter from every word in the name.
    - If the identifier is already in use by another functionality on the same project, the system will suggest one letter more of each word on the name.
    - If the system cannot generate a new set of letters to identify its requirements, a message must be shown to the project manager.
  - The system must check the letters for the identifier are not the same from another functionality on the same project.
    - The system must deny adding a functionality that breaks the rule above.
  - The system must automatically link the project manager to the new functionality
- The system must allow a project manager to modify a functionality.

- The system must forbid other users from modifying the entity
  - The system must release automatically the entity upon saving
  - The system must release automatically the entity after a predetermined timeout period
  - The system must release automatically the entity if the user editing it modifies another entity
  - The system must only accept changes to the entity from the user who holds the entity
- The system must display for other users who is modifying the entity
- The system must allow a project manager to deactivate a functionality.
- The system must allow a project manager to reactivate a functionality.
- The system must allow a project manager to generate a baseline for a project.
  - The system must perform a snapshot of the project once a baseline is set.
- The system must allow a project manager to export the project's requirements onto a pdf file

## **Stakeholders management**

- The system must allow any user linked with the project access to view its stakeholders
  - The system must show if a stakeholder is flagged as pending review
  - The system must show the identifier, the name and part of the description
  - The system must allow to collapse and expand stakeholders with children
  - The system must allow to view the details of a stakeholder
    - The system must show:
      - All attributes of a stakeholder
      - All requirements linked to it
- The system must allow a project manager to add a new stakeholder to a project
  - The system must only allow a stakeholder to be added to a project the user is linked to.
  - The system must ask for the following:
    - Name
    - Description
  - The system must generate the identifier for the stakeholder
- The system must allow a project manager or requirement engineer to link a stakeholder to one or more requirements on the same project
  - The system must only allow the user to link the stakeholder to a requirement on a functionality they are linked to
- The system must allow a project manager or requirement engineer to unlink a stakeholder from one or more requirements
  - The system must only allow the user to unlink a stakeholder from a requirement of a functionality they are linked to.
- The system must allow a project manager to deactivate a stakeholder from a project the user is linked to
  - The system must show the user the amount of entities affected by the deactivations
  - The system must ask for confirmation before deactivating
  - The system must flag all entities linked as pending review
- The system must allow a project manager or requirement engineer to modify a stakeholder
  - The system must forbid other users from modifying the entity
    - The system must release automatically the entity upon saving
    - The system must release automatically the entity after a predetermined timeout period
    - The system must release automatically the entity if the user editing it modifies another entity
    - The system must only accept changes to the entity from the user who holds the entity
  - The system must display for other users who is modifying the entity
  - The system must flag as pending review linked entities upon saving with changes.

## Requirement management

- The system must allow users linked to a project or functionality of a project access to its requirements
  - The system must only allow users linked to the functionality of a functional requirement access to it
  - The system must allow to collapse and expand requirements with children
  - The system must show the dynamic identifier, the name, state and part of the description
    - The system must show as “state” on a non-functional requirement whether it is passed or not
  - The system must allow to view the details of a requirement
    - The system must show:
      - The internal unique identifier
      - All attributes of a requirement
      - All stakeholders linked to it
      - All requirements cross-linked with it
      - All documents linked to it
      - The previous dynamic identifiers
  - The system must allow users to filter the requirements they have access to
    - The system must allow searching by internal unique identifier.
    - The system must allow users to filter the requirements on a functionality they are linked to
    - The system must allow users to filter the non-functional requirements of a project they are linked to
    - The system must allow filtering the requirements based on MOSCOW priority
    - The system must allow filtering the requirements based on state
    - (future work) The system must allow a user to search requirements by text
- The system must allow a requirement engineer or a project manager to add a requirement to a project the user is linked to
  - The system must only allow a functional requirement to be added to a functionality the user is linked to.
  - The system must allow the user to generate a requirement as a child of another requirement (nesting).
  - The system must assign automatically the dynamic identifier
    - The identifier must be based on its relation to other requirements.
    - The identifier must represent if it is a functional or non functional requirement (FR or NFR)
    - The identifier must represent the folder/component that holds the requirement (user management -> UM)
  - The system must assign automatically the internal unique identifier
    - The identifier must represent in some way the project that will hold the requirement
    - The identifier must represent in some way whether the requirement is functional or non functional
    - The identifier must have a random element to ensure a low collision rate
  - The system must ask for the following data for a functional requirement:
    - The following are required:
      - Name
      - Description
      - a priority
        - The system must allow MOSCOW categories (either Must, Should, Could or Won't have)
        - The system must allow to set a priority value from 1-5 with the MOSCOW value
    - The following are not required:

- Stability
- Origin
- The system must ask for the following data for a non functional requirement:
  - The following are required:
    - Name
    - Description
    - Measurement unit
    - comparison operator
      - equal to, less than or greater than
    - Threshold value
      - This value represents the minimum value to mark the requirement as passed
    - Target value
      - This value represents the optimal value desired by the team
    - Actual value
      - This value represents the current status of the measurement
- The system must allow a project manager or requirement engineer to link a requirement on a functionality they are linked to, to another entity
  - The system must allow to link a requirement with a stakeholder of the same project
  - The system must allow to un-link a requirement with a stakeholder
  - The system must allow to link a requirement with one or more requirements of functionalities of the same project the user is linked to
  - The system must allow to un-link a requirement with other requirements of functionalities of the same project the user is linked to
  - The system must allow to link a requirement with one or more documents of the same project.
  - The system must allow to un-link a requirement with one or more documents of the same project.
- The system must allow a requirement engineer or a project manager to deactivate a functional requirement on a functionality they are linked to
  - The system must show the user the amount of entities that will be affected by the deactivation
  - The system must ask for confirmation
  - The system must flag any requirements linked to the deactivated requirement as pending review
- The system must allow a requirement engineer or a project manager to reactivate a requirement on a functionality they are linked to
  - The system must show the user the amount of entities that will be affected by the reactivation
  - The system must ask for confirmation
  - The system must flag any requirements linked to the reactivated requirement as pending review
- The system must allow a requirement engineer or a project manager to modify a requirement on a project
  - The system must only allow a project manager or requirement engineer to modify functional requirements of a functionality the user is linked to.
  - The system must forbid other users from modifying the entity
    - The system must release automatically the entity upon saving
    - The system must release automatically the entity after a predetermined timeout period
    - The system must release automatically the entity if the user editing it modifies another entity
    - The system must only accept changes to the entity from the user who holds the entity
  - The system must display for other users who is modifying the entity
  - The system must flag linked requirements as pending review upon saving with changes.
- The system must allow a project manager or requirement engineer linked to a functionality of the project to change the position of a requirement

- The system must allow reordering of functional requirements to users linked to the same functionality.
- The system must update the dynamic identifier automatically
- The system must set the order of requirements using a floating point order value
- The system must allow a project manager or requirement engineer to review an requirement flagged as pending a review
  - The system must allow removing the flag if no changes are required.
  - The system must allow modifying the requirement upon review.
    - The system must remove the flag upon saving with changes.
    - The system must flag the linked entities as pending a review.

## **User management**

- The system must allow an admin to invite new users to the system
  - The system must provide different levels of authorisation.
    - The system must have the levels: Admin, project manager, requirement engineer and stakeholder user
  - The system must ask the admin to set the name, surname, and email of the invited user
    - The system must generate an signup code as a temporal password
    - The system must automatically send an invitation with the signup code to the email of the invited user
- The system must allow an admin to modify the name and surname of a user from the system
- The system must allow an admin to generate a new invite with a signup code for a user
- The system must allow any user with valid credentials to sign in to the system
  - The system must prompt any user signing in with a signup code to set a permanent password.
    - The system must ensure the password is between 15 and 64 characters long.
    - The system must make use of a random salt specific of each user.
    - The system must remove any password or signup code of the user upon setting a permanent password.
  - The system must temporally block the user after 3 consecutive failed attempts
- The system must allow an admin to deactivate a user from the system
- The system must allow an admin to reactivate a user from the system

## **Document management and modelling**

- The system must allow users linked to a project access to documents of that project
  - The system must show entities linked to the document.
- The system must allow a project manager or a requirement engineer to add document to a project
  - The user must be linked to the project
- The system must allow a document to be linked to one or more requirements of the same project
  - The system must flag those requirements linked to it as pending a review if the document is altered
- The system must allow a project manager or requirement engineer to update a document
  - The user must be linked to the project the document is on.
  - The system must flag as pending a review any requirements linked to the document
- The system must allow a project manager to disable a document
  - The system must flag as pending a review any requirements linked to the document
- The system must allow a requirement engineer to model diagrams using a Draw.io integration
- (future work) The system must allow a requirement engineer fill a tabular use case
  - The system must have the following fields:
    - Name

- Description
- Actors
- Initial Condition
- Preconditions
- Postconditions
- Normal flow
- Alternative flows
- Exceptions
- (future work) The system must allow a requirement engineer fill a scenario
  - The system must have the following fields:
    - Name
    - Description
    - Actors
    - Initial Condition
    - Preconditions
    - Postconditions
    - Flow

## **Variant control**

Highly ambitious—only if time permits. Keep this in mind even if not implemented: requirement reuse. Defining templates or abstract projects. In the industry, “copy and paste” is the standard approach.

## **Requisitos no funcionales**

- RN-01 Persistencia Normativa: La base de datos debe permitir la trazabilidad exigida por la ISO 29148 (id, descripción, prioridad, estado, origen, justificación).
- RN-02 Control de Concurrencia: El sistema debe implementar un mecanismo de bloqueo a nivel de requisito (mutex) para evitar condiciones de carrera durante la edición simultánea.
- RN-03 Disponibilidad y Web: Acceso multiplataforma mediante navegador, garantizando un diseño responsive gracias a Material UI.