Backlog Quarto

Table of contents

1 Observaciones actuales

- El build histórico versionado en `frontend/_site` ocultaba fallas en despliegue. Se migró a un build reproducible que se ejecuta en CI y durante las pruebas.
- Varias rutas (`/catalogo/graficos/`, `/docs/*`, `/proyecto/`, `/api/status`) respondían 404 al no publicar los artefactos generados.
- Las fichas de dataset dependían de datos parciales (`summary.map`) y fallaban cuando el JSON no se convertía correctamente en `Map`.
- Las tarjetas de métricas mantienen una animación que interfiere con la legibilidad y no existen pruebas visuales que detecten regresiones de layout.
- Algunos assets (PDF, DOCX) no se generan en entornos intermedios, lo que produce enlaces rotos.

2 Acciones inmediatas

1. **Pipeline de build reproducible** (`scripts/build_frontend_site.py`) — ejecutar en CI (`quarto render --to html`) antes de los tests y en deploy para producir los artefactos completos. Referencia: Quarto CI.
2. **Normalizar el resumen de catálogos** usando presenters tolerantes a entradas planas para evitar `TypeError` en producción.

3. **Verificar rutas críticas** con pruebas automatizadas (homepage, catálogo, documentos, `/api/docs`, `/api/status`, `/catalogo/graficos/`).
4. **Ajustar layout responsive** siguiendo las guías de article layout y dashboards layout para tarjetas y grids.

3 Backlog temático

3.1 Layout y tipografía

- Revisar los anchos máximos, márgenes y jerarquía tipográfica conforme a page layout e incluir pruebas de screenshot para componentes críticos.
- Documentar y refinar la composición de tarjetas y métricas alineado con data display en dashboards.
- Implementar callouts y bloques informativos reutilizables basados en Authoring Callouts.

3.2 Contenido interactivo

- Migrar gráficos y widgets a componentes declarativos compatibles con Quarto Dashboards y Interactivity.
- Evaluar integración con Shiny for Python para métricas en tiempo real.

3.3 Documentación y navegación

- Auditar la navegación del sitio y menús laterales conforme a Website Navigation y Website Tools.
- Implementar listados y fichas enriquecidas siguiendo Website Listings.
- Revisar páginas `docs/*` para asegurar consistencia con Website About y Website Search.

3.4 Automatización y entornos

- Definir perfiles específicos (`_quarto-profiles.yml`) para `ci`, `preview` y `production`, y documentar las variables en línea con Projects & Profiles y Environment management.
- Integrar checks de lint con `quarto check --quiet` y pruebas unitarias de render conforme a Quarto Scripts.
- Investigar ejecución reproducible con Binder y Virtual Environments.

3.5 Contenido técnico

- Añadir diagramas generados mediante Authoring Diagrams y tablas alineadas con Authoring Tables.
- Centralizar variables reutilizables usando Authoring Variables y condiciones (p.ej. entornos) con Authoring Conditional Content.
- Revisar snippets de código e incluir ejemplos ejecutables basados en Code Execution.

3.6 Seguimiento

- Consolidar métricas y estado de pipelines en `/docs/` siguiendo Dashboard Deployment.
- Documentar flujos de publicación y mantenimiento (CI/CD, seeds) tomando como referencia Websites y Projects.