



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN TECNOLOXÍAS DA INFORMACIÓN

Aplicación SIX-web para proxectos colaborativos de participación pública

Estudante: Adrián Moledo Romaní
Dirección: Jorge Parapar López
Juan Touriño Domínguez

A Coruña, setembro de 2020.

À miãa familia

Agradecementos

Aos meus directores polo seu tempo e axuda.

Aos meus compañeiros e amigos que me axudaron durante todo este tempo e que me permitiu chegar ata aquí.

Á miña familia polo seu apoio durante estes anos.

Resumo

Neste traballo desenvólvese unha aplicación web para a toma de datos en procesos de participación pública nos que a información introducida polos participantes deba estar xeoreferenciada – Public Participation Geographic Information System (PPGIS) [1]. A aplicación intégrase cun servidor de información xeográfica vía web, facendo uso dos protocolos Web Map Service (WMS) e Web Feature Service (WFS). A aplicación web e o servidor xeográfico empregan unha base de datos común con capacidades para a xestión de información espacial (PostgreSQL [2]-PostGIS [3]). A característica principal do sistema son as capacidades de personalización implementadas, que permiten aos administradores configurar os tipos de elementos sobre os que se quere tomar datos, e os atributos dos mesmos, dende a propia interface de usuario da aplicación. Os administradores tamén poden configurar algúns aspectos da interface de usuario dende a propia aplicación. Os principais *frameworks* empregados para o desenvolvemento son React [4], OpenLayers [5] e Spring Boot [6]. Como servidor de información xeográfica empregouse QGIS Server [7]. En xeral, todos os compoñentes empregados son de código aberto.

Abstract

In this work, a web application was developed for data collection in public participation processes in which the information entered by the participants must be georeferenced – Public Participation Geographic Information System (PPGIS) [1]. The application is integrated with a geographic information server via web, making use of the Web Map Service (WMS) and Web Feature Service (WFS) protocols. The web application and the geographic server use a common database with spatial information management capabilities (PostgreSQL [2]-PostGIS [3]). The main feature of the system are the customization capabilities implemented, which allow administrators to configure the types of elements on which they want to collect data and their attributes. Administrators can also configure some aspects of the user interface from within the application itself. The main *frameworks* used for development are React [4], OpenLayers [5] and Spring Boot [6]. QGIS Server [7] was used as the geographic information server. In general, all the components used are open source.

Palabras clave:

- SIX de participación pública
- PPGIS
- SIX
- Java
- React
- REST
- QGIS Server
- Spring Boot
- PostGIS
- OpenLayers
- WMS
- WFS

Keywords:

- Public participation GIS
- PPGIS
- GIS
- Java
- React
- REST
- QGIS Server
- Spring Boot
- PostGIS
- OpenLayers
- WMS
- WFS

Índice Xeral

1	Introdución	1
1.1	Motivación	1
1.2	Obxectivos	1
1.3	Traballos relacionados	2
1.4	Estrutura da memoria	3
2	Tecnoloxías, ferramentas e librarías	5
2.1	Linguaxes de programación	5
2.1.1	Backend	5
2.1.2	Frontend	5
2.2	Frameworks e librarías	5
2.2.1	Backend	5
2.2.2	Frontend	7
2.2.3	Ferramentas para probas	8
2.3	Ferramentas de desenvolvemento	9
2.3.1	Maven	9
2.3.2	Npm	9
2.3.3	Git	9
2.3.4	Redmine	10
2.3.5	QGIS	10
2.3.6	Eclipse	10
2.3.7	Visual Studio Code	11
2.3.8	draw.io	11
2.3.9	Overleaf	11
2.4	Gateway HTTP	11
2.5	Servidores de aplicacións	11
2.6	Servidor xeográfico	12

2.7	Sistema xestor de bases de datos	12
3	Arquitectura global do sistema	13
3.1	Compoñentes do sistema	13
3.2	Interacción entre os compoñentes	14
4	Metodoloxía e planificación	17
4.1	Metodoloxía	17
4.1.1	Estudo e familiarización dos conceptos, tecnoloxías e ferramentas	17
4.1.2	Definición dos obxectivos do sistema e planificación do seu desenvolvemento	17
4.1.3	Desenvolvemento do sistema	17
4.1.4	Redacción da memoria final	18
4.2	Planificación	18
4.2.1	Primeira iteración. Desenvolvemento do esqueleto básico e das funcionalidades para usuarios rexistrados.	18
4.2.2	Segunda iteración. Implementación das funcionalidades relacionadas co visor xeográfico e configuración do servidor xeográfico	19
4.2.3	Terceira iteración. Desenvolvemento das funcionalidades de administración	22
4.2.4	Estimación de custos	24
5	Análise	27
5.1	Requisitos	27
5.2	Actores	29
5.3	Casos de uso	30
6	Deseño da aplicación	37
6.1	Backend	37
6.1.1	Entidades do modelo	37
6.1.2	Acceso a datos e persistencia	41
6.1.3	Servizos da lóxica de negocio	43
6.1.4	Capa de servizos REST	46
6.2	Frontend	52
6.2.1	Capa de acceso aos servizos	52
6.2.2	Capa da interface de usuario	53
6.3	QGIS server	56
6.4	Proxy	59

7	Implementación	61
7.1	Backend	61
7.2	Frontend	64
7.3	Proxy	65
8	Probas	67
8.1	Probas unitarias con JUnit	67
8.2	Probas end to end con Cypress	68
8.3	Probas de aceptación	71
9	Conclusións e traballo futuro	77
9.1	Conclusións	77
9.2	Traballo futuro	78
A	Guía de instalación	81
A.1	Instalación de software de base	81
A.2	Configuración dos servidores	81
A.3	Creación da base de datos e configuración do acceso	83
A.4	Instalación da aplicación	84
B	Manual de usuario	85
B.1	Preparación do sistema	85
	B.1.1 Configuración dos datos xerais do proxecto	85
	B.1.2 Configuración dos tipos de atributos	90
	B.1.3 Configuración dos tipos de elementos	92
B.2	Consultar elementos xeométricos	93
B.3	Creación de elementos xeométricos	96
B.4	Modificación de elementos	99
B.5	Revisión e publicación de elementos	100
B.6	Consultar os usuarios do sistema	100
B.7	Consultar os parámetros de configuración	101
B.8	Consultar tipos de atributos	102
B.9	Consultar tipos de elementos	103
B.10	Creación de usuarios	104
B.11	Modificación do estado dun usuario	105
B.12	Modificar tipos de elementos	106
B.13	Modificar tipos de atributos	109
B.14	Eliminar un tipo de elemento	110
B.15	Eliminar un tipo de atributo	110

B.16 Eliminar un elemento	111
B.17 Manexo do visor xeográfico	111
Relación de Acrónimos	115
Glosario	117
Bibliografía	119

Índice de Figuras

3.1	Arquitectura do sistema.	14
4.1	Diagrama de Gantt correspondente á primeira iteración. Inclúe tamén as fases previas de familiarización e planificación.	20
4.2	Diagrama de Gantt correspondente á segunda iteración.	21
4.3	Diagrama de Gantt correspondente á terceira iteración. Inclúe a redacción da memoria final do traballo.	23
5.1	Diagrama de casos de uso implementados na primeira iteración.	32
5.2	Diagrama de casos de uso implementados na segunda iteración.	34
5.3	Diagrama de casos de uso implementados na terceira iteración.	36
6.1	Diagrama de entidades da aplicación.	38
6.2	Diagrama entidade relación da base de datos.	39
6.3	Diagrama do DAO da entidade <i>GeometricElement</i>	43
6.4	Diagrama da interface de <i>AttributeService</i>	44
6.5	Diagrama da interface de <i>ConfigurationParameterService</i>	44
6.6	Diagrama da interface de <i>CodeService</i>	44
6.7	Diagrama do <i>GeometricElementService</i>	45
6.8	Diagrama da interface de <i>UserAccountService</i>	46
6.9	Diagrama do controlador <i>GeometricElementController</i>	47
6.10	Conversor DTO para o <i>UserRoleDto</i>	48
6.11	Resultado dunha petición WFS.	59
6.12	Diagrama do funcionamento do proxy.	60
7.1	Estrutura na que se divide o backend.	62
7.2	Uso da anotación <i>@Autowired</i> no servizo <i>GeometricElementService</i>	62

7.3	Uso da anotación <i>@Transactional</i> nun método do servizo <i>GeometricElementService</i>	63
7.4	Estrutura do código do frontend.	64
8.1	Cobertura dos test dos servizos do backend.	67
8.2	Estrutura do proxecto de probas con Cypress.	68
8.3	Exemplo do código dun test realizado con Cypress.	69
B.1	Táboa cos parámetros de configuración.	86
B.2	Ventá modal para modificar o valor do título da aplicación.	86
B.3	Ventá modal para modificar o valor do favicon da aplicación.	86
B.4	Ventá modal para modificar o valor do pé de páxina da aplicación.	87
B.5	Ventá modal para modificar a imaxe do menú.	87
B.6	Ventá modal para modificar a imaxe do primeiro oco do pé de páxina.	88
B.7	Ventá modal para modificar o texto do primeiro oco do pé de páxina.	88
B.8	Ventá modal para modificar o valor do link ao que apuntará a imaxe do menú.	89
B.9	Ventá modal para modificar a extensión inicial que terá o mapa do visor xeográfico.	89
B.10	Ventá modal para modificar o valor da extensión máxima que terá o mapa do visor xeográfico.	89
B.11	Ventá modal para engadir un tipo de atributo.	90
B.12	Campos a cubrir na definición dun tipo de atributo tipificado.	91
B.13	Campos a cubrir na definición dun tipo de atributo de texto plano.	92
B.14	Ventá modal de creación dun tipos de elemento xeométrico.	92
B.15	Campos a cubrir para os atributos dun novo tipo de elemento.	93
B.16	Control para a consulta de elementos xunto coa información dun elemento creado logo de facer clic sobre el.	94
B.17	Táboa coa información dos elementos asociados a un tipo de elemento.	95
B.18	Táboa coa información dos ficheiros asociados a un elemento.	95
B.19	Control engadir activado e co tipo de elemento seleccionado no mapa.	97
B.20	Panel lateral co formulario e xeometría.	97
B.21	Panel lateral cos valores do elemento a modificar xunto coa xeometría en modo edición.	99
B.22	Campos relativos á revisión e modificación de elementos, xunto co de observacións internas. Só visibles para os administradores.	100
B.23	Táboa cos usuarios existentes no sistema.	101
B.24	Táboa cos parámetros de configuración existentes no sistema.	101
B.25	Táboa cos tipos de atributos existentes no sistema.	102

B.26 Táboa cos valores dun tipo de atributo do sistema.	103
B.27 Táboa cos tipos de atributos existentes no sistema.	103
B.28 Táboa cos tipos de atributos dun tipo de elemento do sistema.	104
B.29 Ventá modal para a creación dun novo usuario.	105
B.30 Ventá para a modificación do estado e/ou o contrasinal dun usuario.	106
B.31 Ventá para a modificación dun tipo de elemento con instancias asociadas.	107
B.32 Ventá para a modificación dun tipo de elemento sen instancias asociadas.	108
B.33 Ventá para a modificación dun tipo de atributo.	109
B.34 Ventá para a modificación dun tipo de atributo dun parámetro de configuración.	110
B.35 Control para ver a máxima extensión configurada.	111
B.36 Control do zoom por encadre activado xunto coa área á que se ampliará a vista do mapa.	112
B.37 Control de medir áreas activado xunto cunha medición feita.	112
B.38 Control de medir distancias activado xunto cunha medición feita.	113
B.39 Estado do control de capas para un usuario anónimo.	113

Índice de Táboas

4.1	Tarefas e estimacións da primeira iteración.	19
4.2	Tarefas e estimacións da segunda iteración.	22
4.3	Tarefas e estimacións da terceira iteración.	24
4.4	Estimacións de custos dos recursos humanos.	24
4.5	Estimacións de custos dos equipos informáticos.	24
4.6	Resumo dos custos do proxecto.	25

Capítulo 1

Introdución

Neste capítulo comentamos a motivación que levou ao desenvolvemento deste traballo e os obxectivos que se pretendían acadar. Tamén resumimos a estrutura da memoria.

1.1 Motivación

En moitos dos procesos de participación pública levados a cabo por distintas organizacións preténdese que os cidadáns fagan as súas suxestións ou aporten as súas ideas sinalando sobre un mapa a localización de diversos tipos de elementos xeorreferenciados. É habitual referirse a este tipo de sistemas coas súas siglas en inglés: **Public Participation Geographic Information System (PPGIS)** [1]. Dependendo do tipo de proceso, nalgúns casos trátase de sinalar zonas ou obxectos xa existentes que deben ser tidos en conta para unha determinada fin. Noutros casos trátase de propoñer a localización para novos elementos a construír, instalar, protexer, etc. Os tipos de elementos e os atributos de cada tipo de elemento dependen de cada proceso concreto pero o mecanismo de participación é en xeral similar.

Nas organizacións nas que sexa frecuente a realización de procesos de participación pública sería útil dispor dunha aplicación que se poida adaptar doadamente e empregar en distintos procesos, sen verse na necesidade de desenvolver unha aplicación a medida para cada proceso. Neste proxecto quérese desenvolver unha aplicación web que cumpra con ese obxectivo.

1.2 Obxectivos

Os obxectivos principais que se fixaron para a aplicación a desenvolver foron:

- Os administradores deben **poder definir dende a interface de usuario os tipos de elementos** que se poden introducir e os atributos de cada tipo de elemento.
- Algúns **aspectos visuais da aplicación deben poder ser configurados tamén** por parte dos administradores: título, logotipos, textos explicativos... de maneira que se

poidan adaptar, dende a propia interface de usuario da aplicación, aos procesos de participación nos que se empregue a aplicación.

- A aplicación que se desenvolva debe **permitir debuxar sobre un visor xeográfico** obxectos con xeometría de tipo punto, liña ou polígono. Ditos obxectos poderán ser revisados polos administradores do sistema e publicados para a súa consulta mediante a propia aplicación.
- Os *frameworks* e compoñentes a usar deben ser **software libre** ou cando menos gratuito, para evitar custes de licenzas e simplificar os requisitos para realizar despregues.

1.3 Traballos relacionados

Existen un par de frameworks que poderían terse aplicado no desenvolvemento deste proxecto, aínda que finalmente optouse por non usalos. Ambos son aplicables a proxectos relacionados coa toma de datos e permiten, en maior ou menor medida, a introdución de datos xeorreferenciados. Trátase de:

- **KoboToolbox** [8]: é un *fork* parcial das ferramentas do **Open Data Kit** [9]. Ambos son usados frecuentemente en proxectos de axuda humanitaria. O código é software libre e destaca sobre todo o *ecosistema* de compoñentes que ofrecen para o deseño de formularios de toma de datos.
- **LimeSurvey** [10]: tamén destaca nas facilidades para o deseño de formularios. O código ofrécese como software libre e ao mesmo tempo a empresa ofrece servizos comerciais para soporte e hosting dos deseños realizados. O seu uso como compoñente integrado en desenvolvementos a medida posiblemente requira traballo de enxeñaría inversa, por exemplo, para analizar a estrutura de almacenamento da información en base de datos.

O motivo principal que nos fixo non decantarnos polo uso destas ferramentas foi que iso implicaría un maior esforzo de desenvolvemento e un maior risco. Ambas ferramentas implican a instalación e configuración de varios subsistemas e o uso de varias linguaxes de programación para a súa integración. Ademais, seguiría sendo necesario desenvolver compoñentes a medida para a xestión do sistema e un traballo importante de creación de compoñentes de integración. Outros factores que se tiveron en conta foron o tempo necesario para a familiarización cos numerosos compoñentes que os conforman e a maior complexidade para o mantemento do sistema resultante.

Optouse así por desenvolver un sistema máis compacto. Con respecto ás ferramentas anteriores, tense menos flexibilidade para o deseño dos formularios. Os administradores si que poden determinar os atributos dos elementos, e por tanto os campos dos formularios, e

tamén pode indicar a orde na que se deben colocar, pero, por exemplo, non se poden configurar *camións alternativos* en función do valor que se introduza nun selector. Este tipo de cousas é unha das melloras a realizar como traballo futuro.

1.4 Estrutura da memoria

A memoria está organizada nos seguintes capítulos:

- **Introdución:** trátase deste capítulo. Resúmese a motivación e os obxectivos do proxecto.
- **Tecnoloxías, ferramentas e librarías:** coméntanse as ferramentas e compoñentes máis relevantes que se empregaron no desenvolvemento do sistema.
- **Arquitectura global do sistema:** descríbese a estrutura de alto nivel do sistema desenvolvido.
- **Metodoloxía e planificación:** coméntase a metodoloxía seguida para o desenvolvemento do proxecto e planificación realizada para levalo a cabo.
- **Análise:** detállanse os requisitos que debe cumprir o sistema.
- **Deseño da aplicación:** explícase o deseño das distintas partes do sistema.
- **Implementación:** resúmese a codificación realizada.
- **Probas:** explícanse as probas realizadas.
- **Conclusións e traballo futuro:** fanse observacións finais sobre o traballo realizado e menciónanse ideas para un posible traballo futuro.
- **Manual de usuario:** inclúese un manual de usuario.

Tecnoloxías, ferramentas e librarías

Neste capítulo relacionamos as principais ferramentas e compoñentes empregados para o desenvolvemento do sistema. Como se comentou previamente todas eles están dispoñibles sen custe tanto para desenvolvemento como para produción.

2.1 Linguaxes de programación

2.1.1 Backend

Para desenvolver a capa modelo e a API REST empregouse a linguaxe Java.

2.1.2 Frontend

Para o desenvolvemento do frontend empregouse JavaScript, HTML e CSS. Tamén se empregou JSX [11], que é unha extensión de JavaScript que permite mesturar JavaScript e HTML e que é usado habitualmente xunto con React [4].

2.2 Frameworks e librarías

2.2.1 Backend

Frameworks e librarías relacionados cos Sistema de Información Xeográfica (SIX)

- **JTS [12]**: é unha librería Java para crear e manipular xeometría vectorial. Ten implementadas moitas operacións con datos espaciais (intersección, unión, etc). JTS é un proxecto do grupo de traballo LocationTech da fundación Eclipse.
- **Hibernate Spatial [13]**: é unha extensión de Hibernate [14] que lle permite o manexo de datos xeográficos. Hibernate é a ferramenta para Object Relational Mapping (ORM)

empregada neste proxecto. Hibernate Spatial admite a maioría das funcións das especificacións do Open Geospatial Consortium (OGC). Entre as bases de datos soportadas encóntrase PostGIS.

- **Spatial4J [15]**: é unha librería de código aberto que facilita, entre outras cousas, a lectura e escritura de formatos espaciais como por exemplo **GeoJSON [16]**, que é o formato usado na API para a información xeográfica.

Frameworks e librerías xerais

- **Spring [6]**: é un framework para Java que proporciona unha gran cantidade de funcionalidades para o desenvolvemento de aplicacións. Neste caso facemos uso de **Spring Boot [17]**, que simplifica moitas das configuracións e integracións habituais en aplicacións web Java . Algúns aspectos salientables son:
 - **Creación de API REST**: coas anotacións de Spring podemos crear unha clase como un controlador REST e os seus métodos como un punto de acceso do mesmo, sendo Spring quen xestione as peticións.
 - **Contedor de inversión de control**: permite a configuración dos compoñentes da aplicación e a administración do ciclo de vida dos obxectos Java, que se leva a cabo principalmente a través da inxección de dependencias.
 - **Xestión de transaccións**: unifica distintas APIs de xestión e coordina as transaccións para os obxectos.
 - **Seguridade**: a través do subproxecto Spring Security podemos securizar a aplicación con opcións como a autenticación, autorización ou o control de acceso a recursos.
 - **Repositorios de datos**: a través dos repositorios de datos podemos axilizar o desenvolvemento das entidades e as súas operacións de acceso á base de datos. Estes repositorios proporcionan operacións básicas Create, Read, Update and Delete (CRUD) que permiten facer consultas sinxelas sen ter que escribir código. Para consultas máis complexas, podemos estender estes repositorios.
- **Querydsl [18]**: é un framework que permite a construción de consultas sobre datos para varios backend incluídos JPA, MongoDB, etc. As consultas constrúense a través dunha API fluída en lugar de escribindo cadeas de texto estático.
- **JWT [19]**: é unha librería de código aberto para crear e verificar tokens web JSON (JavaScript Object Notation) en Java. Emprégase para os tokens relacionados coa identificación do usuario e da sesión.

- **Tika [20]:** permite extraer metadatos de diferentes tipos de ficheiros. É empregada para obter a extensión dos ficheiros a partir do seu tipo mime.

2.2.2 Frontend

Frameworks e librarías SIX

- **OpenLayers [5]:** é unha librería JavaScript de código aberto que permite mostrar mapas interactivos nos navegadores web. Ofrece unha API para acceder a diferentes fontes de información cartográfica: WMS, WFS, mapas de OpenStreetMap e incluso mapas comerciais como pode ser Google Maps.
- **OpenLayers LayerSwitcher [21]:** é un complemento para OpenLayers que actúa como selector das capas que deben estar visibles.

Frameworks e librarías xerais

- **React [4]:** é unha librería JavaScript para o desenvolvemento de interfaces de usuario co paradigma do desenvolvemento baseado en compoñentes. Isto permítenos dividir a interface de usuario en partes máis pequenas chamadas compoñentes, que manexan o seu propio estado, e integralos en interfaces de usuario máis complexas. Destaca no desenvolvemento de interfaces de usuario de aplicacións web de tipo *páxina única* (Single-page Application (SPA)) [22].
- **Bootstrap:** é un framework que facilita o deseño gráfico de aplicacións web que se adapten a diferentes dispositivos.
- **React Bootstrap [23]:** é unha librería que modifica a forma de engadir elementos Bootstrap. Proporciona compoñentes React que renderizan elementos de Bootstrap.
- **react-bootstrap-table2 [24]:** é un framework que facilita a creación de táboas con estilo Bootstrap en aplicacións React.
- **Redux [25]:** é unha librería de JavaScript de código aberto para o manexo do estado dunha aplicación dunha forma centralizada e modular. Neste proxecto empregárase **React Redux [26]**, que é unha librería para React que permite aos compoñentes ler os datos dun *store* de Redux.
- **React Router [27]:** para navegar entre as distintas *pantallas* da aplicación usamos a librería de enrutamento React Router, concretamente react-router-dom. Permite navegar pola aplicación mediante enlaces e provoca, mediante programación, un cambio de

pantalla ademais de cambiar a URL que se amosa no navegador para que sexa consistente co que se está vendo.

- **React Intl [28]**: é unha librería que permite internacionalizar as aplicacións React de xeito declarativo. Esta librería proporciona compoñentes React e unha API para dar formato a datas, números e mensaxes de texto.
- **Draft.js [29]**: é un editor de texto con formato creado para React. Usa un formato propio. Para convertelo a HTML úsanse librerías complementarias:
 - **draft-js-export-html [30]**: converte de HTML ao formato propio de Draft.js.
 - **draft-js-import-html [31]**: converte a HTML o formato propio de Draft.js.
- **Formik [32]**: é unha librería que simplifica a creación de formularios en React.
- **Yup [33]**: é unha librería JavaScript que permite a validación de valores mediante a creación dun esquema. É empregado por Formik para a validación dos formularios.
- **jQuery [34]**: é unha librería de JavaScript de propósito xeral que simplifica a maneira de interactuar con documentos HTML.
- **Lodash [35]**: é unha librería de JavaScript que fai máis cómoda a manipulación de arrays, números, obxectos, etc.
- **Axios [36]**: é unha librería JavaScript capaz de executarse en Node.js [37] e nun navegador e facilita a realización de peticións como cliente HTTP.
- **React Responsive Carousel [38]**: é unha librería para React que permite visualizar galerías de fotografías de forma vistosa.

2.2.3 Ferramentas para probas

JUnit

JUnit [39] é un framework empregado para facer probas unitarias en Java. Permite executar distintos tipos de métodos para determinar se o comportamento dun determinado compoñente de código é o que debería ser. Emprega para iso funcións que avalían o valor de retorno esperado en función duns valores de entrada, se o valor retornado pola función é o esperado entón JUnit indicará que o método pasou correctamente a proba. Por outra banda, se o valor esperado é distinto ao devolto, JUnit devolverá un erro no método correspondente.

Para revisar a cobertura dos test feitos con JUnit empregouse o plugin para Eclipse **EclEmma [40]**, que á súa vez fai uso da librería **JaCoCo [41]**. Desta maneira pódese analizar a

cobertura de maneira non intrusiva, sen ter que facer ningún cambio específico no código da aplicación nin dos tests.

Cypress

Cypress [42] é unha ferramenta de probas de frontends. Funciona interactuando directamente cun navegador no que se carga a aplicación e está pensada sobre todo para realizar tests sobre aplicativos JavaScript modernos como pode ser os que usan React. Cypress permítenos crear todo tipo de tests: tests end-to-end, tests de integración e tests unitarios. Empregamos Cypress neste proxecto para as probas end to end, que simulan o comportamento dun usuario: facer clic, escribir, desprazarse, etc.

2.3 Ferramentas de desenvolvemento

2.3.1 Maven

Apache Maven é unha ferramenta que axuda á xestión e á construción de proxectos Java. Maven emprega un Project Object Model (POM) para describir o proxecto a construír. Ademais de información relativa ao mesmo indícanse tamén as dependencias de outros módulos e compoñentes externos, que se descargarán e configurarán automaticamente, e tamén a orde de construción dos elementos.

Xunto con Maven empregase o **Frontend Maven Plugin** [43], que é unha librería que descarga e instala npm localmente no proxecto de Maven. Isto permite compilar o backend e o frontend á vez e que se copien os ficheiros resultantes de executar npm install no frontend na carpeta target do backend.

2.3.2 Npm

Npm (Node Package Manager) [44] é un xestor de paquetes para JavaScript. É o xestor de paquetes por defecto de Node.js. Consiste nun cliente de liña de comandos, tamén chamado npm, e unha base de datos de paquetes, chamada rexistro npm. O cliente interacciona co rexistro e permite aos usuarios manexar as dependencias da aplicación e instalar aplicacións Node.js que se atopen no repositorio.

2.3.3 Git

Git é un software de control de versións distribuído libre. Está deseñado para coordinar o traballo entre os programadores pero tamén pode empregarse para levar un rexistro de cambios en calquera tipo de ficheiro. Os seus obxectivos inclúen a velocidade, a integridade dos datos e o soporte para fluxos de traballo non lineais.

2.3.4 Redmine

Redmine [45] é unha ferramenta flexible para a xestión de proxectos desenvolvida co framework Ruby On Rails. Inclúe funcionalidades como:

- Soporte para múltiples proxectos.
- Uso de roles de usuario.
- Unha *wiki* por proxecto.
- Soporte para repositorios (neste caso empregouse Git).

2.3.5 QGIS

QGIS [46] é un Sistema de Información Xeográfica (SIX) de escritorio multiplataforma e de código libre. QGIS admite diversos formatos de datos ráster e vectoriais. Entre as súas principais características atopamos:

- Ver datos: podemos ver combinacións de datos vectoriais e ráster en distintos formatos e proxeccións sen ter que convertelos a un formato interno ou común. Por exemplo:
 - Táboas e vistas habilitadas espacialmente con PostGIS e outros formatos vectoriais.
 - Datos espaciais en liña procedentes de servizos web como WMS, WFS, WCS...
- Explorar datos e compoñer mapas: pódese compoñer mapas e explorar datos espaciais interactivamente coa súa interface de usuario.

No desenvolvemento, a versión de escritorio de QGIS empregouse para consultar os datos da base de datos e para crear o proxecto de configuración de QGIS Sever [7], que é o servidor de información xeográfica que se emprega no sistema desenvolvido, tal como se comenta máis adiante. Esta integración da ferramenta de escritorio cos servizos web é un dos puntos fortes de QGIS.

2.3.6 Eclipse

Eclipse [47] é un entorno de desenvolvemento integrado ou *Integrated Development Environment (IDE)* de código aberto e multiplataforma. É un dos IDE para desenvolvemento Java máis empregado, aínda que tamén soporta outras linguaxes de programación, como por exemplo JavaScript. Ten boa integración con Git e empregouse sobre todo para a programación do backend.

2.3.7 Visual Studio Code

Visual Studio Code [48] é un editor multiplataforma desenvolvido por Microsoft. Vén con soporte integrado para JavaScript, TypeScript e Node.js aínda que existe a posibilidade de engadir extensións para outras linguaxes. Empregouse para a desenvolvemento do frontend.

2.3.8 draw.io

draw.io [49] é un editor gráfico online (aínda que tamén ten aplicación de escritorio) co que se poden crear múltiples diagramas, como por exemplo diagramas de fluxo, diagramas UML, diagramas entidade-relación, etc. Empregouse para documentar o deseño da aplicación.

2.3.9 Overleaf

Overleaf [50] é un editor web e colaborativo de LaTeX, empregado para a elaboración da memoria do traballo.

2.4 Gateway HTTP

Para a captura e modificación de peticións HTTP empregouse **Zuul** [51]. É unha ferramenta creada por Netflix que nos permite filtrar peticións HTTP e modificalas segundo as necesidades de maneira dinámica. Neste caso empregámolo como intermediario (Proxy) anteposto ao servidor xeográfico debido a que non se quere que os usuarios poidan acceder directamente a dito servidor, xa que o servidor xeográfico non ten control de acceso sobre os recursos. Mediante Zuul modifícanse as peticións, engadindo filtros sobre a información a devolver polos servizos de información xeográfica en función do usuario que realiza petición. Zuul permite ser integrado con Spring Boot o cal nos facilita o seu uso no noso sistema.

2.5 Servidores de aplicacións

Durante o desenvolvemento do proxecto empregáronse dous distintos:

- **Tomcat**: é o empregado para servir o backend. En desenvolvemento empregouse a versión que ven integrada por defecto co proxecto base de Spring Boot.
- **Node.js** [37]: é un entorno de execución para JavaScript. É o servidor empregado para servir o frontend.

Xa en produción, tanto o frontend como o backend serán despregados xuntos nun servidor Tomcat.

2.6 Servidor xeográfico

Como servidor da información xeográfica vía web decidiuse empregar **QGIS server** [7]: é unha implementación de código aberto dos servizos Web Map Service (WMS) 1.3, Web Feature Service (WFS) 1.0.0, WFS 1.1.0 e Web Coverage Service (WCS) 1.1.1 que ofrece ademais características cartográficas avanzadas para o mapeo temático. QGIS server funciona como unha aplicación Common Gateway Interface (CGI) / FastCGI, e require estar integrado xunto cun servidor web. Neste caso execútase xunto con Apache. Un punto forte de QGIS Server é que a configuración das capas de información a ofrecer mediante os distintos servizos web pode facerse mediante a interface gráfica da aplicación de escritorio de QGIS, non sendo necesaria a codificación manual de ficheiros de configuración.

2.7 Sistema xestor de bases de datos

Como servidor de bases de datos optouse por usar **PostgreSQL** [2], un servidor relacional clásico moi contrastado. Para este proxecto empregouse a versión 11.4. O motivo de elixir PostgreSQL foi poder usar a extensión espacial **PostGIS** [3], que engade a PostgreSQL soporte para obxectos xeográficos así como numeras funcións para realizar operacións xeométricas. PostGIS soporta a especificación Simple Features for SQL (SFSQL) e, dentro do software libre, é o modulo espacial para bases de datos que ofrece máis garantías.

Arquitectura global do sistema

3.1 Componentes do sistema

Na figura 3.1 podemos ver todos os componentes nos que se divide o sistema implementado.

- **Frontend:**

- Interface de usuario: esta é a capa encargada de mostrar ao usuario a información da aplicación.
- Capa de acceso aos servizos: é a encargada de realizar as peticións aos servizos REST do backend e de recibir as respostas.

- **Backend:**

- Proxy: é o encargado de filtrar e redirixir as peticións sobre servidor xeográfico. Neste caso está metido dentro da propia aplicación web.
- Servizos REST: é a capa encargada de recibir as peticións do frontend e, unha vez obtidos os datos necesarios, devolver os resultados.
- Capa modelo: é onde se definen as entidades do domino da aplicación e onde se encontran os distintos servizos internos. Os servizos REST chaman a estes servizos para obter os datos necesarios.
- Capa de persistencia: nesta capa xestionase a persistencia das entidades do modelo. Os servizos acceden a esta capa para consultar e modificar os datos na base de datos.

- **Base de datos:** neste caso empregárase PostgreSQL coa extensión PostGIS.

- **Servidor xeográfico:**

- Apache: QGIS Server executarase como un módulo CGI/FastCGI dentro do servidor web Apache.
- QGIS server: non está accesible directamente polo frontend, senón que se accede mediante o proxy. Dános a información de todos os elementos xeométricos que figuran na base de datos a través dos protocolos WMS [52] e WFS [53]. WMS é un protocolo estándar definido polo OGC para servir vía web mapas de datos referenciados espacialmente, de forma dinámica, a partir de información xeográfica. WFS empégase para servir obxectos xeográficos xunto cos seus atributos.
- **Servidores WMS externos:** dende o frontend, ademais de empregar WMS para acceder ao servidor xeográfico interno, tamén se usa para incluír no visor xeográfico mapas publicados por servizos externos de terceiros.

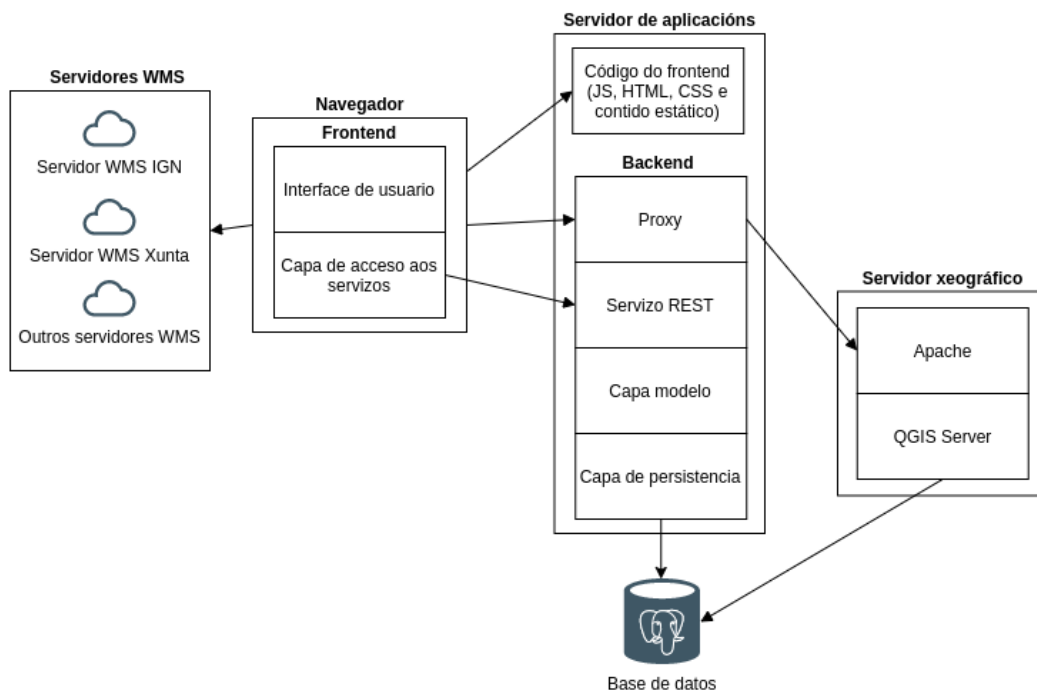


Figura 3.1. Arquitectura do sistema.

3.2 Interacción entre os compoñentes

Unha vez que un usuario accede ao sistema, ao tratarse dunha interface de usuario de tipo *páxina única* (Single-page Application (SPA)) [22], o frontend cárgase completamente no

navegador. Para obter os datos, o frontend realizar chamadas aos servizos REST do backend. As chamadas son realizadas mediante React coa API fetch. O formato usado polos servizos REST é JSON.

Cando a petición chega ao controlador do REST, implementado con Spring, este delega a execución nun servizo interno, que terá en conta a lóxica necesaria para levar a cabo a operación. Os servizos REST non devolven directamente entidades do modelo senón que están baseados en Data Transfer Object (DTO). Os servizos do modelo comunícanse cos Data Access Object (DAO) das entidades, que conteñen os métodos de persistencia da aplicación, empregando *Hibernate*. Este framework é o encargado de facer a comunicación co xestor da base de datos, neste caso PostgreSQL, mediante Java Database Connectivity (JDBC).

Cando o visor xeográfico do frontend fai peticións dirixidas ao servidor xeográfico, todas elas pasarán polo proxy, que engadirá os filtros pertinentes, en función do usuario que faga a petición, e logo será redirixida ao servidor xeográfico, neste caso QGIS Server. Os filtros teñen por obxecto principal que un usuario non poida ver elementos sobre os que non ten permiso. En canto aos servidores WMS externos, dende o frontend, coa librería OpenLayers, fanse as peticións WMS necesarias e é tamén esta librería a que se encarga de procesar as respostas recibidas.

Metodoloxía e planificación

4.1 Metodoloxía

A realización do proxecto levouse a cabo en distintas fases, seguindo unha metodoloxía iterativa incremental. A continuación comentamos cada unha destas fases.

4.1.1 Estudo e familiarización dos conceptos, tecnoloxías e ferramentas

Durante esta primeira fase do proxecto trátase de adquirir coñecementos relacionados coas ferramentas e tecnoloxías a empregar no proxecto. Especialmente, a familiarización con React e aquelas librarías empregadas no frontend, ademais de aquelas empregadas no backend e as relacionadas cos SIX.

4.1.2 Definición dos obxectivos do sistema e planificación do seu desenvolvemento

Nesta segunda fase especificanse os obxectivos que se deben lograr con este traballo e faise a división do desenvolvemento en iteracións, tendo cada unha delas unha serie de tarefas asignadas para poder conseguir os obxectivos establecidos. Estas iteracións, que finalmente foron tres, e as súas tarefas, xunto coa estimación de tempos a empregar en cada unha delas, detállanse na sección [4.2](#).

4.1.3 Desenvolvemento do sistema

Unha vez completada a planificación realízase o desenvolvemento do sistema seguindo as iteracións establecidas.

4.1.4 Redacción da memoria final

Unha vez rematado o desenvolvemento do sistema hai que levar a cabo a elaboración da memoria final.

4.2 Planificación

Para o desenvolvemento dispúxose de dous recursos humanos:

- **Xefe de proxecto:** este rol é asumido de maneira conxunta polos directores do proxecto. As súas actividades consisten principalmente en apoiar e supervisar as tarefas de análise, deseño e documentación e na realización dalgunhas probas.
- **Analista-programador:** este rol é asumido pola persoa que realiza o proxecto, que participa de forma activa en todas as actividades que comprenden a elaboración do sistema.

A continuación detállanse as tarefas establecidas para cada iteración e a súa estimación en horas de traballo. Para cada estimación indícase tamén o tempo que finalmente levou realizar a tarefa, para poder ver a desviación que houbo sobre o estimado. As actividades e tempos indicados corresponden ao analista-programador. Para as actividades e tempos do xefe de proxecto, realizouse unha estimación proporcional sobre a estimación realizada para o analista-programador:

- *Actividades de análise:* o 50% do tempo adicado polo analista-programador.
- *Actividades de deseño:* o 30% do tempo adicado polo analista-programador.
- *Actividades de implementación:* sen participación.
- *Actividades de probas:* o 30% do tempo adicado polo analista-programador.
- *Actividades de documentación:* o 25% do tempo adicado polo analista-programador.

4.2.1 Primeira iteración. Desenvolvemento do esqueleto básico e das funcionalidades para usuarios rexistrados.

Nesta primeira iteración incluíronse as tarefas necesarias para completar o esqueleto básico da aplicación e todas aquelas funcionalidades non administrativas que non necesitaban do visor xeográfico.

Ao remate desta iteración, o sistema debía permitir a autenticación e rexistro de usuarios, así como o cambio de idioma e a actualización do contrasinal.

Ademais incluíronse tamén nesta iteración as consultas das táboas de elementos xeométricos. Como aínda non estaría dispoñible o visor xeográfico, a introdución dos elementos xeométricos necesarios para as probas previuse facela mediante SQL. As funcionalidades para a eliminación dos elementos xeométricos tamén se meteron nesta iteración.

Na figura 4.1 podemos ver o diagrama das tarefas desta iteración xunto co seu tempo estimado. A maiores, por abreviar, tamén se incluíu neste apartado a estimación da fase de familiarización coas ferramentas e o contexto do proxecto. Na táboa 4.1 podemos comparar as estimacións realizadas cos tempos finalmente empregados.

Tarefa	Tempo estimado	Tempo real
Estudo e familiarización cos conceptos, tecnoloxías e ferramentas a empregar	240 h	280 h
Definición dos obxectivos do sistema e planificación do seu desenvolvemento	40 h	60 h
Primeira iteración		
Análise	16 h	16 h
Deseño da capa modelo	30 h	40 h
Implementación da capa modelo	24 h	30 h
Probas unitarias da capa modelo	20 h	16 h
Deseño e implementación do servizo REST	60 h	70 h
Deseño da interface de usuario	20 h	20 h
Implementación da interface de usuario	100 h	110 h
Probas end to end	20 h	13 h
Total	570 h	655 h

Táboa 4.1. Tarefas e estimacións da primeira iteración.

4.2.2 Segunda iteración. Implementación das funcionalidades relacionadas co visor xeográfico e configuración do servidor xeográfico

Nesta segunda iteración lévase a cabo todas a funcionalidades que teñen relación directa co visor xeográfico. Estas funcionalidades inclúen a inserción, modificación e consulta de elementos xeométricos dende o mapa. Ademais de aquelas funcionalidades para o manexo do visor, inclúese tamén a configuración do servidor xeográfico.

Na figura 4.2 podemos ver o diagrama das tarefas. Na táboa 4.2 podemos visualizar as tarefas coa estimación dos tempos así como o tempo real adicado.

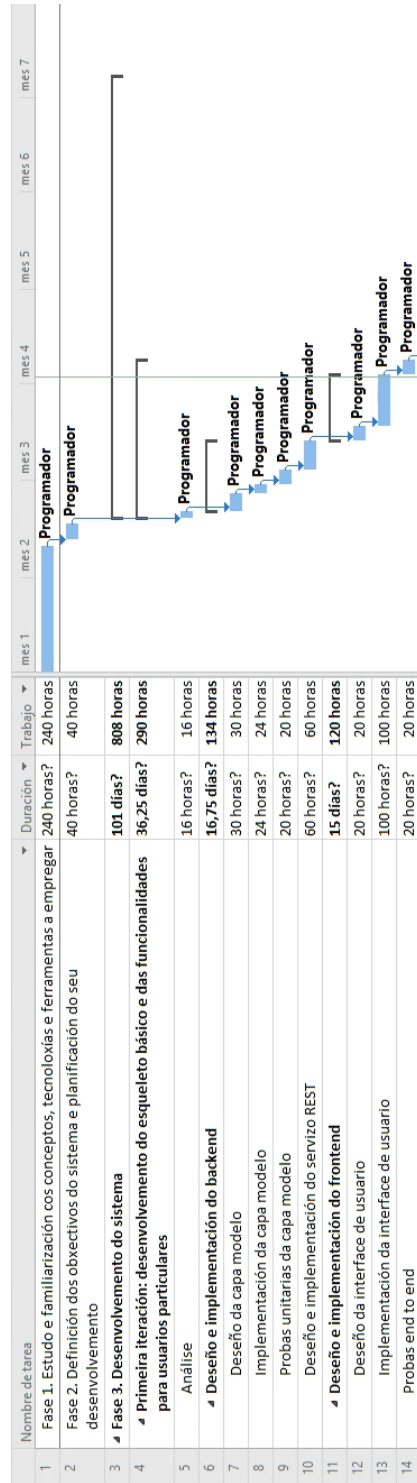


Figura 4.1. Diagrama de Gantt correspondente á primeira iteración. Inclúe tamén as fases previas de familiarización e planificación.



Figura 4.2. Diagrama de Gantt correspondente á segunda iteración.

Tarefa	Tempo estimado	Tempo real
Análise	16 h	16 h
Deseño e implementación de funcionalidades xenéricas do manexo do visor	80 h	66 h
Deseño e implementación de funcionalidades específicas relacionadas cos elementos xeométricos dende o frontend	120 h	140 h
Configuración do visor xeográfico	40 h	41 h
Probas end to end	20 h	20 h
Total	276 h	283 h

Táboa 4.2. Tarefas e estimacións da segunda iteración.

4.2.3 Terceira iteración. Desenvolvemento das funcionalidades de administración

Nesta terceira e última iteración figuran todas aquelas tarefas relacionadas coa administración do sistema. Os administradores do sistema poderán consultar, modificar e eliminar tipos de atributos e tipos de elementos xeométricos. Ademais, teñen a posibilidade de consultar, engadir e modificar o estado (bloquear ou desbloquear) dos usuarios do sistema. Por último, poderán consultar e modificar os parámetros de configuración que se gardan en base de datos.

Na figura 4.3 podemos ver o diagrama de Gantt das tarefas. Tamén se inclúe nese diagrama a redacción da memoria. Na táboa 4.3 podemos visualizar as tarefas coa estimación dos tempos así como o tempo real adicado. Ao igual ca no diagrama de Gantt tamén se inclúe a redacción da memoria final.

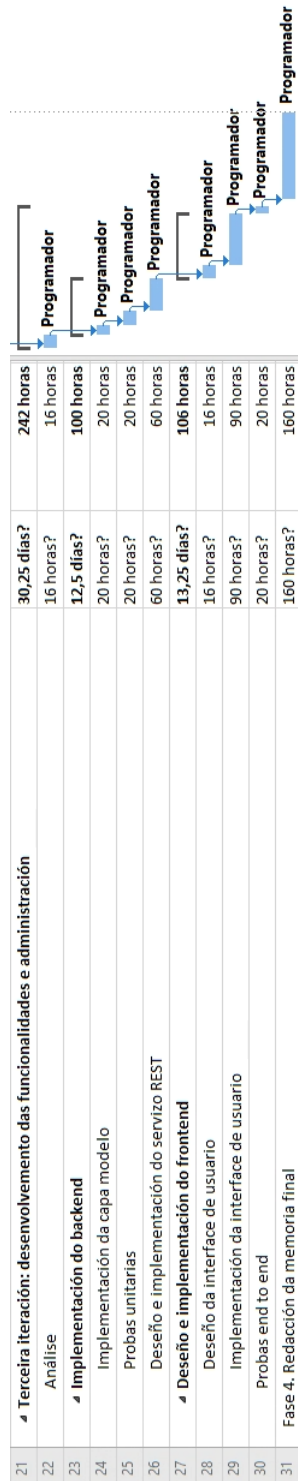


Figura 4.3. Diagrama de Gantt correspondente á terceira iteración. Inclúe a redacción da memoria final do traballo.

Tarefa	Tempo estimado	Tempo real
Análise	16 h	16 h
Implementación da capa modelo	20 h	27 h
Probas unitarias	20 h	13 h
Deseño e implementación do servizo REST	60 h	20 h
Deseño da interface de usuario	16 h	10 h
Implementación da interface de usuario	90 h	100 h
Probas end to end	20 h	30 h
Redacción da memoria final	160 h	180 h
Total	402 h	396 h

Táboa 4.3. Tarefas e estimacións da terceira iteración.

4.2.4 Estimación de custos

Custos directos

Os dous recursos humanos mencionados anteriormente teñen un custo directo sobre o proxecto que podemos ver na táboa 4.4.

Recurso	Tempo estimado	Custo por hora	Custo total
Analista-programador	1248 h	20€	24.960 €
Xefe de proxecto	110 h	40€	4.400 €
Total			29.360 €

Táboa 4.4. Estimacións de custos dos recursos humanos.

A parte dos custos derivados dos dous recursos humanos mencionados anteriormente, está o custo dos equipos informáticos que van empregar os recursos humanos. Na táboa 4.5 podemos ver os custos derivados destes medios. Estimamos que cada equipo ten unha vida útil de 4 anos e para este proxecto será necesario usalos durante 8 meses.

Recurso	Cantidade	Custo total	Porcentaxe aplicada	Prorrateo
Analista-programador	1	1500 €	100%	250 €
Xefe de proxecto	2	3000 €	15%	75 €
Total				325 €

Táboa 4.5. Estimacións de custos dos equipos informáticos.

Custos indirectos

A maiores dos custos directos existen outros que non afectan directamente a implementación do proxecto pero si ao entorno no que este se desenvolve como poden ser gastos en electricidade, auga, infraestruturas, etc. Estimamos que os custos indirectos son un 13% sobre o resto de custos.

Custos totais

Na táboa 4.6 faise un resumo do total dos custos estimados e dos custos reais.

	Horas estimadas	Horas reais	Custo estimado	Custo real
Custo do analista-programador	1.248 h	1.324 h	24.960 €	26.480 €
Custo do xefe de proxecto	110 h	127 h	4.400 €	5.080 €
Custo dos equipos	325 €			
Custos indirectos	3.859 €			
TOTAL	1.358 h	1.451 h	33.544 €	35.744 €

Táboa 4.6. Resumo dos custos do proxecto.

Capítulo 5

Análise

Unha vez coñecida a motivación do proxecto e os obxectivos de alto nivel, identificáronse os requisitos que debería cumprir o sistema. Neste capítulo enumeramos ditos requisitos e os casos de uso aos que dan lugar. Están divididos por iteracións.

5.1 Requisitos

- **Primeira iteración:**

- **Soporte para múltiples idiomas:** calquera visitante terá a posibilidade de establecer o idioma da interface de usuario da aplicación (a elixir entre galego, español e inglés).
- **Autenticación e rexistro de usuarios:** a aplicación distinguirá entre usuarios autenticados e usuarios anónimos. Os usuarios anónimos poderán rexistrarse (crear unha conta) ou autenticarse se teñen conta.
- **Cambio do contrasinal dun usuario:** calquera usuario autenticado terá a posibilidade de cambiar o seu contrasinal actual.
- **Visualización de elementos xeométricos por tipo:** calquera usuario poderá consultar unha lista dos elementos xeométricos de cada tipo:
 - * **Usuarios anónimos:** só poderán consultar aqueles elementos xeométricos que sexan públicos
 - * **Usuarios rexistrados:** poderán consultar os elementos xeométricos públicos e aqueles elementos creados por eles mesmos.
 - * **Usuarios administradores:** poderán consultar todos os elementos xeométricos existentes, podendo ver por separado aqueles que foron creados por eles mesmos.

-
- **Ficheiros asociados a elementos xeométricos:** permitirase anexar ficheiros aos elementos e a consulta dos detalles dos mesmos, así como a súa descarga. Ademais existirá a posibilidade de visualizar as imaxes asociadas a cada elemento.
 - **Eliminar elementos xeométricos:** tanto os usuarios rexistrados como os administradores poderán eliminar elementos xeométricos. Os administradores poderán eliminar calquera elemento xeométrico mentres que os usuarios rexistrados só poderán eliminar aqueles creados por eles mesmos que aínda non fosen revisados polos administradores.
- **Segunda iteración:**
 - **Funcionalidades xenéricas relacionadas co manexo do visor xeográfico:**
 - * **Ampliar unha zona no visor xeográfico:** calquera usuario poderá ampliar a vista do visor a unha área delimitada por un cadro que el mesmo debuxe.
 - * **Medición de distancias e áreas:** calquera usuario terá a posibilidade de facer medicións de distancias e áreas no visor xeográfico.
 - **Funcionalidades relacionadas con elementos xeométricos:**
 - * **Consultar elementos xeométricos no visor xeográfico:** todos os usuarios poderán consultar sobre un mapa os elementos xeométricos:
 - **Usuarios anónimos:** só poderán visualizar aqueles elementos xeométricos que sexan públicos.
 - **Usuarios rexistrados:** poderán visualizar tanto os elementos xeométricos públicos como aqueles creados por eles mesmos. Estes últimos deben poder velos por separado.
 - **Usuarios administradores:** poderán visualizar todos os elementos xeométricos.
 - * **Engadir e modificar elementos xeométricos:** tanto os usuarios rexistrados como os administradores poderán engadir novos elementos xeométricos no sistema. Terán tres vías para iniciar o proceso de introdución:
 - Dende o menú da aplicación.
 - Dende o propio visor xeográfico.
 - Dende a lista das contribucións do propio usuario.
- A modificación poderán realizala:
- Dende os detalles do elemento xeométrico no visor xeográfico.
 - Dende a táboa de elementos xeométricos que se corresponda co tipo de elemento a introducir.

- **Terceira iteración:**

- **Funcionalidades relacionadas con tipos de elementos xeométricos:** os administradores poderán consultar, engadir, modificar e eliminar tipos de elementos xeométricos.
- **Funcionalidades relacionadas con tipos de atributos:** os administradores poderán consultar, engadir, modificar e eliminar tipos de atributos.
- **Funcionalidades relacionadas cos usuarios do sistema:** os administradores poderán consultar, engadir e modificar (bloquear ou desbloquear e cambiar o contrasinal) dos usuarios do sistema.
- **Funcionalidades relacionadas con parámetros de configuración:**
 - * Os administradores poderán modificar o valor dalgúns dos parámetros de configuración do sistema.
 - * Entre os parámetros de configuración modificables dende a interface de usuario deben estar:
 - Título da aplicación.
 - Link ao sitio principal que faga uso do sistema.
 - Imaxe na que colocar o link anterior.
 - Extensión inicial do visor xeográfico.
 - Extensión máxima do visor xeográfico.
 - Elementos a colocar no pé de páxina da web.

5.2 Actores

Identificáronse tres tipos distintos de actores:

- **Usuario anónimo:** todo aquel visitante que non se autentica. Estes usuarios dispoñerán das funcionalidades de consulta de datos.
- **Usuario rexistrado:** son os usuarios particulares que se dan de alta no sistema e se autentican. Ademais das funcionalidades dos usuarios anónimos, dáselles acceso aos casos de uso de introdución, modificación e eliminación de elementos xeométricos.
- **Usuario administrador:** os administradores dispoñen das funcionalidades dos actores anteriores e a maiores tamén disporán das funcionalidades de administración, tales como crear usuarios, crear novos tipos de elementos xeométricos ou incluso cambiar algúns aspectos da interface de usuario modificando parámetros de configuración.

5.3 Casos de uso

A continuación describense os casos de uso, agrupándoos tamén nas tres iteracións nas que se dividiu o proxecto:

- **Primeira iteración:**

- **CU-001. Rexistrarse:** os usuarios anónimos poderán rexistrarse no sistema, crear unha conta de usuario, para acceder ás funcionalidades dos usuarios rexistrados.
- **CU-002. Iniciar sesión:** os usuarios cunha conta creada no sistema poderán iniciar sesión coas súas credenciais.
- **CU-003. Ver elementos xeométricos dun tipo dado:** tanto os usuarios anónimos como os usuarios rexistrados poderán consultar, nunha táboa paxinada, os elementos xeométricos públicos existentes no sistema.
- **CU-004. Ver elementos xeométricos propios dun tipo dado:** tanto os usuarios rexistrados como os administradores poderán consultar, nunha táboa paxinada, as súas contribucións por cada tipo de elemento xeométrico.
- **CU-005. Cambiar idioma:** calquera usuario poderá cambiar o idioma da interface de usuario da aplicación (galego, español ou inglés).
- **CU-006. Cambiar contrasinal:** tanto os usuarios rexistrados como os administradores poderán cambiar o seu contrasinal.
- **CU-007. Ver fotos adxuntas a un elemento xeométrico:** permitirase ver todas as fotos asociadas a un elemento xeométrico.
- **CU-008. Consultar ficheiros adxuntos a un elemento xeométrico:** permitirase consultar a información dos ficheiros asociados a un elemento xeométrico.
- **CU-009. Descargar un ficheiro asociado a un elemento xeométrico:** permitirase a descarga de cada un dos ficheiros adxuntos a un elemento xeométrico.
- **CU-010. Eliminar un elemento xeométrico:** tanto os usuarios rexistrados como os administradores poderán eliminar un elemento xeométrico do sistema. Os administradores poderán eliminar calquera elemento, en cambio, os usuarios rexistrados só poderán eliminar aqueles creados por eles que non fosen aínda revisados.
- **CU-011. Consultar todos os elementos xeométricos dun tipo:** os administradores poderán consultar, nunha táboa paxinada, todos os elementos xeométricos dun tipo dado.

No diagrama da figura 5.1 móstranse estes casos de uso. Podemos comprobar que calquera usuario ten acceso ás funcionalidades de visualización de elementos xeométricos públicos. Dende aí pode acceder ás funcionalidades relacionadas cos ficheiros asociados aos elementos. Tanto os administradores como os usuarios rexistrados poderán consultar as súas contribucións (os elementos creados por eles) e eliminar un elemento xeométrico. Os administradores non teñen a posibilidade de consultar os elementos xeométricos públicos dun tipo, senón que se lles amosa a lista de todos os elementos existentes de cada tipo.

- **Segunda iteración:**

- **CU-012. Engadir un elemento xeométrico:** tanto os usuarios rexistrados como os administradores poderán crear un novo elemento xeométrico.
- **CU-013. Modificar un elemento xeométrico:** tanto os usuarios rexistrados como os administradores poderán modificar un elemento xeométrico. No caso dos administradores poderán modificar calquera elemento. Os usuarios rexistrados poderán modificar só os creados por eles que aínda non fosen revisados por parte dun administrador.
- **CU-014. Consultar elementos xeométricos públicos dun tipo dado no visor xeográfico:** tanto os usuarios anónimos como os usuarios autenticados poderán ver todos os elementos xeométricos públicos de cada tipo no visor xeográfico.
- **CU-015. Consultar elementos xeométricos propios dun tipo dado no visor xeográfico:** tanto os administradores como os usuarios rexistrados poderán consultar as súas contribucións de cada tipo dende o visor xeográfico.
- **CU-016. Consultar todos os elementos xeométricos dun tipo dado no visor xeográfico:** os administradores poderán consultar todos os elementos xeométricos de cada tipo dende o visor xeográfico.
- **CU-017. Ver detalles dun elemento xeométrico no visor xeográfico:** permitirase consultar no visor xeográfico os detalles dun elemento xeométrico facendo clic sobre o mesmo. Os usuarios anónimos só o poderán facer sobre aqueles que sexan públicos, os usuarios rexistrados poderán facelo tanto sobre os públicos como sobre aqueles creados por eles mesmos e os administradores poderán facelo sobre calquera elemento.
- **CU-018. Centrar o visor entornando a un elemento xeométrico:** permitirase centrar o visor xeográfico entornando a un elemento xeométrico.
- **CU-019. Medición de áreas no visor xeográfico:** calquera usuario poderá medir áreas sobre o visor xeográfico.

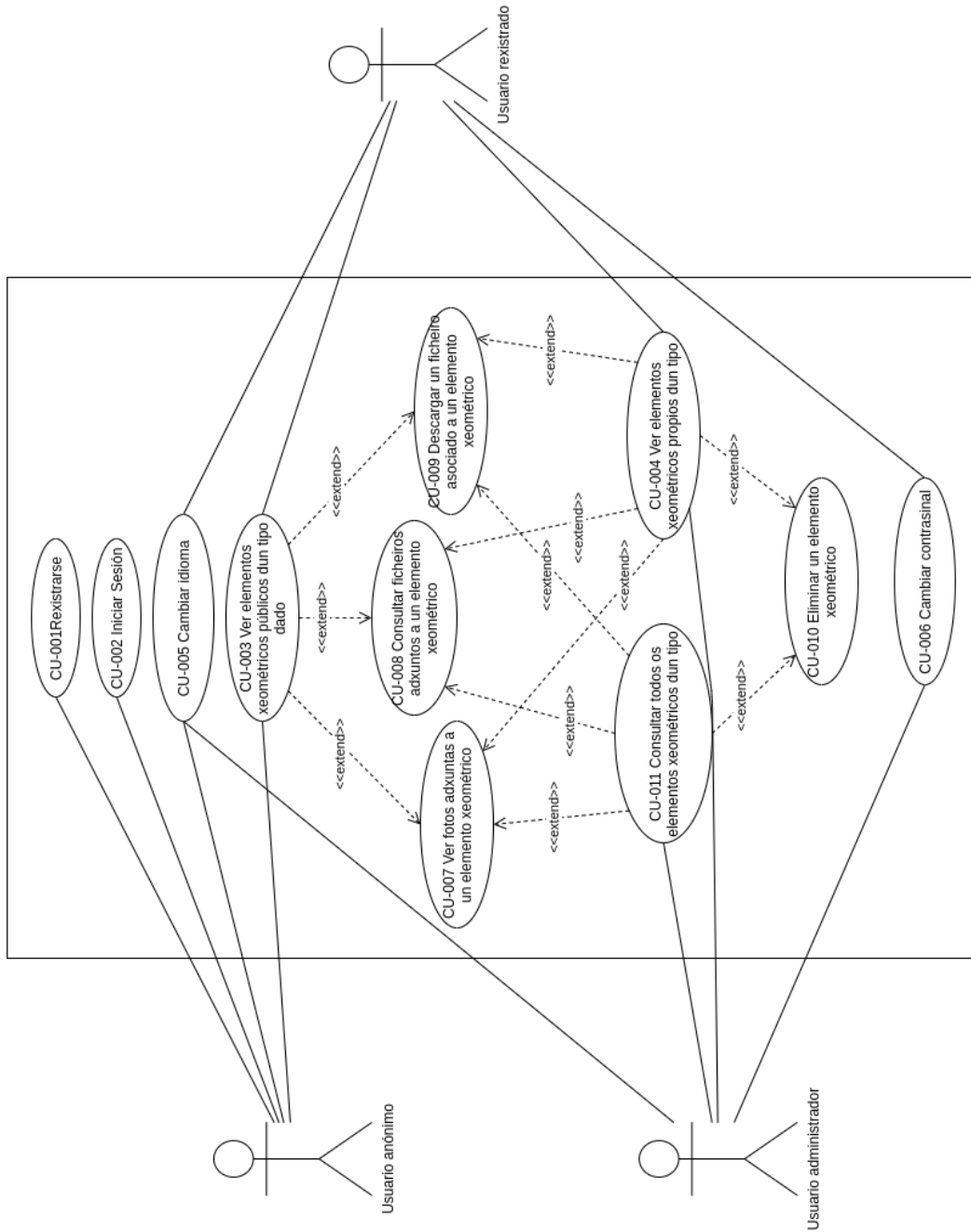


Figura 5.1. Diagrama de casos de uso implementados na primeira iteración.

- **CU-020. Medición de distancias no visor xeográfico:** calquera usuario poderá medir distancias sobre o visor xeográfico.
- **CU-021. Ampliar unha zona no visor xeográfico:** calquera usuario poderá situar a vista do visor nun encadre establecido por el.

No diagrama da figura 5.2 móstranse estes casos de uso. Todos os usuarios poderán consultar elementos xeométricos públicos dun tipo dado no visor xeográfico, así como realizar medicións e ampliar unha zona concreta. Como no caso da consulta en táboas, os administradores teñen en exclusiva a funcionalidade de poder consultar todos os elementos xeométricos dun tipo, esta vez sobre o visor xeográfico. Por último, tanto os administradores como os usuarios rexistrados poderán engadir e modificar elementos xeométricos.

- **Terceira iteración:**

- **CU-022. Consultar os tipos de elementos xeométricos:** os administradores poderán consultar todos os tipos de elementos xeométricos existentes no sistema.
- **CU-023. Consultar os tipos de atributos:** os administradores poderán consultar todos os tipos de atributos existentes no sistema.
- **CU-024. Consultar os usuarios do sistema:** os administradores poderán consultar os usuarios rexistrados no sistema.
- **CU-025. Consultar os parámetros de configuración:** os administradores poderán consultar os valores dos parámetros que se poden modificar dende a interface de usuario.
- **CU-026. Engadir un tipo de elemento xeométrico:** os administradores poderán engadir un novo tipo de elemento xeométrico.
- **CU-027. Engadir un tipo de atributo:** os administradores poderán engadir un novo tipo de atributo.
- **CU-028. Engadir un usuario:** os administradores poderán crear novos usuarios (tanto novos administradores como usuarios rexistrados).
- **CU-029. Modificar un tipo de elemento xeométrico:** os administradores poderán modificar un tipo de elemento xeométrico xa existente. Se xa existen elementos de dito tipo, as modificacións límitanse ao nome do tipo elemento. En particular, non se podería nin engadir nin quitar tipos de atributos.
- **CU-030. Modificar un tipo de atributo:** os administradores poderán modificar un tipo de atributo existente no sistema.

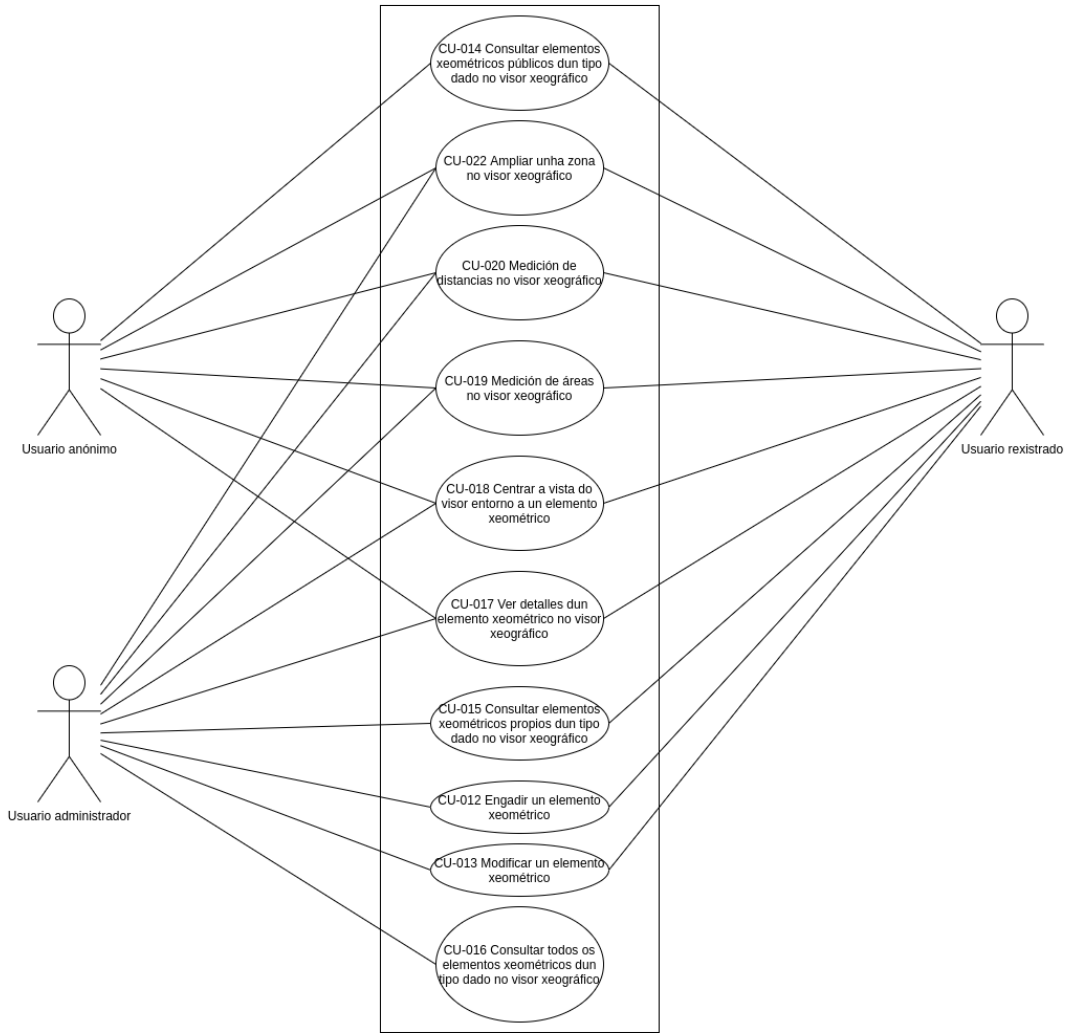


Figura 5.2. Diagrama de casos de uso implementados na segunda iteración.

- **CU-031. Modificar o valor dun parámetro de configuración:** os administradores poderán modificar o valor dun parámetro de configuración, de entre aqueles parámetros que se gardan na base de datos. (Non poderán modificar parámetros configurados en ficheiros *properties*).
- **CU-032. Modificar o estado dun usuario:** os administradores poderán bloquear ou desbloquear aos usuarios do sistema.
- **CU-033. Modificar o contrasinal dun usuario:** os administradores poderán cambiar o contrasinal de calquera usuario do sistema.
- **CU-034. Eliminar un tipo de elemento xeométrico:** os administradores poderán eliminar un tipo de elemento xeométrico existente no sistema, sempre que non existan xa elementos de dito tipo.
- **CU-035. Eliminar un tipo de atributo:** os administradores poderán eliminar un tipo de atributo existente no sistema, sempre que non estea sendo usado nalgún tipo de elemento xeométrico.

No diagrama da figura 5.3 móstranse estes casos, que son exclusivos do administrador. Podemos comprobar que dende as funcionalidades de consulta poden acceder ás funcionalidades de introdución, modificación e eliminación, excepto para os parámetros de configuración que non poden ser eliminados.

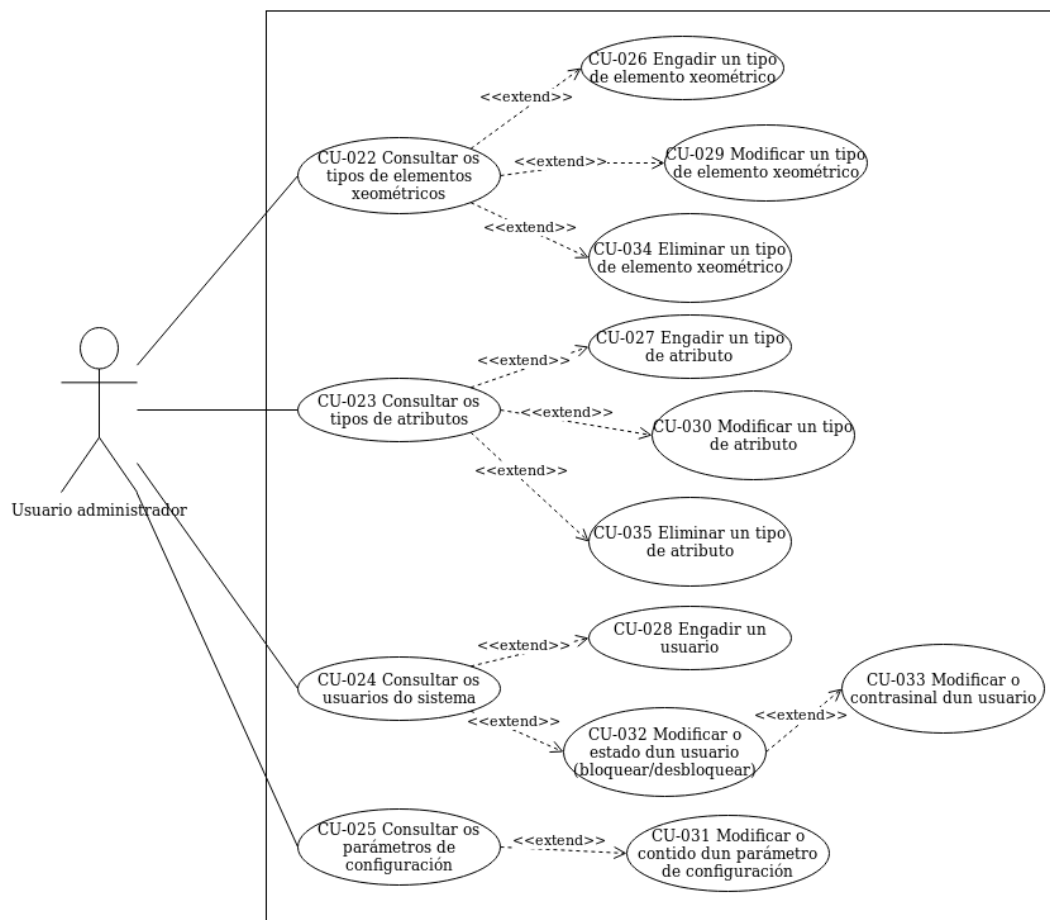


Figura 5.3. Diagrama de casos de uso implementados na terceira iteración.

Deseño da aplicación

No capítulo 3 xa se describiu a arquitectura global do sistema. Neste capítulo aprofundamos un pouco máis en cada unha das súas partes: backend, frontend, servidor xeográfico e proxy de acceso ao servidor xeográfico.

6.1 Backend

O backend é onde se executa a lóxica de negocio da aplicación, se accede á base de datos para a lectura e persistencia de datos e se ofrece a información que se necesita no frontend, usualmente mediante unha API.

Dentro do backend inclúese a definición das clases que representan as entidades do modelo da aplicación. Tamén se inclúen as clases que xestionan a persistencia de ditas entidades, facendo uso do patrón Data Access Object (DAO) [54]. Para a comunicación entre o backend e o frontend deseñouse unha API de tipo Representational State Transfer (REST) [55] que emprega o formato de datos JSON (GeoJSON para os atributos de tipo xeométrico). Para a implementación da capa de servizos REST deseñáronse un conxunto de recursos REST e de controladores para realizar operacións sobre eles. Entre os controladores REST e as entidades do modelo existe unha capa de servizos onde se aplica a lóxica de negocio do sistema.

Para a implementación das distintas capas do backend fíxose uso de Spring Boot, un dos frameworks do ecosistema Spring, que facilita de maneira importante a creación de arquitecturas deste tipo.

Comentamos a continuación os detalles de cada unha das partes do backend.

6.1.1 Entidades do modelo

Nas figuras 6.1 e 6.2 móstranse respectivamente o diagrama de entidades da capa modelo e o diagrama entidade-relación da base de datos. As entidades identificadas no deseño son as seguintes:

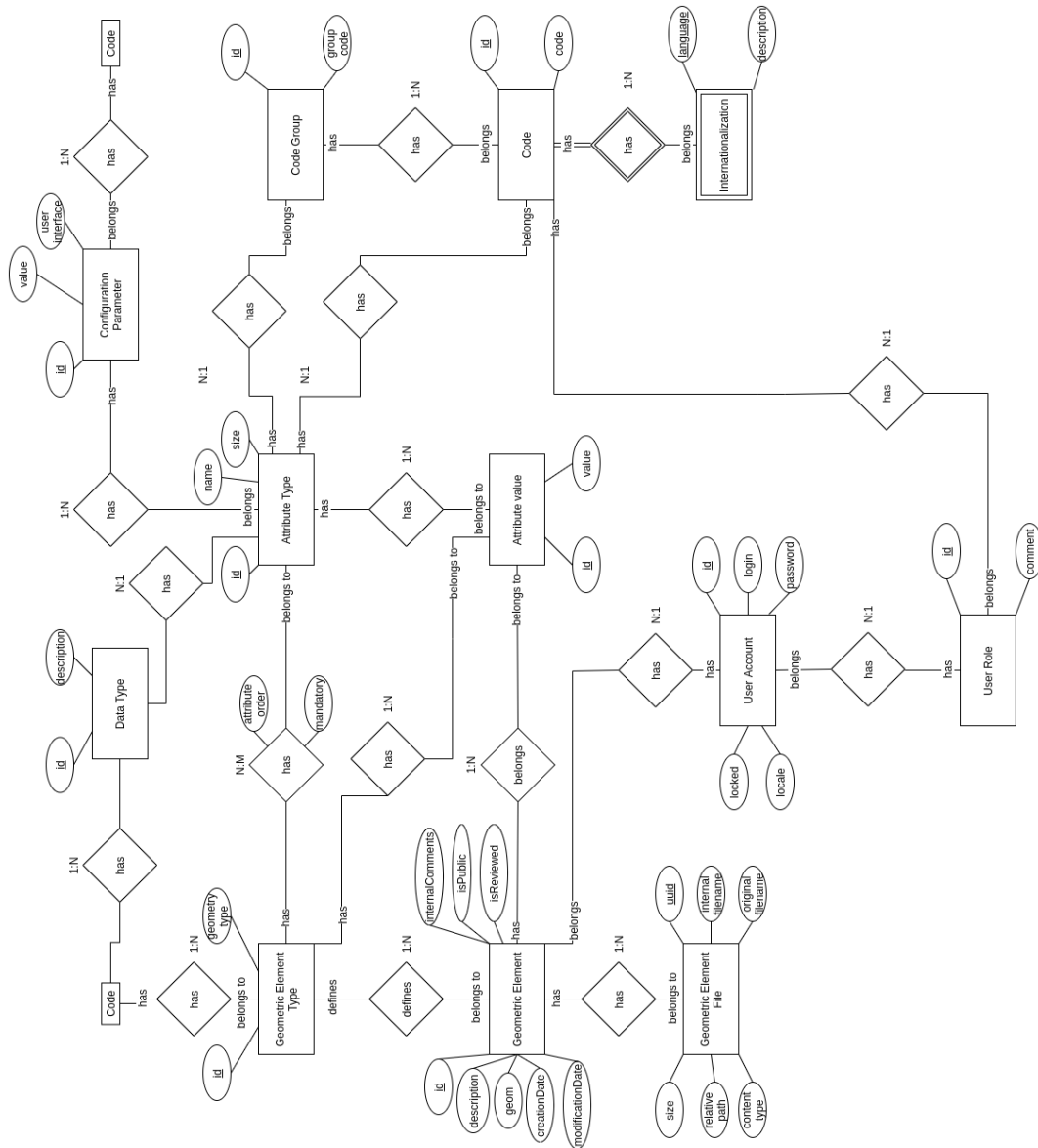


Figura 6.2. Diagrama entidade relación da base de datos.

- **GeometricElementType**: representa os tipos de elementos xeométricos existentes no sistema.
- **AttributeType**: representa os tipos de atributos dos elementos xeométricos que son configurables dinamicamente, por parte dos administradores dende a interface de usuario da aplicación. Cada tipo de elemento xeométrico poderá ter asignados un conxunto de tipos de atributos. Previamente á creación dun novo tipo de elemento xeométrico,

todos os tipos de atributos que vaia ter deben xa existir. Un tipo de atributo pode ser asignado a máis dun tipo de elemento. A orde e o carácter obrigatorio do atributo pode asignarse para cada tipo de elemento no que se use.

- **DataType:** representa os tipos de datos que se poden asignar aos tipos de atributos. Entre eles están: texto plano, texto con formato (HTML), data, conxunto de valores predefinido (*enumerado*), etc.
- **GeometricElement:** representa os elementos xeométricos existentes na aplicación. A maiores dos atributos asignados dinamicamente a cada tipo de elemento, existen atributos *estáticos* comúns a todos os tipos de elementos (descricaoión, usuario creador do elemento, xeometría, etc.).
- **AttributeValue:** representa os valores dos atributos *dinámicos* dos elementos xeométricos.
- **GeometricElementFile:** representa os ficheiros asociados a cada elemento xeométrico.
- **UserRole:** representa os posibles roles dun usuario da aplicación. Considéranse dous roles: administrador (ADMIN) e usuario rexistrado (USER).
- **UserAccount:** representa a un usuario da aplicación. Cada usuario só pode ter un rol.
- **ConfigurationParameter:** representa os parámetros de configuración existentes que se poden configurar dende a propia a interface de usuario da aplicación.
- **Code:** contén todos os códigos das distintas entidades do modelo da aplicación. Por exemplo, os códigos dos tipos de elementos, os códigos dos parámetros de configuración, etc. Estes poden formar grupos e poden ter descrições nos distintos idiomas da aplicación, a empregar cando se amosan na interface de usuario.
- **CodeGroup:** representa os grupos de códigos existentes. Un exemplo de grupo de códigos sería o formado polos posibles valores que poden asignarse a un atributo de tipo *enumerado* que o usuario cubre mediante un selector.
- **Internationalization:** representa a descrição/nome dos código en cada un dos idiomas soportados pola aplicación (galego, español ou inglés).

A maiores das entidades anteriores, na base de datos creouse tamén unha vista SQL para facilitar o acceso aos datos por parte de QGIS Server. Desta vista fálase con máis detalle na sección 6.3.

6.1.2 Acceso a datos e persistencia

Para a implementación dos DAO empregouse **Spring Data** [56], un módulo de persistencia integrado dentro de Spring Boot. Para realizar a Object Relational Mapping (ORM), Spring Data entrega a API estándar Java Persistence API (JPA) [57]. Como implementación de JPA neste caso empregouse Hibernate [14], unha implementación popular de JPA que ademais soporta tipos de datos xeométricos grazas á extensión Hibernate Spatial [13].

A continuación lístanse os métodos que foi necesario engadir aos DAO. Non se inclúen os DAO nos que non foi necesario engadir ningún método:

- **AttributeTypeDao:**

- *Optional<AttributeType> findByCode(Code code)*: devolve un tipo de atributo a partir do seu código.

- **AttributeValueDao:**

- *List<AttributeValue> findByGeometricElement(GeometricElement geometricElement)*: devolve unha lista dos valores dos atributos *dinámicos* dun elemento xeométrico dado.

- **CodeDao:**

- *Optional<Code> findByCode(String code)*: devolve un obxecto Código a partir do *string* do código.

- **CodeGroupDao:**

- *Optional<CodeGroup> findByGroupCode(String groupCode)*: devolve un grupo de códigos a partir do *string* do código do grupo.

- **GeometricElementDao:**

- *Page<GeometricElement> findByGeometricElementType(GeometricElementType geometricElementType, Pageable pageable)*: devolve unha páxina de elementos xeométricos dun tipo dado.
- *Page<GeometricElement> find(GeometricElementType geometricElementType, Long ownerId, Boolean isPublic, int page, int size)*: busca avanzada que devolve unha páxina de elementos xeométricos podendo filtrar polo tipo de elemento xeométrico, polo seu creador e por se é público ou non.

- **GeometricElementFileDao:**

-
- *List<GeometricElementFile> findByGeometricElement(GeometricElement geometricElement)*: devolve os ficheiros asociados a un elemento xeométrico dado.

- ***GeometricElementTypeAttributeTypeDao***:

- *List<GeometricElementTypeAttributeType> findByGeometricElementType(GeometricElementType geometricElementType)*: devolve a lista de asociacións entre tipos de atributos e tipos de elementos xeométricos para un tipo de elemento xeométrico dado.
- *Page<GeometricElementTypeAttributeType> findByAttributeType(AttributeType attributeType, Pageable pageable)*: devolve a lista de asociacións entre tipos de atributos e tipos de elementos xeométricos para un tipo de atributo dado.

- ***UserAccountDao***:

- *Optional<UserAccount> findByLogin(String login)*: devolve unha conta de usuario a partir do nome de usuario.

- ***UserRoleDao***:

- *Optional<UserRole> findByCode(String code)*: devolve un rol de usuario a partir do código do mesmo.

Na figura 6.3 podemos ver un diagrama coa estrutura dunha entidade e o seu DAO. Este último está estendido cun método que require ser implementado especificamente (non abonda coas implementacións por defecto que proporciona Spring Data).

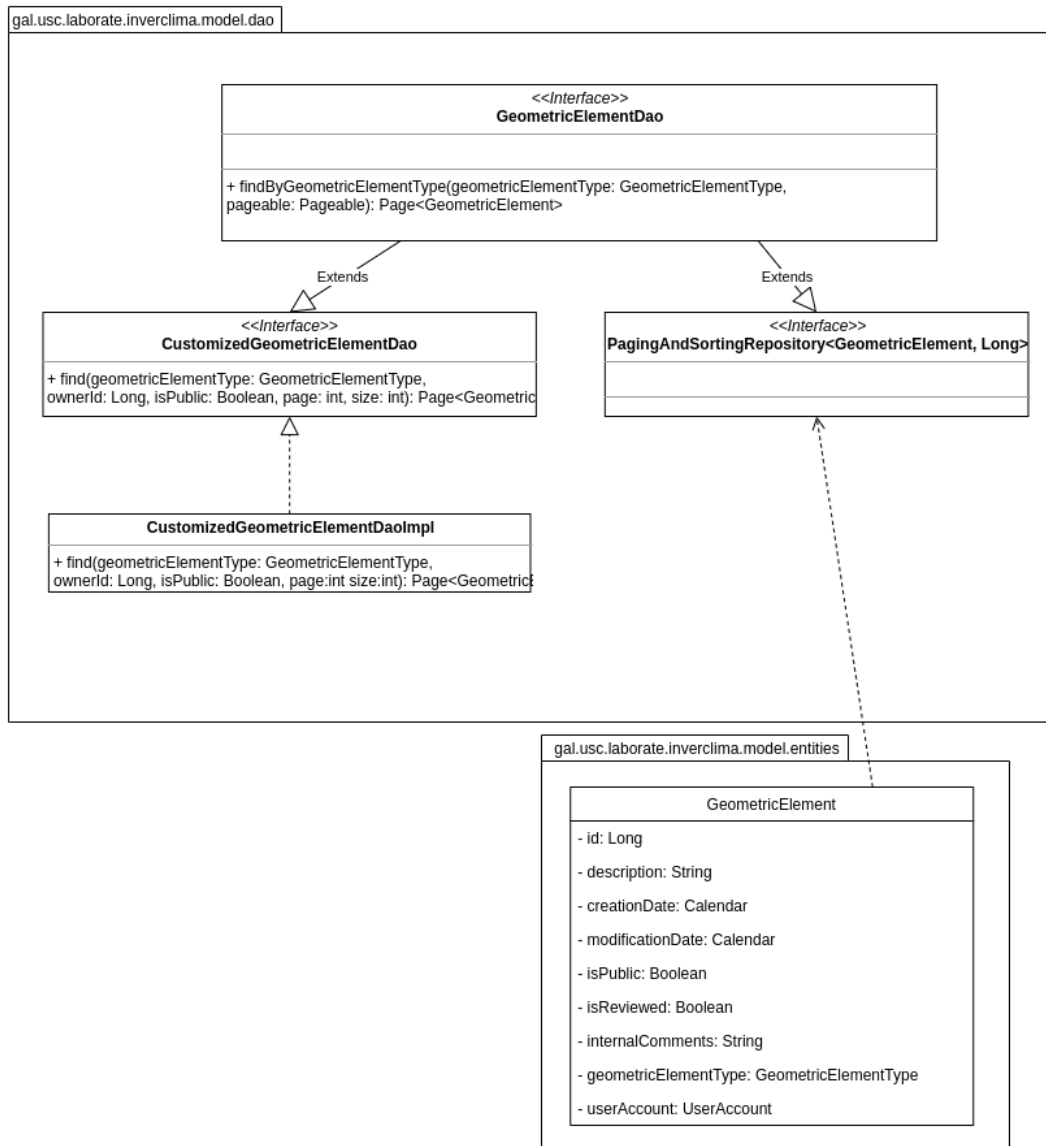


Figura 6.3. Diagrama do DAO da entidade GeometricElement.

6.1.3 Servizos da lóxica de negocio

Os servizos son clases que usan o patrón *facade* [58] para expoñer as funcionalidades e operacións de lóxica de negocio relacionadas. Identificáronse os seguintes cinco servizos:

- **AttributeService:** neste servizo están as operacións de creación, modificación, busca e eliminación das entidades relacionadas cos atributos configurables (AttributeType e AttributeValue). Na figura 6.4 móstrase a súa interface.

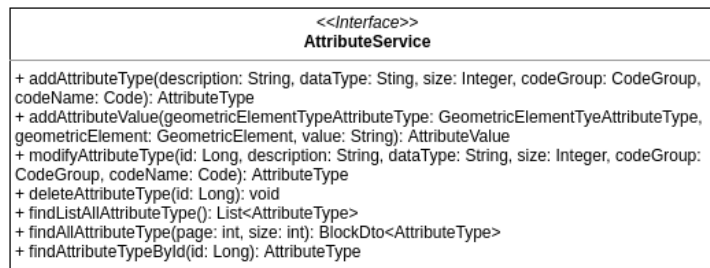


Figura 6.4. Diagrama da interface de *AttributeService*.

- **ConfigurationParameterService:** ocúpase dos parámetros de configuración que son configurables dende a interface de usuario. Na figura 6.5 móstrase a súa interface.

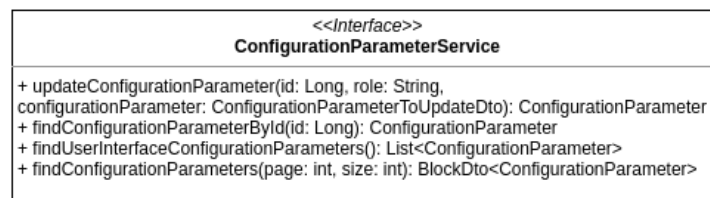


Figura 6.5. Diagrama da interface de *ConfigurationParameterService*.

- **CodeService:** xestiona a creación, modificación, busca e eliminación das entidades relacionadas coa internacionalización (Code, CodeGroup e Internationalization). Na figura 6.6 podemos ver a súa interface.

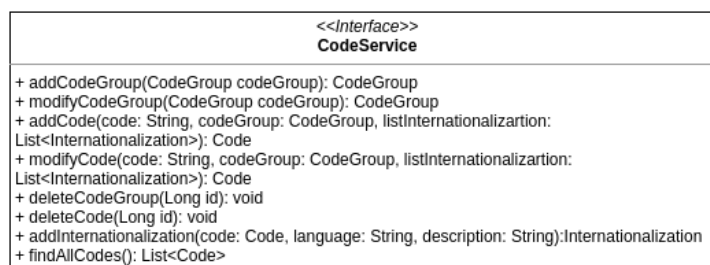


Figura 6.6. Diagrama da interface de *CodeService*.

- **GeometricElementService:** xestiona a creación, modificación, borrado e busca das

entidades relacionadas con elementos xeométricos (GeometricElement, GeometricElementType, GeometricElementFile e GeometricElementTypeAttributeType). Na figura 6.7 podemos ver a estrutura deste servizo, neste caso coas súas relacións cos DAOs.

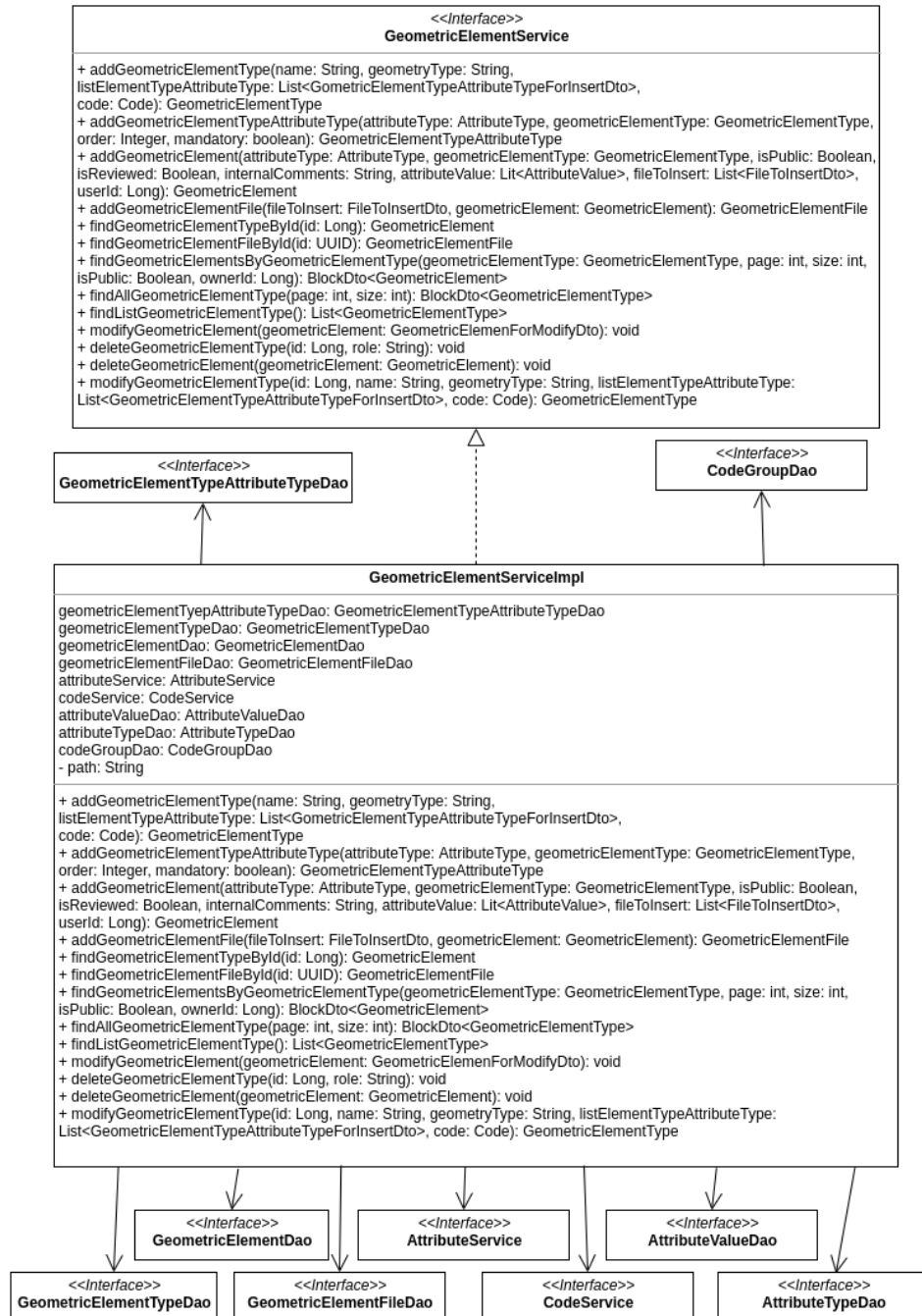


Figura 6.7. Diagrama do GeometricElementService.

- **UserAccountService:** ocúpase das operacións de rexistro e autenticación de usuarios, cambio de linguaxe e contrasinal. Os usuarios administradores ademais poderán bloquear/desbloquear e cambiar o contrasinal dos usuarios do sistema. Por último tamén se facilita a busca por roles e a recuperación de todas as contas de usuario do sistema. Na figura 6.8 podemos ver a súa interface.

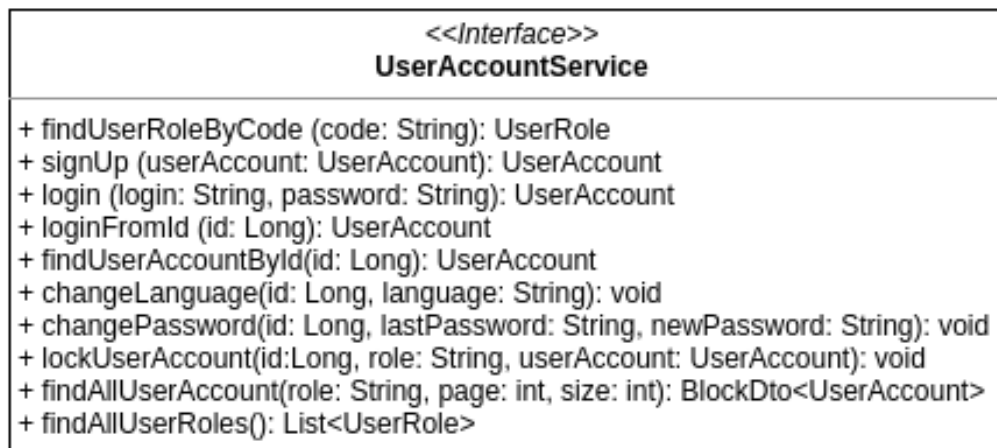


Figura 6.8. Diagrama da interface de *UserAccountService*.

6.1.4 Capa de servizos REST

Controladores REST

Son clases Java empregadas para expoñer os puntos de acceso da API REST. Para o intercambio de información entre o backend e o frontend empregaranse os Data Transfer Object (DTO) que se comentarán no seguinte apartado.

Na figura 6.9 podemos ver a estrutura do controlador que traballa cos elementos xeométricos.



Figura 6.9. Diagrama do controlador *GeometricElementController*.

DTOs (Data Tranfer Objects)

Son obxectos sen estado empregados na comunicación entre o backend e o frontend. Es- tritamente, o intercambio de información entre estas capas faise empregando o formato JSON pero os DTOs son convertidos a JSON e viceversa cando o backend envíe ou reciba peticións, respectivamente.

Un DTO é un obxecto plano que encapsula os datos cos que se está tratando e devólveos nunha soa resposta. Por outra banda, os DTO tamén permiten crear obxectos que agrupen información de varias entidades.

Para transformar os obxectos do modelo en DTOs e viceversa, faise uso de clases de conversión. Na figura 6.10 podemos ver a clase para a conversión a DTOs da entidade *UserRole*.

```
public class UserRoleDtoConversor {

    public static final UserRoleDto toUserRoleDto(UserRole userRole) {
        return new UserRoleDto(userRole.getId(), userRole.getCode(), userRole.getComment());
    }

    public static final UserRole toUserRole(UserRoleDto userRoleDto){
        UserRole userRole = new UserRole(userRoleDto.getCode(), userRoleDto.getComment());
        userRole.setId(userRoleDto.getId());
        return userRole;
    }

    public static final List<UserRoleDto> toUserRoleListDto(List<UserRole> listUserRole) {
        List<UserRoleDto> resultList = new ArrayList<>();

        for(UserRole userRole : listUserRole) {
            resultList.add(toUserRoleDto(userRole));
        }

        return resultList;
    }
}
```

Figura 6.10. Conversor DTO para o *UserRoleDto*.

Recursos REST

A API REST publica unha serie de recursos sobre os que os clientes, neste caso o frontend, poden realizar operacións. Cada recurso correspóndese cunha ruta relativa, path, nas URLES de acceso á API. En función do método HTTP con que se accede ao recurso (GET, POST, DELETE...) a operación a realizar sobre o recurso variará. A definición dos paths de cada recurso faise nos controladores REST mediante anotacións. Para cada recurso e operación que se quere soportar, no controlador debe existir un método, convenientemente anotado, que leve a cabo a correspondente operación. A continuación lístanse os recursos que se expoñen na aplicación e os métodos soportados en cada un:

- **/elements:** baixo este path agrúpanse os recursos relacionadas con elementos xeométricos:
 - */element_type/*: operación para a creación dun novo tipo de elemento xeométrico. Recíbese o tipo de xeometría, a lista de atributos e os datos para a internacionalización do nome do tipo de elemento. Como resposta envíase o recurso creado.
 - */geometric_element_type/*: operación para a recuperación de tipos de elementos xeométricos paxinados. Recíbese o tamaño e o número da páxina solicitada e

- devólvese a lista dos tipos de elementos correspondentes á páxina solicitada e o número total de tipos de elementos existentes.
- */geometric_element_type/all/*: operación que devolve todos os tipos de elementos xeométricos.
 - */geometric_element_type/id/geometric_elements/admin*: operación para a recuperación dos elementos xeométricos do tipo do "id" incluído no path. Só os administradores realizar esta operación. Recíbense varios parámetros: número de páxina, tamaño de páxina e a maiores dous parámetros para filtrar por elementos públicos e polo autor. Devólvese a lista dos elementos para esa páxina e o número total de elementos que cumpren as condicións de filtrado.
 - */geometric_element_type/id/geometric_elements/user*: operación para a recuperación dos elementos xeométricos do tipo do "id" incluído no path. Só os usuarios rexistrados poderán empregar este recurso. Recíbense varios parámetros: número de páxina, tamaño de páxina e a maiores dous parámetros para filtrar por elementos públicos e polo autor. Tamén se recupera do token JWT o id do usuario que fai a petición. Se o autor polo que filtrar non coincide co usuario que fai a petición lánzase unha excepción. Se si coincide devólvese a lista dos elementos para esa páxina e o número total de elementos que cumpren as condicións de filtrado.
 - */geometric_element_type/id/geometric_elements*: operación para a recuperación dos elementos xeométricos públicos do tipo do "id" incluído no path. Só os usuarios anónimos poderán empregar este recurso. Recíbense dous parámetros: número de páxina e tamaño de páxina. Devólvese a lista dos elementos para esa páxina e o número total de elementos públicos existentes.
 - */geometric_element/*: operación para a creación dun novo elemento xeométrico. Recíbense os parámetros necesarios para a súa creación, entre os que se inclúen: lista dos ficheiros a engadir, lista dos valores dos atributos dinámicos e o tipo de elemento xeométrico. Devólvese o elemento creado.
 - */geometric_element/id/*: dependendo do método HTTP empregado a operación que se executa é distinta:
 - * GET: operación para a recuperación do elemento xeométrico que teña o "id" incluído no path.
 - * POST: operación para a modificación do elemento xeométrico que teña o "id" o incluído no path. Recíbense os valores necesarios para a súa modificación.
 - * DELETE: operación para a eliminación do elemento xeométrico que teña o "id" incluído no path.

- */geometric_element_type/id/*: dependendo do método HTTP empregado a operación que se executa é distinta:
 - * GET: operación para a recuperación do tipo de elemento xeométrico que teña o "id" incluído no path.
 - * POST: operación para a modificación do tipo de elemento xeométrico que teña o "id" incluído no path. Recíbense os novos valores a asignar ao tipo de elemento. Devólvese o tipo de elemento modificado.
 - * DELETE: operación para a eliminación do tipo de elemento xeométrico que teña o "id" incluído no path.
- */geometric_element/files/uuid/*: operación para a recuperación do ficheiro asociado a un elemento xeométrico que teña o "uuid" incluído. Devólvese o contido do ficheiro.
- ***/attributes***: baixo este path agrúpanse os recursos relacionados cos atributos dinámicos dos elementos xeométricos:
 - */attribute_type*: dependendo do método HTTP empregado a operación que se executa é distinta:
 - * GET: operación para a recuperación de todos os tipos de atributos existentes no sistema. Recíbese o número da páxina a recuperar e o tamaño de páxina. Devólvese a lista dos tipos atributos correspondentes a esa páxina e o número total de tipos de atributos existentes.
 - * POST: operación para a creación dun novo tipo de atributo. Recíbense os datos necesarios para a súa creación e devólvese o recurso creado.
 - */attribute_type/id*: dependendo do método HTTP empregado a operación que se executa é distinta:
 - * GET: operación para a recuperación do tipo de atributo co "id" incluído no path.
 - * POST: operación para a modificación do tipo de atributo co "id" incluído no path. Recíbense os novos datos do tipo de atributo. Devólvese o tipo de atributo actualizado.
 - * DELETE: operación para a eliminación do tipo de atributo co "id" incluído no path.
 - */attribute_type/all*: operación para a recuperación da lista de todos os tipos de atributos existentes na base de datos.
- ***/configuration_parameter***: baixo este path agrúpanse os recursos relacionados cos parámetros de configuración:

- */update/id/*: operación para a modificación do parámetro de configuración co "id" incluído no path. Recíbense os valores necesarios para a modificación do seu valor. Devólvese o parámetro actualizado.
 - */all*: operación para a recuperación de todos os parámetros de configuración existentes na base de datos. Recíbese o número de páxina a recuperar e o tamaño de páxina e devólvese a lista dos elementos desa páxina así como o número total de parámetros de configuración existentes na base de datos.
 - */user_interface*: operación para a recuperación de todos os parámetros de configuración existentes na base de datos que afectan ao aspecto da interface de usuario.
 - */user_interface/image/id*: operación para a recuperación da imaxe asociada a un parámetro de configuración que sexa de tipo imaxe. Se o parámetro de configuración asociado ao "id" incluído no path é de tipo imaxe envíase o contido desta na resposta HTTP.
- ***/users***: baixo este path agrúpanse os recursos relacionados coa xestión de usuarios:
 - */signUp*: operación de rexistro. Recíbense os datos do usuario e devólvese o usuario creado e o token JWT.
 - */login*: operación de inicio de sesión. Recíbense as credenciais e devólvese o usuario xunto co token JWT.
 - */createUser*: operación para a creación dun novo usuario por parte dun administrador. Recíbense os datos do usuario e devólvese o usuario creado.
 - */id*: operación POST para o cambio de contrasinal do usuario co "id" incluído no path. Se o que fai a petición non coincide con ese id lanzarase unha excepción.
 - */all*: operación para a recuperación paxinada de todos os usuarios do sistema. Recíbese o número da páxina e o tamaño da páxina e devólvese a lista dos elementos desa páxina e o número total de usuarios existentes.
 - */loginFromServiceToken*: operación para a reautenticación do usuario a partir do token JWT que devolveu o backend cando o usuario se autenticou explicitamente.
 - */changeLanguage*: operación para o cambio de linguaxe do usuario que lanza a petición.
 - */roles/code*: operación para a recuperación do rol de usuario que teña o código incluído no path.
 - */roles/all*: operación para a recuperación de todos os roles de usuario existentes na base de datos.

- **/internationalization:** baixo este path agrúpanse os recursos relacionados coa internacionalización:
 - */codes/*: operación para a recuperación de todos os códigos introducidos na base de datos. Devolve a lista dos textos nos tres idiomas soportados.

6.2 Frontend

O obxectivo do frontend é proporcionar unha interface ao usuario que lle permita ver e interactuar cos datos da aplicación. A tecnoloxía empregada para a programación do frontend é o framework web React, que usa unha arquitectura Single-page Application (SPA) [54]. Esta arquitectura consiste na reescritura da páxina inicial ou raíz da aplicación de maneira dinámica, dende o propio frontend, co contido que se necesite en cada momento, en lugar de recuperar e cargar novas páxinas do servidor.

Dentro do frontend podemos distinguir dúas capas: capa de acceso aos servizos e capa de interface de usuario. Comentámolas a continuación.

6.2.1 Capa de acceso aos servizos

Para as peticións ao backend empregase unha función `appFetch` cos seguintes argumentos:

- **path:** path para o envío da petición.
- **options:** indica o tipo de petición, un obxecto JSON que irá no corpo e o idioma.
- **onSuccess:** función que se invocará se o código de estado da resposta é 2XX. Se o corpo da resposta inclúe datos en formato JSON, a función recibirá como parámetro o obxecto correspondente.
- **onErrors:** función que se invocará se o código de estado da resposta é 4XX. Se o corpo da resposta inclúe datos en formato JSON, a función recibirá como parámetro o obxecto correspondente.

Se o código de estado é 5XX ou non se pode enviar a petición ou recibir a resposta, abrírase indirectamente unha ventá modal de erro. Ademais xestiónase automaticamente o envío do token JWT. Para o acceso a API REST creouse un ficheiro js por cada controlador, excepto para controlador da internacionalización e o dos tipos de atributos, cuxas peticións se engaden xunto coas peticións para os elementos xeométricos.

6.2.2 Capa da interface de usuario

Para a capa da interface de usuario empregouse un desenvolvemento multi-modular, cada módulo contén o código da interface de usuario correspondente a un dos grupos de recursos expostos polos servizos REST. A maiores hai dous módulos máis: *app*, para o renderizado do esqueleto común da interface de usuario, e *common*, no que se inclúe código usable por distintos módulos. Cada módulo contén as seguintes partes:

- ***components***: compoñentes propios do módulo.
- ***actions.js***: cada vez que se quere modificar o estado hai que enviar un obxecto acción ao store.
- ***actionTypes.js***: tipos de accións.
- ***reducer.js***: contén as funcións encargadas de producir un novo estado despois de executada unha acción.
- ***selectors.js***: contén as funcións que aceptan un estado Redux e devolve un dato que é derivado dese estado.
- ***index.js***: define o que se exporta do módulo (convención).

Unha das partes importantes do deseño é planificar ben a navegación entre os distintos compoñentes da web. Para facilitar isto decidiuse empregar a librería React Router [27]. O compoñente Router permite manter a interface de usuario sincronizada coa URL que aparece no navegador e inxecta aos compoñentes propiedades relativas ao enrutamento.

```

1  <Router>
2    <header>
3      <Header />
4    </header>
5    <main role="main" className="flex-shrink-0">
6      <Body />
7    </main>
8    <Footer
9      footerInfo={footerInfo ? footerInfo : "&copy;2020"}
10     ...
11     footerText8={footerText8 ? footerText8 : ""}
12  />
13 </Router>

```

O elemento *Switch* renderizará o primeiro *Route* que concorde co *path* seleccionado, este renderizará o compoñente *component*. Aquelas rutas que teñan *background* serán compoñentes con ventás modais, isto fará que de fondo se vexa o renderizado anteriormente. Se *background* non existe a ventá modal non se mostrará.


```

1 <Switch location={background || location}>
2   <Route exact path="/" component={OpenLayersMap} />
3   ...
4   <Route component={Component404} />
5 </Switch>
6 {background && <Route exact path="/admin/elementType/add"
7 component={AddGeometricElementType} />}
8 ...
9 {background && <Route exact path="/users/signUp"
  children={<SignUp />} />}

```

Módulos

A continuación detallanse compoñentes planificados para cada módulo:

- **Módulo *app*:**

- *App*: é o compoñente encargado de mostrar o esqueleto da aplicación, renderiza o *Header*, o contedor do *Body* e o *Footer*.
- *Body*: será o encargado de renderizar os compoñentes en función do path.
- *Component404*: é o compoñente que se mostrará se un path non existe.
- *Footer*: é o compoñente que renderizará o pé de páxina.
- *InternationalizationRender*: compoñente empregado para a internacionalización dos distintos tipos de códigos.

- **Módulo *common*:**

- *ErrorDialog*: mostrará unha ventá modal cando a resposta do backend sexa 5XX.
- *Errors*: mostrará unha alerta cos erros que chegan do backend que teñan un código de resposta do tipo 4XX.

- **Módulo *configurationParameter*:**

- *AllConfigurationParameters*: compoñente encargado de renderizar unha táboa que contén todos os parámetros de configuración paxinados. Será accesible só para usuarios administradores dende o menú.
- *DraftJsRender*: compoñente que renderizará unha instancia de Draft.js, librería encargada de modificar os textos con formato. Será empregado no compoñente *UpdateConfigurationParameter* se o parámetro a modificar é do tipo *FORMATTED_TEXT*.

- *UpdateConfigurationParameter*: será o encargado de renderizar unha ventá modal para modificar o valor dun parámetro de configuración. Será accesible dende o compoñente *AllConfigurationParameters*.

- **Módulo *geometricElements*:**

- *AddAttributeType*: será o encargado de renderizar unha ventá modal que permita modificar ou engadir un tipo de atributo. Será accesible dende o compoñente *AllAttributeType*.
- *AddGeometricElement*: será o encargado de renderizar un panel que permita engadir ou modificar un elemento xeométrico. Será accesible dende o compoñente *OlMapTiledWms*.
- *AddGeometricElementType*: compoñente encargado de renderizar unha ventá modal que permita engadir ou modificar un tipo de elemento xeométrico. Será accesible dende o compoñente *AllGeometricElementType*.
- *AllAttributeType*: compoñente encargado de renderizar unha táboa que contén todos os tipos de atributos paxinados. Será accesible para os administradores dende o menú da aplicación.
- *AllGeometricElementType*: compoñente encargado de renderizar unha táboa que contén todos os tipos de elementos xeométricos paxinados. Será accesible para os administradores dende a cabeceira.
- *DeleteAttributeType*: encárgase de renderizar unha ventá modal que eliminará, despois de confirmar, un tipo de atributo. Será accesible dende o compoñente *AllAttributeType*.
- *DeleteGeometricElement*: encárgase de renderizar unha ventá modal que eliminará, despois de recibir confirmación por parte do usuario, un elemento xeométrico. Será accesible dende o compoñente *GeometricElementsByGeometricElementType*.
- *DeleteGeometricElementType*: encárgase de renderizar unha ventá modal que eliminará, despois de confirmar, un tipo de elemento xeométrico. Será accesible dende o compoñente *AllGeometricElementType*.
- *GeometricElementFileGallery*: encárgase de renderizar unha ventá modal cunha galería das fotos asociadas a un elemento xeométrico. Será accesible dende o compoñente *GeometricElementsByGeometricElementType*.
- *GeometricElementsByGeometricElementType*: encárgase de renderizar unha táboa que contén os elementos xeométricos dun tipo dado paxinados. Será accesible

dende o varias opcións do menú. Para usuarios anónimos empregárase para mostrar só aqueles elementos que sexan públicos. Para usuarios rexistrados empregárase para mostrar os elementos públicos e os elementos dos que o usuario é autor. Para administradores empregárase para mostrar todos os elementos dun mesmo tipo ou aqueles dos que son autores.

- *OlMapTiledWMS*: compoñente encargado de renderizar o visor xeográfico.
- *OpenlayersCustomControllers*: contén os compoñentes que serán encargados de renderizar os seguintes controis sobre o visor xeográfico: mostrar a información dun elementos xeométrico a través dun *popup*, engadir un novo elemento xeométrico, medir áreas, medir distancias e facer zoom a unha zona do visor.
- *OpenLayersMap*: será o encargado de renderizar conxuntamente os compoñentes *OlMapTiledWms* e *AddGeometricElement*. Será accesible dende o menú e dende o compoñente *GeometricElementsByGeometricElementType*.

• **Módulo users:**

- *AllUserAccounts*: compoñente encargado de renderizar unha táboa con todos os usuarios paxinados (só os administradores poderán acceder a este compoñente).
- *ChangePassword*: compoñente encargado de mostrar un formulario para o cambio de contrasinal do usuario. Será accesible dende o menú se o usuario está autenticado.
- *LockUserAccount*: compoñente encargado de mostrar unha ventá modal que permitirá bloquear unha conta de usuario. Só os administradores poderán acceder a este compoñente.
- *Login*: renderizará unha ventá modal que contén un formulario para iniciar sesión na aplicación. Será accesible dende o menú se o usuario non está autenticado
- *Logout*: compoñente encargado de pechar sesión. Eliminará o token JWT do *sessionStorage*. Unha vez feito fará unha redirección á páxina principal. Será accesible dende o menú se o usuario está autenticado.
- *SignUp*: compoñente encargado de mostrar unha ventá modal que contén un formulario para o rexistro dun novo usuario. Será accesible dende a ventá modal de iniciar sesión.

6.3 QGIS server

O obxectivo do uso deste servidor é publicar os datos xeoespaciais inseridos na base de datos (elementos xeométricos) mediante servizos web, para que poidan ser incluídos no visor

xeográfico do frontend. Nun futuro podería valorarse facer dispoñibles estes servizos tamén para clientes externos. Os dous servizos empregados son WFS e WMS:

- **WMS:** proporciona unha interface HTTP para solicitar mapas. Nunha petición típica indícanse as capas de información a incluír, o formato da imaxe de mapa a devolver e zona que se quere visualizar.
- **WFS:** proporciona unha interface HTTP para solicitar información sobre os atributos de obxectos xeorreferenciados.

Como xa se comentou en seccións previas, QGIS Server execútase *baixo* Apache e precisa o módulo de apache mod_fcgid para funcionar. Tamén precisa dun *X Server*. Neste caso empregouse Xvfb.

Para que QGIS Server sirva os datos, necesitamos crear un proxecto de QGIS e configurar nel as capas de información que debe servir. Aínda que na base de datos todos os elementos xeométricos se atopan na mesma táboa, como pode haber elementos con xeometría de tipo punto, de tipo liña e de tipo polígono, para que QGIS poida servir ben os datos, créanse tres capas no servidor, unha por cada tipo de xeometría.

No visor xeográfico, sen embargo, non se terá a mesma configuración de capas que no servidor senón que se terán unha especie de capas *virtuais*. Isto é debido a que, no visor xeográfico:

- Non se queren ter todos os elementos na mesma capa, senón que se quere poder seleccionar que tipos de elementos se queren visualizar e para iso se precisa ter unha capa para cada tipo de elemento.
- Quérese que para un tipo de elemento dado, todos os obxectos dese tipo si estean na mesma capa, independentemente de se a xeometría e de tipo punto, liña ou polígono.
- Quérese ter unha capa distinta para os elementos creados polo propio usuario.

Así, os nomes das capas do visor teñen a forma [MY_][ELEMENT-TYPE-CODE] sendo [MY_] opcional. Como xa se comentou, dende o frontend non se fan as peticións directamente a QGIS Server senón que se pasa por un proxy. É no proxy onde se *traducen* as capas do visor ás capas do servidor. Por exemplo, cando ao proxy lle chega unha petición WMS dunha capa MY_CODEA, o proxy fai o seguinte

- Sobrescribe o parámetro LAYERS da petición, pasando a incluír nese parámetro as tres capas existentes no servidor.
- Engade ou sobrescribe o parámetro FILTER da petición pasando a incluír un filtro polo tipo de elemento (CODEA), e un filtro polo id do usuario.

Para estruturar a información que se quere ofrecer sobre un elemento cando un usuario fai clic sobre el no visor xeográfico foi necesario deseñar e crear unha **vista na base de datos**. Esta necesidade vén dada porque, como xa se ten explicado, os valores dos atributos dinámicos dos elementos xeométricos están almacenados nunha táboa distinta da propios elementos. Para que QGIS poida incluír eses valores como información sobre os elementos é necesario programar un *SELECT*, a vista, que xunte a información de ambas táboas. As columnas da vista son as seguintes:

- *id*: o id do elemento xeométrico.
- *geom*: a xeometría do elemento.
- *value*: columna que agrega nun *string* de tipo JSON todos os valores dos atributos dinámicos do elemento xeométrico. Por exemplo: `[{"INFRASTRUCTURE_ZONE": ["aaa", "bbb"]}, {"PROVIDED_ECOSYSTEM_SERVICES": ["ccc", "ddd"]}]`.
- *is_public*: indica se o elemento é público ou non.
- *is_reviewed*: indica se o elemento está revisado ou non.
- *internal_comments*: observacións internas, só visible polos administradores.
- *geometric_element_type_id*: id do tipo de elemento xeométrico ao que pertence o elemento.
- *geometric_element_type*: contén o código do tipo de elemento xeométrico ao que pertence o elemento.
- *user_account_id*: contén o id do usuario que creou o elemento.

Na figura 6.11 podemos ver o resultado dunha petición WFS:

```

type: FeatureCollection
  ▼ bbox: [...]
    0: -8.459691
    1: 43.333021
    2: -8.429866
    3: 43.35098
  ▼ features: [...]
    ▼ 0: {}
      ▼ bbox: [...]
        0: -8.458408
        1: 43.341301
        2: -8.451032
        3: 43.344683
      ► geometry: {}
        id: geometric_element_polygon.3
      ▼ properties:
        description: aaaaaaaaaa
        geometric_element_type: PROBA_TYPE
        geometric_element_type_id: 1
        id: 3
        internal_comments: aaaaaa
        is_public: false
        is_reviewed: false
        user_account_id: 1
      ▼ value: [...]
        ▼ 0: {}
          ▼ PROBA: [...]
            0: aaaaaa aaaaaa
          type: Feature

```

Figura 6.11. Resultado dunha petición WFS.

6.4 Proxy

O proxy actúa de intermediario en todas as peticións ao servidor xeográfico, aplicándolles os cambios e filtros pertinentes. Aínda que para o proxy podería *montarse* nunha aplicación independente, neste caso, para simplificar a arquitectura, decidiuse integralo na propia aplicación web do backend. Para a implementación fixose uso de Zuul [51].

Unha das manipulacións que se fai no proxy e á conversión xa comentada na sección anterior das capas do visor xeográfico ás capas do servidor. Outra das cousas para as que serve o proxy e para evitar que un usuario acceda a elementos xeométricos que non ten permiso para ver. Isto faise tamén engadindo novos filtros ás peticións WMS e WFS en función do usuario que fai a petición e do seu rol:

- Se se trata dun usuario anónimo só se lle devolven os elementos nos que o atributo *isPublic* é *true*.
- Se se trata dun usuario rexistrado só se lle devolven os elementos nos que o atributo

isPublic é *true* ou ben nos que o creador é el mesmo.

Na figura 6.12 podemos ver un diagrama do funcionamento do proxy. O frontend fará unha petición dirixida ao servidor xeográfico que o proxy interceptará. O *PreFilter* fará as modificacións necesarias (dependendo das capas e o usuario que chega no token JWT) para filtrar a petición final que irá redirixida ao QGIS server. O QGIS server procesa esa petición e envía a resposta ao Zuul, que á súa vez a envía ao *frontend*.

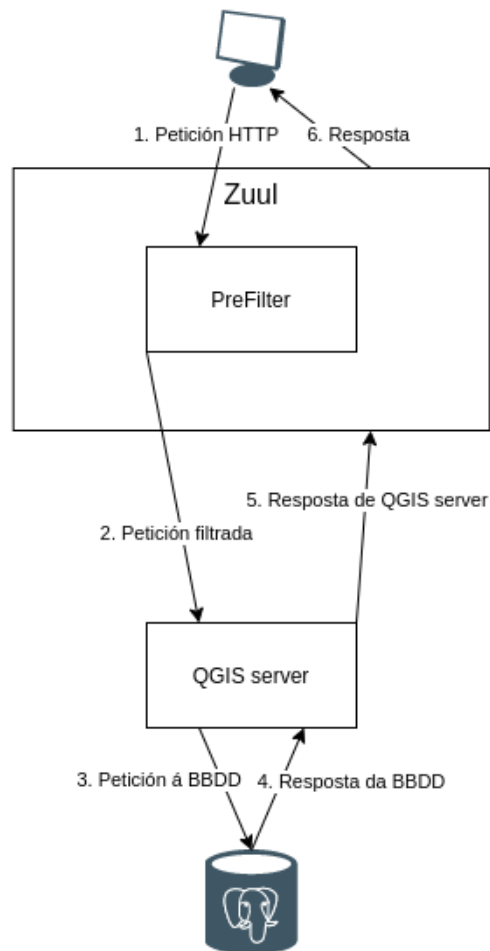


Figura 6.12. Diagrama do funcionamento do proxy.

Implementación

Nese capítulo comentamos algúns detalles sobre a implementación do sistema.

7.1 Backend

O backend é un proxecto Maven, polo tanto a súa estrutura de directorios é a estándar para este tipo de proxectos. A continuación lístanse os directorios e arquivos máis relevantes.

- ***src/main/java***: directorio no que se encontra todo o código fonte da aplicación.
- ***src/main/resources***: directorio no que se encontran os ficheiros de configuración da aplicación.
- ***src/test/java***: directorio no que se encontra o código fonte dos tests de unidade da aplicación.
- ***src/test/resources***: directorio no que se encontran os ficheiros de configuración dos tests.
- ***pom.xml***: arquivo onde se describe o proxecto, inclúense as dependencias e instrucións de compilación e empaquetado.

Na figura 7.1 podemos observar todos os paquetes nos que se divide o backend. No paquete *model* atoparemos os paquetes *dao* e *entities*, neles atópase a capa de acceso a datos. No paquete *services* están os servizos e as súas fachadas. Por outra banda, no paquete *rest*, atoparémonos o paquete *controllers* onde están os controladores REST e en *zuulFilters* atoparémonos os filtros do proxy. No paquete *converter* só atoparemos unha clase que é a encargada de deserializar os GeoJSON. O paquete *dtos* contén os DTO, o paquete *exceptions* que contén as clases das excepcións do modelo, e por último *common* que contén o mapeo das excepcións do modelo ademais da lóxica necesaria para procesar os token JWT.

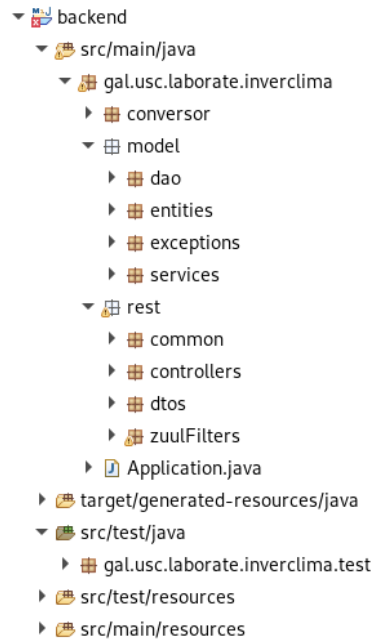


Figura 7.1. Estructura na que se divide o backend.

Inyección de dependencias

Para as facer uso dos DAO nas clases dos servizos, estes están dispoñibles mediante a inyección de dependencias coa notación `@Autowired` de Spring. Na figura 7.2 podemos ver os DAO inxectados no servizo `GeometricElementService`.

```

@Autowired
GeometricElementTypeAttributeTypeDao geometricElementTypeAttributeTypeDao;

@Autowired
GeometricElementTypeDao geometricElementTypeDao;

@Autowired
GeometricElementDao geometricElementDao;

@Autowired
GeometricElementFileDao elementFileDao;

@Autowired
CodeService codeService;

@Autowired
AttributeService attributeService;

@Autowired
UserAccountService userAccountService;

@Autowired
AttributeValueDao attributeValueDao;

@Autowired
AttributeTypeDao attributeTypeDao;

@Autowired
CodeGroupDao codeGroupDao;

```

Figura 7.2. Uso da anotación `@Autowired` no servizo `GeometricElementService`.

Transaccionalidade

No caso de que unha operación do servizo necesite máis dun acceso á base de datos, débense realizar todos estes accesos dentro dunha mesma transacción, co fin de cumprir coas propiedades Atomicity, Consistency, Isolation and Durability (ACID). Nestes casos empregase a anotación `@Transactional` para abrir e pechar transaccións, de maneira que non teñan que ser abertas e pechadas á man en cada método. Na figura 7.3 podemos ver o uso desta anotación.

```
@Override
@Transactional
public void deleteElementType(Long id, String role) throws InstanceNotFoundException, PermissionDeniedException, ElementWithInstancesException {
    if(!role.equals("ADMIN")) {
        throw new PermissionDeniedException();
    }

    GeometricElementType geometricElementType = findElementTypeById(id);

    Page<GeometricElement> elementsPage = geometricElementDao.findByGeometricElementType(geometricElementType, PageRequest.of(0, 10));

    if(elementsPage.hasContent()) {
        throw new ElementWithInstancesException("project.entities.GeometricElementType", id);
    }

    geometricElementDao.delete(geometricElementType);
}
```

Figura 7.3. Uso da anotación `@Transactional` nun método do servizo `GeometricElementService`.

7.2 Frontend

O frontend é un proxecto React, e o seu esqueleto foi creado coa utilidade *create-react-app*, tendo, así, a disposición dos directorios habitual en aplicacións React, que podemos ver na figura 7.4.

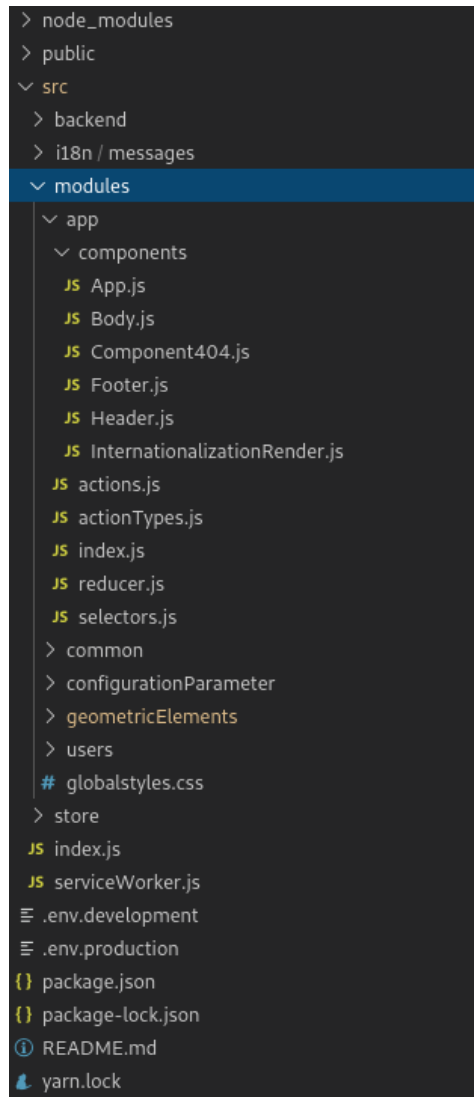


Figura 7.4. Estrutura do código do frontend.

- **public:** neste directorio encóntranse todos os ficheiros estáticos da aplicación. O ficheiro *index.html* é o modelo (esqueleto) onde se renderizará a aplicación.
- **src:** neste directorio encóntrase todo o código fonte:

- **backend**: capa de acceso aos servizos.
 - **i18n**: ficheiros de mensaxes internacionalizadas.
 - **modules**: capa da interface de usuario.
 - **store**: creación do store Redux.
 - **index.js**: é o ficheiro principal que se encarga de inicializar a aplicación React. A través da función `ReactDOM.render` renderiza o compoñente `App` no elemento con id `root` que se atopa no ficheiro `index.html`. Tamén é o encargado de inicializar o `store` de Redux.
- **package.json**: a súa función é análoga ao do `pom.xml` dun proxecto Maven, pero coa sintaxe JSON.

Internacionalización

Para a internacionalización da aplicación emprégase a a librería React Intl. Entre outras cousas, permite a internacionalización de mensaxes, números e datas a través dos seus compoñentes. En `/src/i18/messages` atopámonos os ficheiros de mensaxes, hai un por cada idioma. Cada ficheiro contén un obxecto JavaScript cos códigos para a internacionalización das mensaxes. O formato é o seguinte:

- `messages_en.js`

```
1   export default {  
2     ...  
3     'project.users.password': 'Password',  
4     ...  
5   }  
6
```

7.3 Proxy

Como xa se comentou, o proxy está implementando con Zuul e está integrado dentro da aplicación web do backend. Para configurar Zuul e programar os cambios a realizar polo proxy é suficiente con:

- Empregar a anotación `@EnableZuulProxy` na clase principal da aplicación (`Application`).
- Configurar as rutas de captura e redirección nun **ficheiro properties** (`src/main/resources/application.properties`). Aí indícase que todas as peticións que cheguen ás rutas do tipo `/mapserver/**` se redirixan á URL interna do visor xeográfico.

- Crear unha subclase de **ZuulFilter** (*src/main/java/gal.usc.laborate.inverclima.rest.zuulFilters/Prefilter.java*) e realizar aí a manipulación das peticións, engadindo á petición os filtros por usuario e/ou visibilidade antes de reenviar a petición ao servidor xeográfico. Dependendo do tipo de petición recibida engádense os parámetros necesarios para filtrar na URL (con WMS), ou modifícase o XML do corpo da petición (con WFS).

Capítulo 8

Probas

Neste capítulo comentamos os distintos tipos de probas realizadas: de unidade, end to end e de aceptación.

8.1 Probas unitarias con JUnit

No backend probáronse as distintas operacións dos servizos existentes. Para realizar as probas creouse e empregouse unha base de datos específica e usouse de maneira que a execución dun test non afecte á execución do resto dos tests. Para que o estado da base de datos ao final da execución dun test sexa o estado inicial emprégase a anotación *@Transactional* de Spring, que converte cada test nunha transacción que fai un *rollback* ao chegar ao final do mesmo.

Para comprobar unha suficiente cobertura do código por parte dos test empregouse o plugin para Eclipse **EclEmma** [40]. Mirouse que a cobertura de cada servizo fose cando menos dun 80% das instrucións que compoñen o servizo. Na figura 8.1 podemos ver os valores de cobertura para cada un dos servizos.











Element	Coverage	Covered Instructions	Missed Instructions
▶  ElementServiceImpl.java	 83,1 %	793	161
▶  CodeServiceImpl.java	 91,1 %	357	35
▶  AttributeServiceImpl.java	 86,1 %	253	41
▶  UserAccountServiceImpl.java	 96,3 %	258	10
▶  ConfigurationParameterServiceImpl.java	 87,3 %	178	26

Figura 8.1. Cobertura dos test dos servizos do backend.

8.2 Probas end to end con Cypress

Para as probas dende o frontend empregouse Cypress. Como xa se explicou no apartado de [Ferramentas para probas](#), é un framework de teste do frontend. Aínda que permite máis tipos de test empregouse neste caso para os tests End-to-End [59], aqueles que comprobamos que o fluxo dunha aplicación funciona correctamente de principio a fin. Creouse un proxecto npm independente onde se instalou Cypress. Podemos ver a estrutura do proxecto na figura 8.2, onde na carpeta *integration* están todos os tests e no ficheiro *support/commands.js* podemos meter algúns comandos para ampliar os que xa veñen por defecto en Cypress.

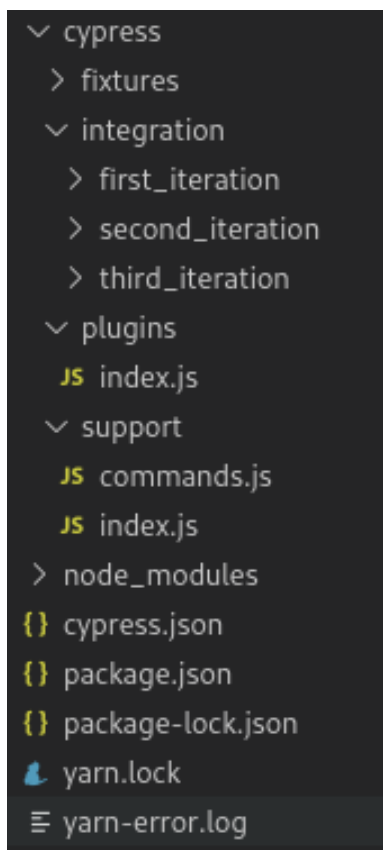


Figura 8.2. Estrutura do proxecto de probas con Cypress.

Para este cometido creouse especialmente un ficheiro que contén os datos iniciais de proba. Cada vez que se executan os tests quedan na base de datos restos da execución dos mesmos, polo tanto é necesario volver ao estado inicial da base de datos en cada execución.

Na figura 8.3 podemos ver un exemplo de como serían os pasos para visualizar os elementos xeométricos públicos existentes na base de datos. Para facilitar o desenvolvemento das

probas, a cada elemento do DOM [60] engadíuselle un id específico.

```
describe('GIVEN: the app url', () => {
  const sutUrl='https://localhost:3000';

  context('WHEN: I visit the page', () => {
    before(() => cy.visit(sutUrl));

    let dropdown;

    it('THEN: should have a dropdown', () => {
      dropdown = cy.get('#geometricElementsByType')
      dropdown.should('exist');
      dropdown.click();
    });

    let dropdownElements;

    it('THEN: should have a list of geometric element type', () => {
      dropdownElements = cy.get("#geometricElementsByType + div")
      dropdownElements.should('have.class', 'show');
    });

    it('THEN: should have at least one geometric element type', () => {
      cy.get('#elementType0').should('exist');
    });
  });

  context('WHEN: I select the first geometric element type', () => {
    before(() => cy.get('#elementType0').click());

    it('THEN: should have a table', () =>{
      cy.get('.react-bootstrap-table > table').should('exist');
    })
  });
});
```

Figura 8.3. Exemplo do código dun test realizado con Cypress.

A secuencia do test é a seguinte:

1. Visita a URL da aplicación.
2. Comproba que existe un elemento do DOM co id indicado e fai clic nel, neste caso un *dropdown*.
3. Comproba que ese *dropdown* foi despregado.
4. Fai clic no primeiro elemento do *dropdown*.

5. Comproba se existe unha táboa no DOM.

A continuación enuméranse as probas que se implementaron e executaron con Cypress:

1. Crear conta de usuario: probouse a creación dunha conta de usuario.
2. Iniciar sesión: probouse a iniciar sesión cun usuario non existente e cun usuario si existente na base de datos.
3. Cambiar contrasinal: probouse o cambio de contrasinal dun usuario, comprobando que se mostran os erros correctamente.
4. Ver elementos públicos: probouse o acceso a elementos públicos dun tipo como un usuario non autenticado e como un usuario rexistrado.
5. Cambiar idioma: probouse que se cambia o idioma da aplicación como un usuario anónimo.
6. Ver fotos asociadas a un elemento xeométrico: probouse a visualización das fotos asociadas a un elemento xeométrico como un usuario non autenticado.
7. Ver a lista dos ficheiros asociados a un elemento xeométrico: probouse a ver a lista dos ficheiros asociados a un elemento xeométrico como un usuario anónimo.
8. Ver a lista das contribucións dun usuario: probouse o acceso á lista dos elementos xeométricos dun tipo creados por un usuario autenticado.
9. Ver todos os elementos xeométricos: probouse o acceso á lista de todos os elementos xeométricos dun tipo como administrador.
10. Descargar ficheiro asociado a un elemento xeométrico: probouse a descargar un ficheiro asociado a un elemento xeométrico que non é posible descargar. Debido a limitacións coa librería non se puido comprobar a descarga dun ficheiro que si é posible descargar.
11. Eliminar un elemento xeométrico: probouse a eliminación dun elemento xeométrico como un usuario rexistrado.
12. Realizar zoom a unha área debuxada: probouse como un usuario non autenticado a realizar zoom no visor xeográfico.
13. Realizar unha medición dunha área: probouse como un usuario non autenticado a realizar unha medición dunha área no visor xeográfico.
14. Realizar unha medición dunha distancia: probouse a medir unha distancia no visor xeográfico como un usuario non autenticado.

15. Engadir un elemento xeométrico dende a lista de contribucións, dende o visor xeográfico e dende a barra de navegación: probouse a inserción dun elemento xeométrico dende tres lugares distintos como un usuario rexistrado.
16. Centrar un elemento xeométrico no visor dende as táboas e dende o visor: probouse que se centra o visor entorno ao elemento xeométrico desexado.
17. Consultar elementos públicos dende o visor xeográfico: probouse a consultar os elementos públicos existentes no sistema dende o visor xeográfico.
18. Modificar un elemento xeométrico dende a lista de elementos e dende o visor xeográfico: probouse a modificar un elemento xeométrico dende dous lugares distintos.
19. Ver e modificar usuarios do sistema: probouse a ver a lista dos usuarios que figuran no sistema e a a modificar o seu estado (bloquear/desbloquear) como administrador.
20. Ver e modificar os parámetros de configuración: probouse a visualización dos parámetros de configuración existentes na base de datos e ademais a modificación do seu valor como administrador.
21. Ver, engadir, modificar e eliminar tipos de elementos: probouse a visualización, inserción, modificación e eliminación de tipos de elementos xeométricos como administrador.
22. Ver, engadir, modificar e eliminar tipos de atributos: probouse a visualización, inserción, modificación e eliminación de tipos de atributos.

8.3 Probas de aceptación

Na revisión final do sistema realizáronse unha serie de probas de aceptación manuais. Son as que se indican a continuación:

- **Como usuario anónimo.**
 - O usuario pode darse de alta introducindo un nome de usuario, un correo electrónico e un contrasinal.
 - * O contrasinal debe ser introducido dúas veces e os valores deben coincidir.
 - O usuario debe poder iniciar sesión introducindo o nome de usuario e o contrasinal.
 - * Se se introduce un contrasinal incorrecto, débesele amosar un aviso ao usuario e non se debe iniciar sesión.

-
- O usuario debe poder consultar o visor xeográfico e nel deben aparecer os elementos xeométricos públicos e non deben aparecer elementos que non está publicados.
 - Sobre o visor xeográfico, o usuario debe poder:
 - * Facer zoom por encadre.
 - * Medir lonxitudes.
 - * Medir áreas.
 - * Facer clic sobre un elemento xeométrico visible no mapa. A aplicación amosaralle a información pública sobre dito elemento nunha ventá *pop-up*.
 - Na *pop-up* aparecerá unha opción de centrar. Se o usuario preme nela, o visor xeográfico centrase no elemento.
 - * Seleccionar as capas de información xeográficas visibles, de entre as dispoñibles.
 - * Facer zoom á máxima extensión do mapa: no mapa debe pasar a visualizarse a zona configurada na base de datos da aplicación como a *máxima extensión*.
 - * Face clic sobre un elemento xeométrico visible no mapa:
 - A aplicación debe amosar unha ventá *pop-up* coas atributos do elemento.
 - No *pop-up* non deben aparecer as observacións internas.
 - No menú de *Táboas de elementos*, debe aparecer unha entrada para cada un dos tipos elementos existentes.
 - Se usuario preme nunha entrada do menú de *Táboas de elementos*, amósase unha táboa paxinada coas elementos públicos correspondentes ao tipo seleccionada.
 - Na táboa non deben aparecer atributos reservados a xestores.
 - **Como usuario rexistrado:**
 - O usuario pode cambiar o idioma no que quere que se presente a aplicación:
 - * Os texto da aplicación pasan a estar no novo idioma.
 - * Se pecha sesión e a volve a iniciar, o idioma corresponde ao último seleccionado.
 - No selector de capas do visor xeográfico aparece o grupo de capas *As miñas contribucións*, e, dentro del, aparece unha capa para cada tipo de elemento existente:
 - * Se o usuario desactiva unha capa de *As miñas contribucións*, desaparecen do visor os elementos dese tipo que foron creados por el e aínda non está publicados.

- No menú principal debe aparecer unha opción para *As miñas contribucións*. Dentro de *As miñas contribucións* debe aparecer unha entrada para cada un dos tipos elementos existentes.
- Se o usuario accede a unha táboa calquera de elementos, debe aparecer unha opción na parte superior da táboa para introducir un elemento xeométrico novo.
- No visor xeográfico aparece un control para realizar a introducir dun novo elemento xeométrico. O control está acompañado por un selector que se activa xunto co control no que o usuario pode e debe indicar que tipo de elemento desexa introducir.
- Tanto accedendo dende unha táboa de elementos como activando o control do visor, o usuario pode culminar o proceso de introdución dun novo elemento. Pode debuxar a xeometría do elemento no visor e pode cubrir os atributos obrigatorios mediante un formulario que se amosa á dereita do visor xeográfico.
 - * Se o usuario preme na opción de aceptar do formulario sen ter cuberto algún campo obrigatorio, amósaselle un aviso ao respecto.
- Se o usuario accede a algunha das táboas de *As miñas contribucións*, na columna de accións aparece a opción de modificar para os elementos que non están revisados e non aparece para os elementos que están revisados.
- Se, no visor xeográfico, o usuario activa o control de consulta de información e fai clic sobre un elemento visible no visor, se se trata dun elemento introducido por el, se aínda non foi revisado, no *pop-up* informativo está visible a opción de modificación. Noutro caso no debe estar visible.
- Tanto accedendo dende unha táboa de elementos como activando facendo clic sobre o visor, o usuario pode culminar o proceso de modificación dun dos seus elementos non revisados correctamente.
- Tanto nas táboas dos elementos de *As miñas contribucións*, para aqueles elementos que aínda non figuren como revisados amósase unha opción para eliminar o elemento.
 - * Antes de realizar a eliminación, pídeselle ao usuario que confirme que a quere realizar.
 - * O usuario pode culminar o proceso de eliminación correctamente. Unha vez eliminado un elemento, este deixa de aparecer na táboa.
- Se o usuario preme en pechar sesión, o estado da aplicación pasa a ser o correspondente a un usuario anónimo.

- **Como administrador:**

- A través da entrada *Usuarios* do menú de *Administración*, o administrador pode poden consultar unha táboa paxinada cos usuarios existentes.
- Na táboa de usuarios, para cada usuario, existe unha opción de modificación que abre un formulario modal para modificar o usuario correspondente.
 - * Se o administrador marca un usuario como bloqueado, se dito usuario tenta iniciar sesión, non se lle permite.
 - * Se o administrador cambia o contrasinal dun usuario, dito usuario pode iniciar empregando o novo contrasinal.
 - * Premendo na opción *Parámetros de configuración* do menú de *Administración* debe aparecer unha táboa paxinada cos parámetros de configuración existentes e os seus valores.
 - * En cada fila da táboa de parámetros de configuración debe existir unha opción para modificar o parámetro. Ao premer na opción debe aparecer un formulario modal no que introducir o novo valor do parámetro. Se o administrador preme en Aceptar, o valor do parámetro debe quedar correctamente gardado e aparecer actualizado na táboa de parámetros.
 - * Cando se cambia, por exemplo, o parámetro de configuración correspondente ao título da aplicación, ao recargar a mesma, esta xa aparece co novo título.
 - * Premendo na opción *Tipos de atributos* do menú de *Administración* debe aparecer unha táboa paxinada cos tipos de atributos existentes.
 - * Na táboa de tipos de atributos, na parte superior, debe aparecer unha opción para introducir un novo tipo. Ao premer na opción debe aparecer un formulario modal no que introducir os datos do novo tipo. Se o administrador preme en Aceptar, o novo tipo debe quedar correctamente gardado e aparecer na táboa de tipos de atributos.
 - * En cada fila da táboa de tipos de atributos debe existir unha opción para modificar o tipo. Ao premer na opción debe aparecer un formulario modal no que introducir os novo valores do tipo. Se o administrador preme en Aceptar, os valores deben quedar correctamente gardados e aparecer actualizados na táboa de tipos de atributos. Non se debe permitir modificar o código do tipo de atributo.
 - * Premendo na opción *Tipos de elementos* do menú de *Administración* debe aparecer unha táboa paxinada cos tipos de elementos xeométricos existentes.
 - En cada fila da táboa existirán opcións para eliminar e modificar o tipo de elemento, así como para ver a táboa paxinada con todos os elementos do tipo correspondente a cada fila.

- A opción de eliminar debe abrir un cuadro de diálogo pedindo a confirmación da eliminación. Se o usuario preme ene aceptar e existen instancias do tipo en cuestión, avísase ao usuario de que son se pode realizar a eliminación por dito motivo.
- A opción de modifica abrirá unha ventá modal cun formulario cos dato do tipo de elemento. No caso de que xa existan elementos dese tipo, permitirase modificar o nome do tipo de elemento, pero non os seus atributos.

Conclusións e traballo futuro

9.1 Conclusións

Pode dicirse que os resultados do traballo son satisfactorios. Creouse un sistema que cumpre cos obxectivos que motivaron o proxecto. A aplicación desenvolvida pode ser aplicada a distintos procesos colaborativos de participación pública. De feito, nas vindeiras semanas está previsto aplicar esta ferramenta no proxecto Inverclima [61], en colaboración co Laboratorio do Territorio (LaboraTe) da Universidade de Santiago de Compostela. Trátase dun proxecto onde se recompilará información ambiental considerando en principio tres tipos de elementos: servizos ecosistémicos, infraestruturas verdes e zonas de risco.

Con respecto ao tempo finalmente empregado respecto ao tempo estimado na planificación, pode observarse que houbo unha infraestimación en varias das tarefas. Isto foi debido, principalmente, a que o programador non estaba familiarizado coa maioría das ferramentas empregadas no desenvolvemento e, a maiores da familiarización inicial coas mesmas, houbo que empregar un tempo importante na resolución de bugs e de problemas de integración de compoñentes.

Cabe mencionarse tamén que dentro do Grupo de Arquitectura de Computadores, no seo do cal xurdiu a idea do proxecto, o traballo realizado serviu para validar na práctica o funcionamento dalgunhas ferramentas que ata momento non se empregaran nos proxectos do grupo. Entre elas están QGIS Server e a última versión de OpenLayers. As conclusións son satisfactorias e prevese o seu uso en futuros proxectos. A experiencia e coñecementos adquiridos polo alumno é outro dos resultados a destacar. Cóntase con que o alumno poida seguir colaborando co Grupo de Arquitectura de Computadores, ampliando este sistema e colaborando noutros proxectos.

9.2 Trabajo futuro

Entre as posibles liñas de traballo futuro podemos mencionar as seguintes:

- **Novas posibilidades de configuración de formularios:** actualmente os administradores poden definir os atributos de cada tipo de elemento e, por tanto, os campos dos formularios. Tamén pode indicar a orde en que deben colocar, pero por exemplo, non se poden configurar *camións alternativos* no fluxo dos formularios en función do valor que se introduza, por exemplo, nun selector.
- **Ampliación das capacidades de personalización da interface de usuario:** na práctica observouse que as capacidades ofrecidas por Draft.js [29] nalgúns aspectos son limitadas. Sería útil para os administradores poder facer, por exemplo, un deseño máis completo do pé de páxina do sitio, sen estar constringidos a un conxunto de *placeholders* predefinidos. Pode valorarse, para iso, o uso de TinyMCE [62].
- **Ampliación das capacidades do visor xeográfico:** en particular, sería interesante permitir aos usuarios o centrado por provincia, concello ou parroquia. Tamén se podería valorar a implementar a busca por topónimos de entidades de poboación.
- **Permitir configurar o estilo de visualización de cada tipo elemento xeométrico:** nunha opción menos ambiciosa podería permitírselle aos administradores indicar o nome do estilo a usar para cada tipo de elemento (estando os estilos predefinidos no servidor xeográfico). A opción ambiciosa sería incluír na propia aplicación un deseñador gráfico de estilos de representación de liñas, puntos e polígonos.
- **Permitir engadir novas capas de información xeográfica dende a interface de usuario:** trataríase de que, ademais das capas prefiguradas que se amosan no visor, os administradores poidan crear novos grupos de capas e engadir ao visor novas capas WMS xa existentes en servidores externos.
- **Funcionalidades de importación e exportación de datos:** en concreto sería útil habilitar a exportación a formato Shape das táboas de elementos, para facilitar a explotación de datos en ferramentas GIS de escritorio con capacidades avanzadas de análise.

Apéndices

Guía de instalación

A continuación descríbense os pasos para a posta en marcha do sistema en produción. Danse as instrucións para a instalación nun sistema operativo Debian Buster.

A.1 Instalación de software de base

- Instalar Apache xunto cos módulos fcgid e jk:

```
$ sudo apt -y install apache2 libapache2-mod-fcgid libapache2-mod-jk
```

- Instalar PostgreSQL e PostGIS:

```
$ sudo apt -y install postgresql-11 oidentd postgis
```

- Instalar Tomcat:

- *\$ sudo apt -y install default-jdk tomcat9*

- Instalar QGIS server

```
$ sudo apt -y install qgis-server
```

A.2 Configuración dos servidores

- Habilitar HTTPS en Apache:

```
$ a2enmod ssl
```

```
$ a2enmod soache_shmcb
```

```
$ a2ensite default-ssl
```

- Configurar o módulo *mod_jk*:

- Cambiar o contido de `/etc/libapache2-mod-jk/workers.properties` por:


```
ps=/
worker.list=tomcat
workers.java_home=$JAVA_HOME
worker.tomcat.port=8009
worker.tomcat.host=ip6-localhost
worker.tomcat.type=ajp13
```
- En `/etc/apache2/mods-available/jk.load` engadir o seguinte contido:


```
JkWorkersFile /etc/libapache2-mod-jk/workers.properties
JkLogFile /var/log/apache2/mod_jk.log
JkLogLevel info
JkShmFile /var/log/jk-runtime-status
```
- Engadir ao ficheiro `/etc/apache2/apache2.conf`:


```
JkOptions +ForwardSSLCertChain
JkMount / tomcat
JkMount /* tomcat
```
- Engadir `JkMountCopy On` no bloque `<VirtualHost>` dos ficheiros `/etc/apache2/sites-available/000-default.conf` e `/etc/apache2/sites-available/default-ssl.conf`.
- Configurar de Tomcat:
 - No ficheiro `/etc/tomcat9/server.xml`:

No apartado `<Host>` debemos meter dúas directivas para que React poida cargar os ficheiros estáticos:

```
<Context path="" docBase="inverclima">
<WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>

<Context path="/ROOT" docBase="/ROOT">
<WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>
```

Débase descomentar o `<Connector>` que se encontra no comentario Define AJP e engadimos `secretRequired="false"`:

```
<Connector protocol="AJP/1.3"
address="::1"
```

```
port="8009"
secretRequired="false"
redirectPort="8443" />
```

- Configurar QGIS Server:
 - Copiar o ficheiro qgisserver.conf en *sites-available*

```
$ sudo cp qgisserver.conf/etc/apache2/sites-available
```
 - Habilitar o módulo fcgid e habilitar o sitio de QGIS Server

```
$ sudo a2enmod fcgid
$ sudo a2ensite qgisserver.conf
$ sudo service apache2 reload
```
 - Engadir o porto 8000 para QGIS Server no arquivo */etc/apache2/ports.conf*

```
Listen 8000
```
 - Configurar o servidor *xvfb*:
 - * Crear o ficheiro de configuración en */etc/systemd/system/xvfb.service* co seguinte contido:


```
[Unit]
Description=X Virtual Frame Buffer Service After=network.target
[Service]
ExecStart=/usr/bin/Xvfb :99 -screen 0 1024x768x24 -ac +extension GLX +render -noreset
[Install]
WantedBy=multi-user.target
```
 - * Habilitar o servizo creado:

```
$ systemctl enable xvfb.service
```
 - * Arrancar o servizo:

```
$ systemctl start xvfb.service
```

A.3 Creación da base de datos e configuración do acceso

- Acceder a PostgreSQL como superusuario:

```
$ sudo -u postgres psql
```
- Crear os usuarios *inverclima* e *www-data*:

```
$ CREATE USER inverclima WITH PASSWORD ContrasionalSeguro;
```

```
$ CREATE USER www-data;
```

- Crear a base de datos e engadir as extensións necesarias:

```
$ CREATE DATABASE inverclima;
```

```
$ \c inverclima
```

```
$ CREATE EXTENSION postgis;
```

```
$ CREATE EXTENSION pgcrypto;
```

```
$ \i create_tables.sql;
```

```
$ \i insert_common_data.sql;
```

- Dar acceso aos usuarios anteriores. Para iso, no ficheiro `/etc/postgresql/11/main/pg_hba.conf`, engadir:

```
host all www-data ::1/128 ident
```

```
host all all ::1/128 md5
```

A.4 Instalación da aplicación

- Acceder ao directorio do código fonte onde se atopa o `pom.xml`:

- Xerar o ficheiro war coa aplicación:

```
$ mvn install
```

- Copiar o ficheiro war xerado no paso anterior ao directorio `webapps` de Tomcat (`/var/lib/tomcat9/webapps/`).

Manual de usuario

Neste anexo descríbense as funcionalidades da aplicación co obxectivo de facilitar a súa utilización.

Na sección [Preparación do sistema](#) explícanse os pasos a realizar polo administrador para deixar o sistema listo para ser utilizado. En [Configuración dos datos xerais do proxecto](#) explícanse os pasos para configurar os detalles necesarios da interface de usuario da aplicación (logos, título, etc.). En [Configuración dos tipos de atributos](#) explícanse os pasos para que o administrador cree os tipos de atributos que sexan necesarios para cada tipo de elemento. En [Configuración dos tipos de elementos](#) explícanse os pasos para que o administrador cree os tipos de elementos xeométricos cos que se vaia traballar. Nas seguintes seccións explícanse as funcionalidades relacionadas coas operacións sobre elementos xeométricos, orientadas sobre todo a usuarios rexistrados. Unha vez que os usuarios van introducindo elementos no sistema, na sección [Revisión e publicación de elementos](#), explícase como debe facer o administrador para revisalos e publicalos. Inclúense tamén seccións sobre o resto de funcionalidades da aplicación: uso do visor xeográfico, xestión de usuarios, etc.

B.1 Preparación do sistema

Nesta sección explícanse as accións que, unha vez despregado o sistema en produción, deben realizar os administradores antes de abrir a aplicación ao público.

B.1.1 Configuración dos datos xerais do proxecto

Unha vez iniciada a sesión como administrador, débese dirixir ao menú de **Administración** e seleccionar **Parámetros de configuración**. A continuación detállanse os parámetros máis relevantes:

Parámetros de configuración

Id	Descrición	Código	Tipo de dato	Valor	Interface de usuario?	Accións
1	información do pé de páxina	FOOTER_INFO	Campo de texto con formato	©2020	Si	Modificar
2	título da aplicación	APP_TITLE	Campo de texto plano	PPGIS	Si	Modificar
3	favicon	APP_FAVICON	Imaxe	-	Si	Modificar
4	imaxe da barra de navegación	NAVBAR_LOGO	Imaxe	-	Si	Modificar
5	Imaxe do pé de páxina 1	FOOTER_IMAGE_1	Imaxe	-	Si	Modificar
6	Texto do pé de páxina 1	FOOTER_TEXT_1	Campo de texto con formato	-	Si	Modificar
7	Imaxe do pé de páxina 2	FOOTER_IMAGE_2	Imaxe	-	Si	Modificar
8	Texto do pé de páxina 1	FOOTER_TEXT_2	Campo de texto con formato	-	Si	Modificar
9	Imaxe do pé de páxina 3	FOOTER_IMAGE_3	Imaxe	-	Si	Modificar
10	Texto do pé de páxina 3	FOOTER_TEXT_3	Campo de texto con formato	-	Si	Modificar

Amosando do 1 ao 10 de 23 10 - 1 2 3 >

Figura B.1. Táboa cos parámetros de configuración.

- *APP_TITLE*: título da aplicación, sairá un campo de texto plano para modificar o seu valor.



Figura B.2. Ventá modal para modificar o valor do título da aplicación.

- *APP_FAVICON*: Favicon da aplicación, sairá un campo para engadir unha imaxe.

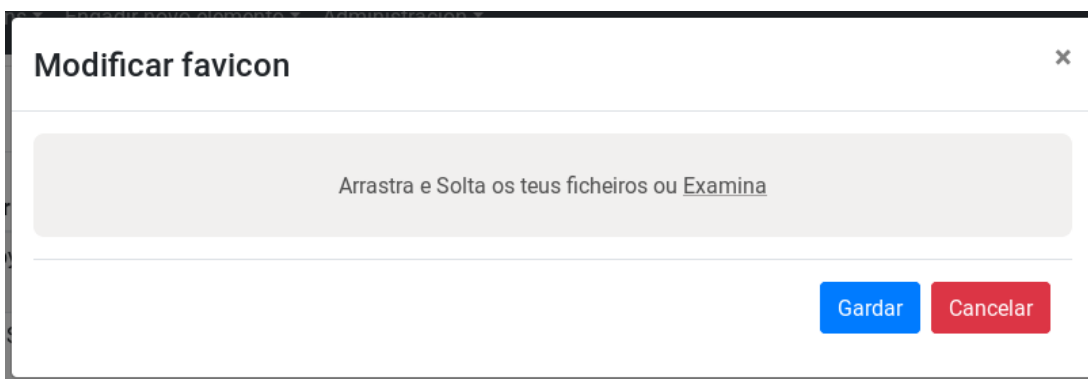


Figura B.3. Ventá modal para modificar o valor do favicon da aplicación.

- *FOOTER_INFO*: información do pé de páxina da aplicación, sairá un editor de texto con formato para modificar o seu valor.

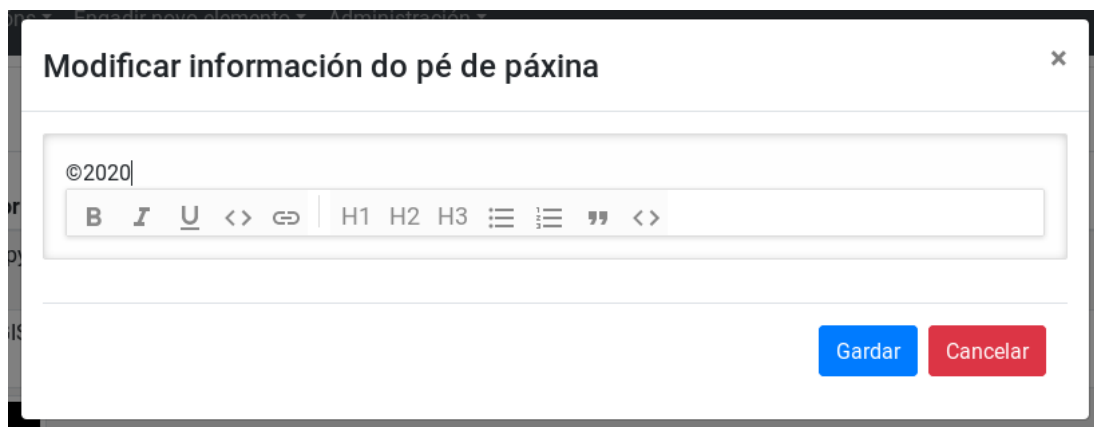


Figura B.4. Ventá modal para modificar o valor do pé de páxina da aplicación.

- *NAVBAR_LOGO*: imaxe que irá na barra de navegación, sairá un campo para engadir unha imaxe.

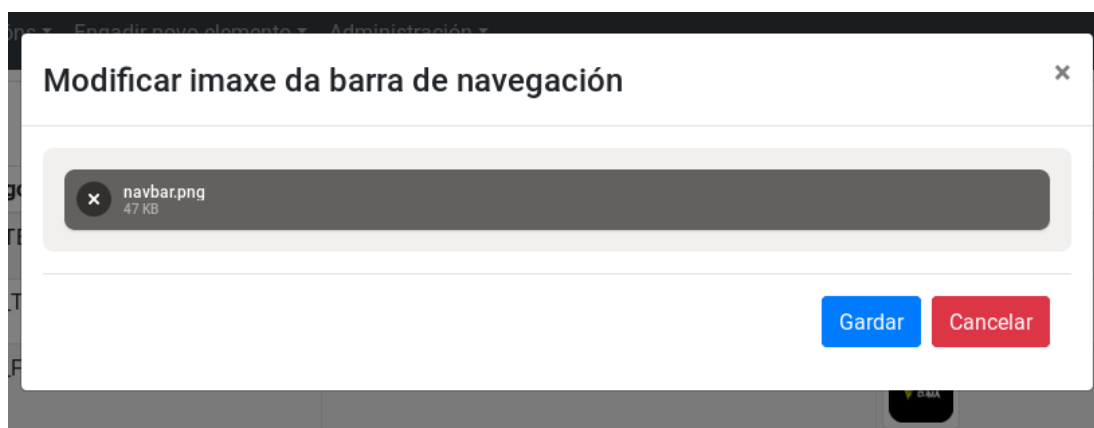


Figura B.5. Ventá modal para modificar a imaxe do menú.

- *FOOTER_IMAGE_[1-8]*: existen oito ocos distintos para engadir imaxes no pé de páxina.

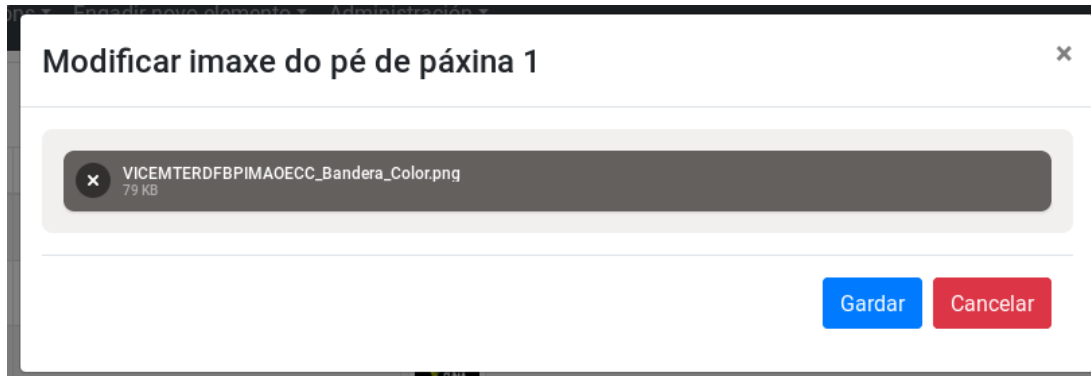


Figura B.6. Ventá modal para modificar a imaxe do primeiro oco do pé de páxina.

- *FOOTER_TEXT_[1-8]*: existen oito ocos distintos para meter textos no pé de páxina.

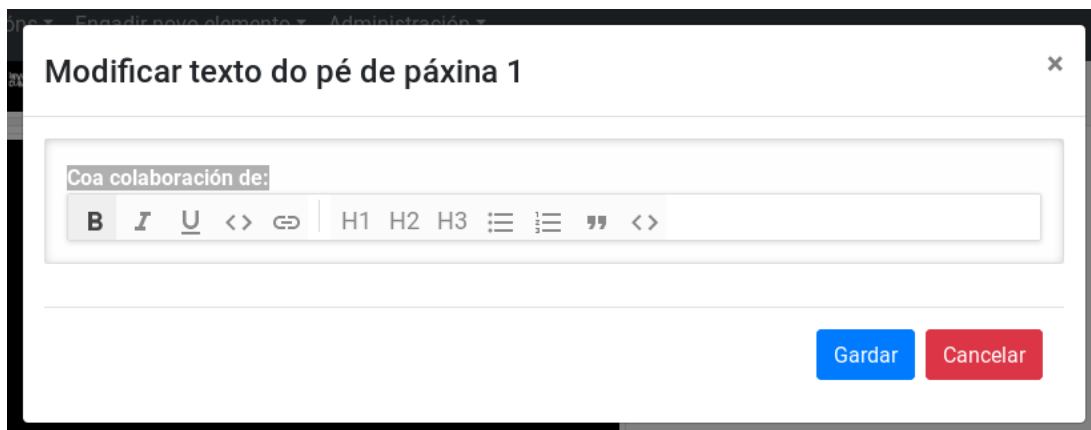


Figura B.7. Ventá modal para modificar o texto do primeiro oco do pé de páxina.

- *NAVBAR_IMAGE_LINK*: URL que se asignará á imaxe da barra de navegación. Se non ten valor o link ao que apuntará esta imaxe será ao visor xeográfico. Á hora de modificalo sairá un campo de texto plano.

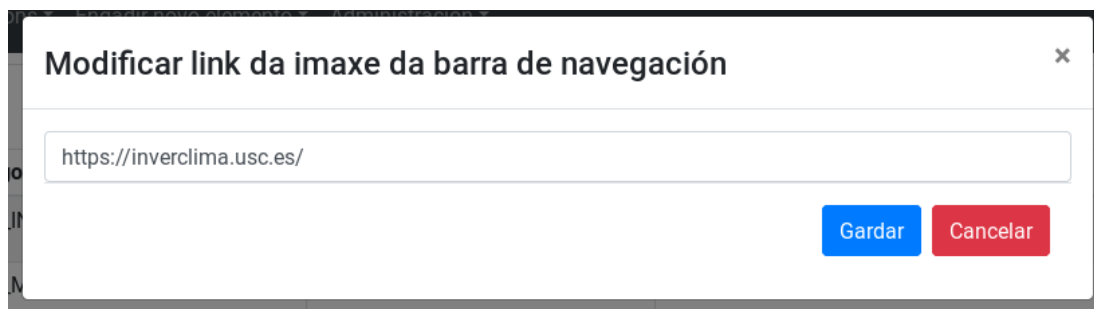


Figura B.8. Ventá modal para modificar o valor do link ao que apuntará a imaxe do menú.

- *MAP_INITIAL_EXTENT*: extensión inicial do mapa no visor xeográfico. Á hora de modificalo sairá un campo de texto plano. O valor debe consistir en dúas coordenadas separadas por coma, no sistema de referencia *EPSG:3587*.

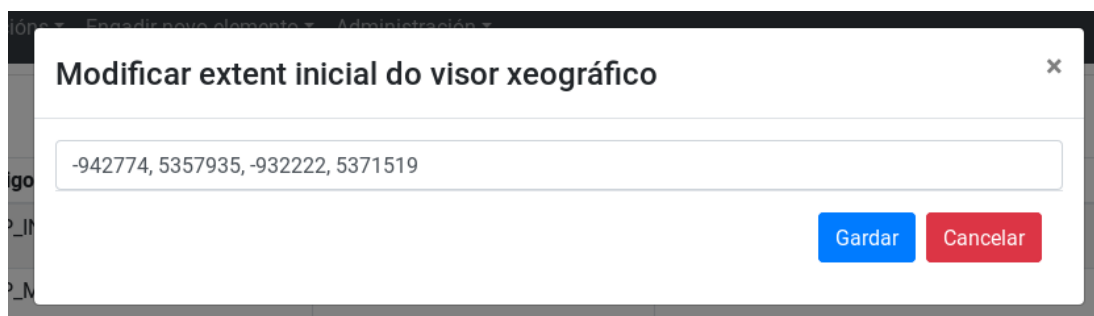


Figura B.9. Ventá modal para modificar a extensión inicial que terá o mapa do visor xeográfico.

- *MAP_MAX_EXTENT*: extensión máxima do mapa no visor xeográfico. Á hora de modificalo sairá un campo de texto plano. O valor debe consistir en dúas coordenadas separadas por coma, no sistema de referencia *EPSG:3587*.

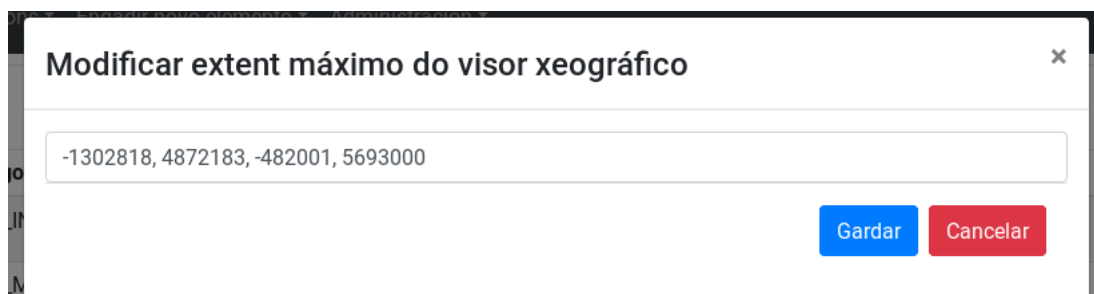


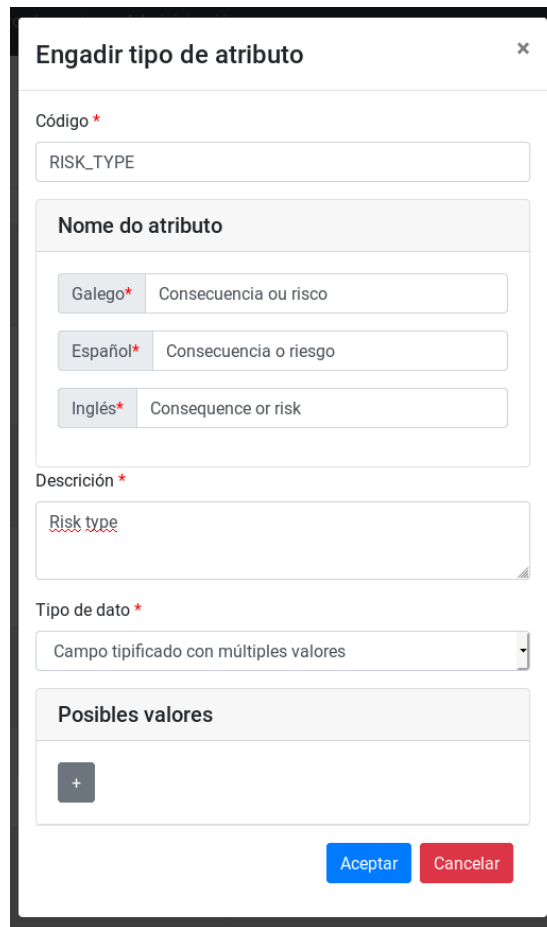
Figura B.10. Ventá modal para modificar o valor da extensión máxima que terá o mapa do visor xeográfico.

Cando se modifica o valor dun parámetro, debe recargarse a páxina para que o cambio teña efecto.

B.1.2 Configuración dos tipos de atributos

Unha vez iniciada sesión como administrador, débese dirixir ao menú de **Administración** e seleccionar **Tipos de atributos**.

A continuación deberase facer clic no botón **Engadir tipo de atributo**.



The screenshot shows a modal window titled "Engadir tipo de atributo" with a close button (X) in the top right corner. The form contains the following fields and sections:

- Código ***: A text input field containing "RISK_TYPE".
- Nome do atributo**: A section with three rows, each containing a language label and a text input field:
 - Galego ***: "Consecuencia ou risco"
 - Español ***: "Consecuencia o riesgo"
 - Inglés ***: "Consequence or risk"
- Descrición ***: A text area containing "Risk type".
- Tipo de dato ***: A dropdown menu set to "Campo tipificado con múltiples valores".
- Posibles valores**: A section with a single "+" button.
- At the bottom right, there are two buttons: "Aceptar" (blue) and "Cancelar" (red).

Figura B.11. Ventá modal para engadir un tipo de atributo.

Sairá unha ventá modal onde se cubrirán os seguintes campos:

- **Código:** un valor único que identificará ao tipo de atributo.
- **Nome do atributo:** hai que cubrir tres valores que corresponden coas distintas traducións do nome do tipo de atributo (galego, español e inglés).

- **Descripción:** unha pequena descrición do tipo de atributo, só visible para os administradores nas funcionalidades de administración
- **Tipo de dato:** Dependendo do tipo de dato escollido sairán novos campos que se detallan a continuación:
 - **Campo tipificado con múltiples valores ou campo tipificado:** sairá un campo de **Posibles valores** no que haberá un botón "+" para engadir cada un dos posibles valores. Para cada atributo cubriranse catro campos: código, galego, español e inglés.

Tipo de dato *

Campo tipificado con múltiples valores

Posibles valores

Código *

Galego*

Español*

Inglés*

-

+

Figura B.12. Campos a cubrir na definición dun tipo de atributo tipificado.

- **Campo de texto plano:** sairá un campo *Tamaño máximo* que se corresponde co número máximo de caracteres que permitirá o atributo.

Tipo de dato *

Campo de texto plano

Tamaño máximo *

Figura B.13. Campos a cubrir na definición dun tipo de atributo de texto plano.

B.1.3 Configuración dos tipos de elementos

Unha vez engadidos os atributos necesarios, para a creación dos tipos de elementos, no menú de **Administración** seleccionárase **Tipos de elementos** e a continuación premer no botón **Engadir novo tipo de elemento**. Sairá unha ventá modal cos seguintes campos:

Engadir novo tipo de elemento

Código *

RISK_ZONE

Nome do novo tipo

Galego* Zona de risco

Español* Zona de riesgo

Inglés* Risk zone

Tipo de xeometría *

Punto, polígono ou liña

Atributos

+

Aceptar Cancelar

Figura B.14. Ventá modal de creación dun tipos de elemento xeométrico.

- **Código:** un valor único que identificará o tipo de elemento.
- **Nome do novo tipo:** hai que cubrir tres valores que corresponden coas distintas traducións do nome do tipo de elemento (galego, español e inglés)

- **Tipo de xeometría:** elixirase o tipo de xeometría aceptada para o tipo de elemento.
- **Atributos:** haberá un botón "+" que engadirá tres campos:

The image shows a form titled "Atributos" with the following fields and controls:

- Tipo de atributo:** A dropdown menu with the selected value "Consecuencia ou risco".
- Orde:** A text input field containing the number "1" and a small up/down arrow icon on the right.
- Obrigatorio:** A checkbox that is checked, followed by the text "Obrigatorio".
- Below the checkbox are two square buttons: a minus sign "-" and a plus sign "+".

Figura B.15. Campos a cubrir para os atributos dun novo tipo de elemento.

- **Tipo de atributo:** seleccionarse entre unha lista o tipo de atributo que se quere engadir, de entre os dispoñibles.
- **Orde:** será un número que determinará a orde en que se colocarán o atributo na visualización dun elemento deste tipo.
- **Obrigatorio/Non obrigatorio:** determinará se o atributo é obrigatorio ou non.

B.2 Consultar elementos xeométricos

Calquera usuario, estea autenticado ou non, poderá consultar por cada tipo os elementos engadidos no sistema. No caso dos usuarios non autenticados só poderán visualizar aqueles que sexan públicos. Se o usuario está autenticado e non é un administrador terá a posibilidade de visualizar aqueles públicos e aqueles que cuxo autor sexa o propio usuario. Os administradores poderán consultar todos os elementos dun tipo e aqueles que cuxo autor sexa o propio usuario. Existen dúas vías para consultar a información dun elemento xeométrico:

- **Dende o visor xeográfico:**



Figura B.16. Control para a consulta de elementos xunto coa información dun elemento creado logo de facer clic sobre el.

Existe unha capa no mapa por cada tipo de elemento. Os usuarios autenticados terán a maiores unha capa *As miñas contribucións* onde se encontran os elementos cuxo autor sexa o propio usuario. Para poder consultar a información un elementos haberá que activar o control pertinente e facer clic nunha zona entorno ao elemento. Unha vez feito sairá un popup que contén como título o tipo de elemento, o id do elemento e a súa descrición. No contido sairán os atributos do elemento e se o usuario é un administrador as observacións internas. Ademais todos os usuarios poderán centrar a vista do mapa entorno ao elemento facendo clic no botón *Centrar*. Os usuarios autenticados terán un botón *Modificar* se teñen os permisos para poder facelo.

Se existe máis de un elemento nesa zona no popup aparecerán todos os elementos que o usuario poida ver. Cando o usuario faga clic nun dos elementos despregarase a información asociada a ese elemento.

- **Dende a barra de navegación:**

- **Dende o despregable *Táboas de elementos*:**

Para acceder á táboa dun tipo de elemento particular débese facer clic no despregable *Táboas de elementos* e seleccionar o tipo desexado. Os usuarios non autenticados e os que non sexan administradores só poderán ver aqueles elementos que

sexan públicos. Os administradores poderán ver todos os elementos asociados a ese tipo de elemento.

Unha vez elixido sairá unha táboa coas seguintes columnas:

Id	Descrición	Atributos	Introdución	Última modificación	É público?	Está revisado?	Xeometría	Obs. internas	Accións	Ficheiros asociados
1	Zona de risco de proba	<p>Consecuencia ou risco Maior incidencia de pragas ou enfermidades</p> <p>Valoración Risco alto</p>	7/9/2020 11:53	7/9/2020 11:53	Non	Non	Poligono	-	Ver fotos Ver no mapa Modificar Eliminar	+

Figura B.17. Táboa coa información dos elementos asociados a un tipo de elemento.

- * **Id:** o id único do elemento
- * **Atributos:** unha lista dos atributos asociados ao elemento xunto cos seus valores.
- * **Introdución:** a data de introdución do elemento.
- * **Última modificación:** a data da última vez que se modificou o elemento.
- * **Accións:** un listado das accións ás que ten acceso o usuario.
 - *Ver no mapa:* centrará o visor xeográfico entorno a ese elemento.
 - *Ver fotos:* só aparecerá se o elemento ten imaxes asociados poderán verse como unha galería
 - *Modificar:* só aparecerá no caso en que o usuario teña permiso para facelo, permitirá modificar o elemento no mapa.
 - *Eliminar:* só aparecerá no caso en que o usuario teña permiso para facelo, permitirá eliminar o elemento.
- * **Ficheiros asociados:** consiste nun botón que, unha vez se fai clic nel, desprega unha táboa que mostra a información de todos os ficheiros asociados ao elemento. A táboa ten os seguintes campos:

Ficheiros asociados					
UUID	Nome	Formato	Tipo mime	Tamaño	Accións
ff378520-f0ef-11ea-ac1b-3ba13d08eb60	unnamed.png	PNG	image/png	19,91 KB	Descargar

Figura B.18. Táboa coa información dos ficheiros asociados a un elemento.

- **UUID:** só visible para administradores, indica o UUID do ficheiro.
- **Nome:** indica o nome do ficheiro.

- **Formato:** indica o tipo de ficheiro que é.
- **Tamaño:** indica o tamaño que ocupa o ficheiro.
- **Accións:** botón descargar que permite gardar o ficheiro no dispositivo do usuario.
- **Tipo mime:** só visible para os administradores, indica o tipo do ficheiro.
- * **É público?:** só visible para usuarios autenticados, indica se o elemento está publicado.
- * **Está revisado?:** só visible para usuarios autenticados, indica se o elemento está revisado
- * **Xeometría:** só visible para administradores, indica o tipo de xeometría do elemento.
- * **Obs. internas:** só visible para administradores, mostra as observacións internas inseridas polos administradores.
- * **Autor:** só visible para administradores, mostra o usuario que creou o elemento.

A maiores os usuarios rexistrados teñen un link que lles dirixirá a *As miñas contribucións* para ese tipo de elemento.

– **Dende o despregable *As miñas contribucións*:**

Para acceder á táboa das contribucións do usuario para un tipo de elemento, débese facer clic no despregable *As miñas contribucións* e seleccionar o tipo de elemento desexado. Mostrarase unha táboa análoga a que se pode atopar en *Táboas de elementos* excepto que para os administradores eliminouse a columna *Autor*.

B.3 Creación de elementos xeométricos

Un usuario autenticado (administrador ou usuario particular) poderá crear elementos xeométricos dun tipo. Existen catro vías para facelo que se describen a continuación.

- **Dende o visor xeográfico:** Existe un control no lateral do visor xeográfico que, unha vez activado, o usuario poderá elixir o tipo de elemento xeométrico que quere engadir así como a xeometría que terá o elemento creado. Poden existir tres tipos de xeometría:



Figura B.19. Control engadir activado e co tipo de elemento seleccionado no mapa.

- *Punto*: para debuxar un punto no mapa simplemente debe facerse clic na zona do mapa onde se queira crear o elemento.
- *Polígono*: para debuxar un polígono débese facer clic na zona do mapa onde queira comezar a debuxar, cada vez que se faga clic nunha zona o polígono irase creando. Unha vez se queira rematar de debuxar pódese facer dobre clic ou ben facer clic no primeiro punto que se debuxou.
- *Liña*: para debuxar unha liña débese facer clic na zona do mapa onde se queira comezar a debuxar, cada vez que se faga clic nunha zona a liña irase creando. Para rematar de debuxar débese facer dobre clic.

Unha vez se remate de debuxar a xeometría, aparecerá un panel lateral onde o título será *Crear + nome do tipo de elemento*. Este panel conterá:

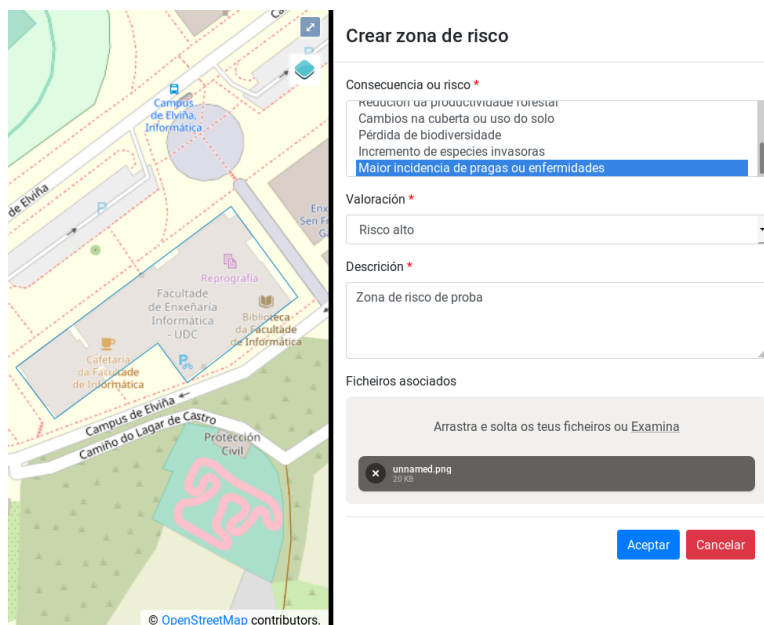


Figura B.20. Panel lateral co formulario e xeometría.

- *Atributos do tipo de elemento xeométrico*: aparecerán os campos para cada un dos tipos de atributos asociados a ese tipo de elemento.
- *Descrición*: campo de texto onde se poderá describir o elemento a crear.
- *Ficheiros asociados*: poderanse anexar ficheiros ao elemento a crear, pódese arrastrar e soltar os ficheiros que se desexen así como examinar no equipo do usuario. Unha vez se engadan os ficheiros, existe a posibilidade de eliminar cada un deles por separado.
- *Obs. internas*: campo que só aparecerá no caso de que o usuario autenticado sexa un administrador. É un campo de texto que só será visible sempre por un usuario de tipo administrador.
- *É público?*: só os administradores poderán ver este campo, indica se o elemento a crear será público.
- *Está revisado?*: só os administradores poderán ver este campo, indica se o elemento está xa revisado.

- **Dende a barra de navegación:**

- **Dende o despregable *Engadir novo elemento*:**

Unha vez se fai clic, aparecerá unha lista de todos os tipos de elementos xunto coas xeometrías válidas para ese tipo. Unha vez seleccione o tipo e a xeometría desexada activarase o control para engadir no mapa co tipo de elemento e xeometría seleccionada anteriormente. Os pasos para engadir a partir de agora son análogos ao método de engadir dende o visor xeográfico

- **Dende o despregable *Táboas de elementos*:** Unha vez se fai clic, aparecerá unha lista cos tipos de elementos existentes no sistema. Débese seleccionar o tipo deseado e, unha vez feito mostrarase a lista de elementos públicos, se o usuario non é administrador, ou a lista de todos os elementos dese tipo, en caso contrario. Para engadir un novo elemento seleccionárase o despregable *Crear + nome do tipo de elemento*. Sairá unha lista das xeometrías válidas para o tipo de elemento. Unha vez seleccionada a xeometría desexada activarase no visor o control para engadir co tipo de elemento e xeometría seleccionada anteriormente. Os pasos a partir de agora son análogos aos métodos anteriores.

- **Dende o despregable *As miñas contribucións*:** Os pasos son análogos a engadir dende *Táboas de elementos*.

B.4 Modificación de elementos

Un usuario autenticado poderá modificar os elementos xeométricos dun tipo. No caso de que o usuario non sexa administrador só poderá modificar aqueles que foron creados por el e que non estean revisados. Existen tres vías para modificar un elemento creado.



Figura B.21. Panel lateral cos valores do elemento a modificar xunto coa xeometría en modo edición.

- **Dende o visor xeográfico:**

Para poder modificar un elemento no visor xeográfico deberá activarse o control para consultar información sobre os elementos mediante clic. Unha vez apareza a información do elemento deberase facer clic no botón *Modificar*.

Unha vez feito no visor aparecerá un panel lateral onde aparecerán todos os valores engadidos polo autor. Permitirase modificar estes valores e ademais a xeometría engadida. Se se coloca o cursor sobre a xeometría poderá modificarse a xeometría engadida pulsando e arrastrando. Ademais o administrador terá tres campos a maiores: **É público?**, **Está revisado?** e **Obs. internas** que xa explicaron a súa función na sección [Creación de elementos xeométricos](#). Unha vez feitas as modificacións pertinentes aceptará os cambios.

- **Dende a barra de navegación:**

Os pasos son análogos á consulta de elementos xeométricos. Unha vez se encontre na táboa cos elementos o usuario buscará o elemento a modificar e pulsará o botón *Modificar* e a aplicación dirixirase ao visor xeográfico e mostrará a xeometría en modo modificación e o panel lateral con todos os valores. Os pasos son análogos á forma de modificar anterior.

B.5 Revisión e publicación de elementos

Cando un usuario introduce un elemento, o elemento gárdase como non público e non revisado. Mentres o elemento siga como non revisado, a persoa que introduciu o elemento pode modificalo. Os encargados de marcar un elemento como revisado son os administradores. Os administradores son tamén os que poden marcar un elemento como público. Os elementos públicos son visibles para todos os usuarios mentres que os elementos non públicos só son visibles para as persoas que os crearon e para os administradores.

As opcións para marcar un elemento como revisado e como público atópanse dentro do formulario de [Modificación de elementos](#) (ver figura B.22).

Obs. internas

Observacións que só poderán ver os administradores

É público?

Está revisado?

Figura B.22. Campos relativos á revisión e modificación de elementos, xunto co de observacións internas. Só visibles para os administradores.

B.6 Consultar os usuarios do sistema

Os administradores poderán ver todos os usuarios rexistrados no sistema. Para iso débese dirixir ao menú *Administración* e seleccionar a opción *Usuarios*.

Aparecerá un botón para engadir un novo usuario e unha táboa paxinada cos usuarios existentes no sistema cos seguintes campos:

Usuarios

[Crear usuario](#)

Id	Nome de usuario	Rol	Está bloqueado?	Accións
1	admin	Administrador do sitio	Non	Modificar
2	test	Usuario do sitio	Non	Modificar

Amosando do 1 ao 2 de 2 10 1

Figura B.23. Táboa cos usuarios existentes no sistema.

- **Id:** o identificador único para ese usuario
- **Nome de usuario:** o nome de usuario co que se rexistrou
- **Rol:** o rol ao que pertence ese usuario.
- **Está bloqueado?:** indica se o usuario está bloqueado
- **Accións:** só está o botón *Modificar* co que se poderá cambiar o estado do usuario.

B.7 Consultar os parámetros de configuración

Os administradores poderán consultar todos os parámetros de configuración do sistema. Para iso débese dirixir ao menú *Administración* e seleccionar a opción *Parámetros de configuración*.

Aparecerá unha táboa paxinada cos parámetros de configuración existentes no sistema cos seguintes campos:

Parámetros de configuración


Id	Descrición	Código	Tipo de dato	Valor	Interface de usuario?	Accións
1	información do pé de páxina	FOOTER_INFO	Campo de texto con formato	©2020	Si	Modificar
2	título da aplicación	APP_TITLE	Campo de texto plano	INVERCLIMA	Si	Modificar
3	favicon	APP_FAVICON	Imaxe		Si	Modificar

Figura B.24. Táboa cos parámetros de configuración existentes no sistema.

- **Id:** o identificador único para ese parámetro
- **Descrición:** unha breve descrición do parámetro.
- **Código:** código único que identifica ao parámetro.

- **Tipo de dato:** o tipo de dato do parámetro.
- **Valor:** o valor do parámetro, se é de tipo imaxe aparecerá unha previsualización da mesma.
- **Interface de usuario?:** indica se é un parámetro da interface de usuario.
- **Accións:** só está o botón *Modificar* co que se poderá cambiar o valor do parámetro.

B.8 Consultar tipos de atributos

Os administradores poderán consultar todos os tipos de atributos existentes no sistema. Para iso débese dirixir ao menú *Administración* e seleccionar a opción *Tipos de atributos*.

Aparecerá un botón para engadir un novo tipo de atributo e ademais unha táboa paxinada cos tipos de atributos existentes no sistema cos seguintes campos:

Tipos de atributos

[Engadir tipo de atributo](#)

Id	Código	Descrición	Tipo de dato	Accións	Posibles valores
1	FOOTER_INFO	información do pé de páxina	Campo de texto con formato	Modificar Eliminar	
2	APP_TITLE	título da aplicación	Campo de texto plano	Modificar Eliminar	
3	APP_FAVICON	favicon	Imaxe	Modificar Eliminar	

Figura B.25. Táboa cos tipos de atributos existentes no sistema.

- **Id:** o identificador único para ese tipo de atributo.
- **Código:** o código único que identifica ao tipo de atributo.
- **Descrición:** o nome do tipo de atributo internacionalizado no idioma do usuario.
- **Tipo de dato:** o tipo de dato do tipo de atributo.
- **Accións:** listado cos botóns *Modificar*, co que se poderá modificar o tipo de atributo, e *Eliminar* co que se eliminará o atributo se este non pertence a ningún tipo de elemento ou non é dun parámetro de configuración.
- **Posibles valores:** aparecerá un botón "+" para aqueles tipos de atributos que teñan valores tipificados que despregará unha táboa cos seguintes campos:

25	RISK_ASSESSMENT	Valoración	Campo tipificado	Modificar Eliminar	-
----	-----------------	------------	------------------	---	---

Posibles valores

Id	Código	Descripción
62	HIGH_RISK	Risco alto
63	MEDIUM_RISK	Risco medio
64	LOW_RISK	Risco baixo

Figura B.26. Táboa cos valores dun tipo de atributo do sistema.

- *Id*: o identificador único para ese valor tipificado.
- *Código*: o código único que identifica ao valor.
- *Descripción*: o código internacionalizado no idioma do usuario.

B.9 Consultar tipos de elementos

Os administradores poderán consultar todos os tipos de elementos existentes no sistema. Para iso débese dirixir ao menú *Administración* e seleccionar a opción *Tipos de elementos*.

Aparecerá un botón para engadir un novo tipo de elemento e ademais unha táboa paxinada cos tipos de elementos existentes no sistema cos seguintes campos:

Tipos de elementos

[Engadir novo tipo de elemento](#)

Id	Código	Nome	Tipo de xeometría	Accións	Atributos
1	RISK_ZONE	Zona de risco	Punto, polígono ou liña	Ver elementos Modificar Eliminar	+
2	ECOSYSTEM_SERVICE	Servizo ecosistémico	Punto, polígono ou liña	Ver elementos Modificar Eliminar	+
3	GREEN_INFRASTRUCTURE_ELEMENT	Elemento de infraestrutura verde	Punto, polígono ou liña	Ver elementos Modificar Eliminar	+

Amosando do 1 ao 3 de 3 10 1

Figura B.27. Táboa cos tipos de atributos existentes no sistema.

- *Id*: o identificador único para ese tipo de elemento.
- *Código*: o código único que identifica ao tipo de atributo.

- **Nome:** o nome do tipo de atributo internacionalizado no idioma do usuario.
- **Tipo de xeometría:** o tipo de xeometría aceptado polo tipo de elemento.
- **Accións:** listado cos botóns *Ver elementos*, co que se mostrará a táboa con todos os elementos asociados ao tipo de elemento, *Modificar*, co que se poderá modificar o tipo de elemento, e *Eliminar* co que se eliminará o tipo de elemento se non ten instancias asociadas.
- **Atributos:** aparecerá un botón "+" co que se poderá consultar os tipos de atributos do tipo de elemento. Os campos que aparecen son os seguintes:

Id	Código	Nome	Tipo de xeometría	Accións	Atributos
1	RISK_ZONE	Zona de risco	Punto, polígono ou liña	Ver elementos Modificar Eliminar	-

Atributos			
Id	Nome	Tipo de dato	Orde
24	Consecuencia ou risco	Campo tipificado con múltiples valores	1
25	Valoración	Campo tipificado	2

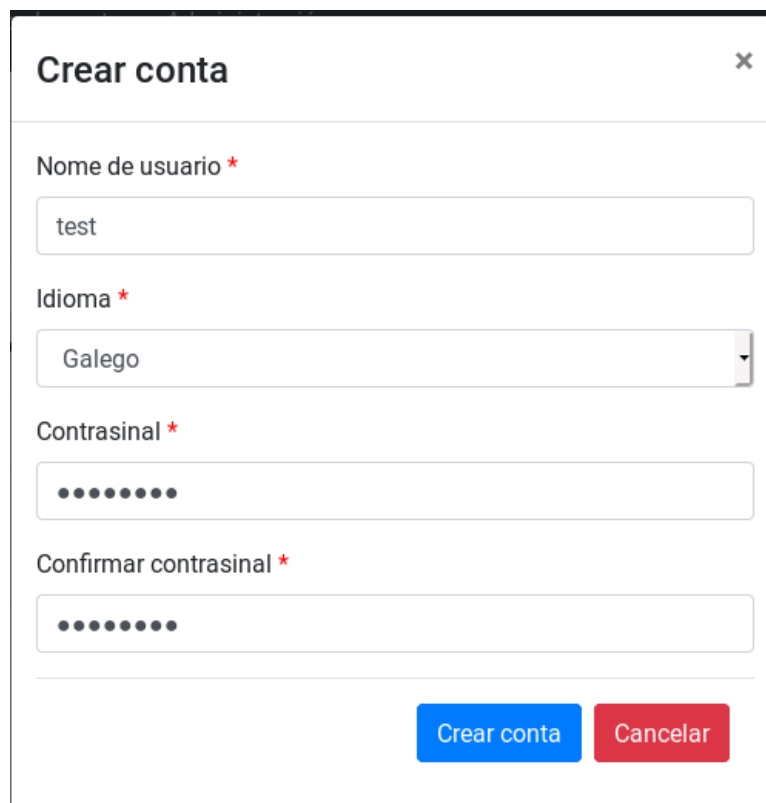
Figura B.28. Táboa cos tipos de atributos dun tipo de elemento do sistema.

- *Id:* o identificador único para ese atributo.
- *Nome:* o nome do tipo de atributo internacionalizado no idioma do usuario.
- *Tipo de dato:* o tipo de dato do atributo.
- *Orde:* a orde coa que se mostrará o atributo.

B.10 Creación de usuarios

Para a creación de novos usuarios, o administrador dirixirase ao menú *Administración* e seleccionará a opción *Usuarios*. Mostrarase unha táboa paxinada con todos os usuarios existentes no sistema. Para a creación do novo usuario deberase pulsar no botón *Crear usuario* e elixir o rol do mesmo.

Mostrarase unha ventá modal cun formulario onde se cubrirán os datos do usuario: nome de usuario, idioma (galego, español ou inglés) e deberase engadir dúas veces o contrasinal.



Crear conta ×

Nome de usuario *

Idioma *

Contrasinal *

Confirmar contrasinal *

Crear conta **Cancelar**

Figura B.29. Ventá modal para a creación dun novo usuario.

B.11 Modificación do estado dun usuario

Para modificar o estado dun usuario o administrador deberase dirixir á táboa dos usuarios e pulsar o botón *Modificar* do usuario desexado.

Mostrarase unha ventá modal con dous campos:

- Bloquear/Desbloquear: no caso de que o usuario elixido se encontre bloqueado na etiqueta mostrarase *Desbloquear*, en caso contrario, *Bloquear*. Pulsando, ben na etiqueta ou ben na caixa de verificación, cambiarase o estado do usuario unha vez se confirmen os cambios.
- Cambiar contrasinal: no caso de que o administrador queira cambiar o contrasinal do usuario elixido, deberá activar a caixa de verificación *Cambiar contrasinal*. Amosaranse dous campos onde se debe cubrir o novo contrasinal para o usuario.

The image shows a modal window titled "Usuario test" with a close button (X) in the top right corner. The form contains two checked checkboxes: "Desbloquear" and "Cambiar contrasinal". Below these are two password input fields, each with a red asterisk indicating a required field. The first field is labeled "Novo contrasinal" and the second is labeled "Confirme o novo contrasinal". At the bottom right, there are two buttons: "Gardar" (blue) and "Cancelar" (red).

Figura B.30. Ventá para a modificación do estado e/ou o contrasinal dun usuario.

B.12 Modificar tipos de elementos

Para modificar un tipo de elemento existente, o administrador dirixirase ao menú *Administración* e seleccionará a opción *Tipos de elementos*. Mostrarase a táboa paxinada con todos os tipos de elementos existentes. O administrador debe pulsar *Modificar* no tipo de elemento escollido.

Para aqueles tipos de elementos que teñan instancias asociadas, é dicir, algún elemento existente no sistema pertence a este tipo, soamente se poderán modificar as traducións do nome do tipo de elemento (campos *Galego*, *Español* e *Inglés*).

Modificar elemento xeométrico ✕

Código *

RISK_ZONE

Nome do tipo de elemento

Galego*	Zona de risco
Español*	Zona de riesgo
Inglés*	Risk zone

Tipo de xeometría *

Punto, polígono ou liña

Atributos

Tipo de atributo

Consecuencia ou risco

Orde

1 ⬇ ⬆

Obrigatorio

-

Tipo de atributo

Figura B.31. Ventá para a modificación dun tipo de elemento con instancias asociadas.

No caso en que o tipo de elemento non teña instancias, permitirase modificar, ademais das traducións do nome, o tipo de xeometría e os tipos de atributos asociados ao tipo de elemento.

Pulsando no botón "+" engadiranse novos atributos, por outra banda facendo clic no botón "-" eliminarase o tipo de atributo.

Modificar elemento xeométrico ×

Código *

ECOSYSTEM_SERVICE

Nome do tipo de elemento

Galego* Servizo ecosistémico

Español* Servicio ecosistémico

Inglés* Ecosystem service

Tipo de xeometría *

Punto, polígono ou liña

Atributos

Tipo de atributo

Tipo de servizo

Orde

1

Obrigatorio

-

Tipo de atributo

Figura B.32. Ventá para a modificación dun tipo de elemento sen instancias asociadas.

B.13 Modificar tipos de atributos

Para modificar un tipo de atributo existente, o administrador dirixirase ao menú *Administración* e seleccionará a opción *Tipos de atributos*. Mostrarase a táboa paxinada con todos os tipos de atributos existentes. O administrador deberá pulsar o botón *Modificar* no tipo de atributo escollido.

Mostrarase unha ventá modal onde se poderán modificar as traducións do nome do atributo e, ademais, a descrición. Se o tipo do elemento é tipificado, permitirase modificar os posibles valores, engadindo con "+" un novo ou, por outra banda, eliminar un existente con "-".

Modificar tipo de atributo ✕

Código *

RISK_TYPE

Nome do atributo

Galego* Consecuencia ou risco

Español* Consecuencia o riesgo

Inglés* Consequence or risk

Descrición *

Risk type

Tipo de dato *

Campo tipificado con múltiples valores

Posibles valores

Código *

FOREST_FIRES_RISK

Galego* Incendios forestais

Español* Incendios forestales

Figura B.33. Ventá para a modificación dun tipo de atributo.

Se o atributo está asociado a un parámetro de configuración só é posible modificar as

traducións do nome e tamén a descrición.

Modificar tipo de atributo ✕

Código *

FOOTER_INFO

Nome do atributo

Galego* información do pé de páxina

Español* información del pie de página

Inglés* footer information

Descrición *

Footer info

Aceptar Cancelar

Figura B.34. Ventá para a modificación dun tipo de atributo dun parámetro de configuración.

B.14 Eliminar un tipo de elemento

Os administradores poderán eliminar un tipo de elemento se este non ten instancias asociadas, é dicir, que non existen elementos creados para ese tipo.

Para poder eliminar deberase dirixir ao menú *Administración* e seleccionar a opción *Tipos de elementos*. Unha vez se mostre a táboa cos tipos de elementos existentes no sistema, o administrador elixirá o botón *Eliminar* de aquel tipo de elemento que queira quitar do sistema. Se o elemento ten instancias asociadas mostrarase un erro.

B.15 Eliminar un tipo de atributo

Os administradores poderán eliminar un tipo de atributo se este non está asociado a ningún tipo de elemento. Para poder eliminar deberase dirixir ao menú *Administración* e seleccionar a opción *Tipos de atributos*. Unha vez se mostre a táboa cos tipos de atributos existentes no sistema, o administrador elixirá o botón *Eliminar* de aquel tipo de atributo que queira quitar do sistema. Se o atributo está asociado a algún tipo de elemento mostrarase un erro.

B.16 Eliminar un elemento

Os usuarios do sistema teñen a posibilidade de eliminar os elementos engadidos. Os administradores poderán eliminar calquera elemento engadido, pola contra, os usuarios particulares só poderán eliminar aqueles que cuxo autor sexa o propio usuario e non estean revisados.

Para eliminar un elemento deberán dirixirse ao menú *As miñas contribucións* e seleccionar o tipo de elemento desexado. Unha vez se mostre a táboa de elementos para ese tipo, o usuario seleccionará o botón *Eliminar* e confirmará que quere eliminalo. Se o elemento xa está revisado, os usuarios particulares non terán o botón eliminar.

Os administradores poderán eliminar os elementos creados por eles e calquera outro. Para eliminar un elemento dun tipo dirixirase a *Táboas de elementos* e seleccionar o tipo de elemento xeométrico desexado. Unha vez se amose a táboa cos elementos para ese tipo, seleccionará o botón *Eliminar* do elemento desexado.

B.17 Manexo do visor xeográfico

Existen diferentes controis no visor xeográfico que non se mencionaron anteriormente. Detállanse a continuación:

- **Ver a máxima extensión:** encadrará o mapa á máxima extensión configurada.



Figura B.35. Control para ver a máxima extensión configurada.

- **Zoom por encadre:** unha vez activado, deberá pulsar e arrastrar na zona elixida para facer zoom ao encadre debuxado.

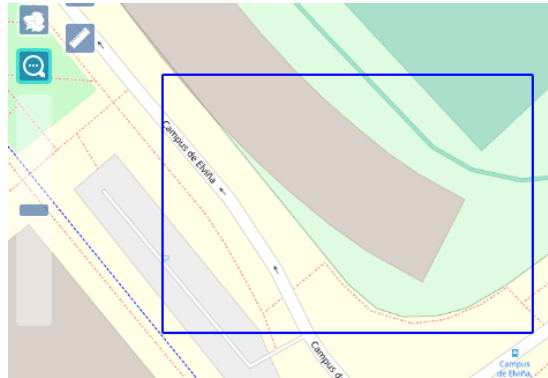


Figura B.36. Control do zoom por encadre activado xunto coa área á que se ampliará a vista do mapa.

- **Medir áreas:** unha vez activado, o usuario poderá debuxar un polígono que mostrará a información da área debuxada.

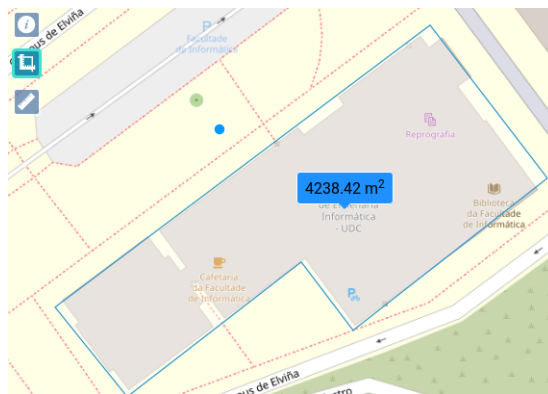


Figura B.37. Control de medir áreas activado xunto cunha medición feita.

- **Medir distancias:** unha vez activado, o usuario poderá debuxa unha liña que mostrará a información da lonxitude da liña debuxada.



Figura B.38. Control de medir distancias activado xunto cunha medición feita.

- **Control de capas:** na lenda, permitirase ao usuario activar e desactivar as capas configuradas no mapa. Unha vez que activa unha capa, o visor mostrará no mapa a información da mesma, en caso contrario, o visor quitará do mapa a información desa capa.

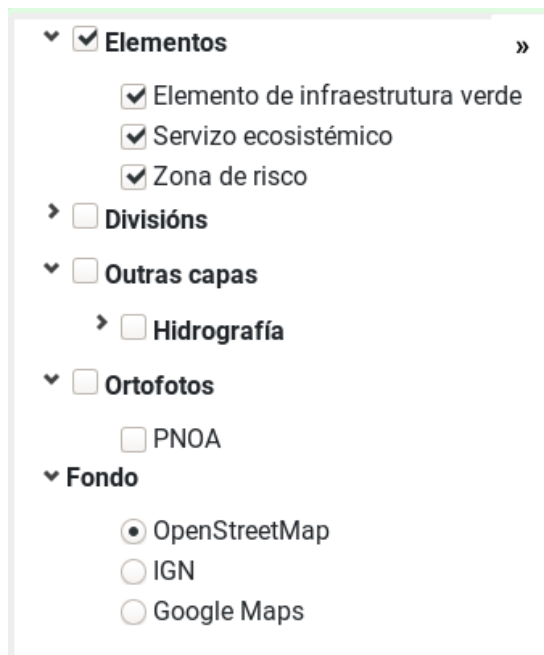


Figura B.39. Estado do control de capas para un usuario anónimo.

Relación de Acrónimos

- ACID** Atomicity, Consistency, Isolation and Durability. 63
- API** Application Programming Interface. 5
- CGI** Common Gateway Interface. 12
- CRUD** Create, Read, Update and Delete. 6
- DAO** Data Access Object. 15, 37
- DTO** Data Transfer Object. 15, 46
- JDBC** Java Database Conectivity. 15
- JPA** Java Persistence API. 41
- JSON** JavaScript Object Notation. 6, 15
- OGC** Open Geospatial Consortium. 6, 14
- ORM** Object Relational Mapping. 5, 41
- POM** Project Object Model. 9
- PPGIS** Public Participation Geographic Information System. 1
- REST** Representational State Transfer. 5, 37
- SFSQL** Simple Features for SQL. 12
- SIX** Sistema de Información Xeográfica. 5, 17
- SPA** Single-page Application. 7, 14, 52

WCS Web Coverage Service. 12

WFS Web Feature Service. 1, 12, 14

WMS Web Map Service. 1, 12, 14

Glosario

Favicon Do inglés favorites icon (icono de favoritos), é unha pequena imaxe asociada a un sitio web que normalmente é usada para que os navegadores á amosen xunto á URL da páxina na barra de navegación e/ou nos favoritos.. 86

Proxy Nun sistema informático, programa ou mecanismo que fai de intermediario entre as peticións que fai un cliente a algún determinado servizo.. 11

Shape Formato de almacenamento de información xeorreferenciada moi popular, baseado no uso de ficheiros.. 78

Bibliografía

- [1] Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc, “Public participation geographic information system,” 2020. [Online]. Available: https://en.wikipedia.org/wiki/Public_participation_geographic_information_system
- [2] “PostgreSQL,” 2020. [Online]. Available: <https://www.postgresql.org/>
- [3] “PostGIS,” 2020. [Online]. Available: <https://postgis.net/>
- [4] “React,” 2020. [Online]. Available: <https://es.reactjs.org/>
- [5] “OpenLayers,” 2020. [Online]. Available: <https://openlayers.org/>
- [6] “Spring,” 2020. [Online]. Available: <https://spring.io/>
- [7] “QGIS as OCG Data Server,” 2020. [Online]. Available: https://docs.qgis.org/3.10/en/docs/user_manual/working_with_ogc/server/index.html
- [8] “KoBoToolbox,” 2020. [En línea]. Disponible en: <https://www.kobotoolbox.org/>
- [9] “Open Data Kit,” 2020. [En línea]. Disponible en: <https://opendatakit.org/>
- [10] “LimeSurvey,” 2020. [En línea]. Disponible en: <https://www.limesurvey.org>
- [11] “Presentando JSX,” 2020. [En línea]. Disponible en: <https://es.reactjs.org/docs/introducing-jsx.html>
- [12] “JTS Topology Suite,” 2020. [En línea]. Disponible en: <https://github.com/locationtech/jts>
- [13] “Hibernate Spatial,” 2020. [En línea]. Disponible en: <http://www.hibernate.org/>
- [14] “Hibernate,” 2020. [En línea]. Disponible en: <https://hibernate.org/>
- [15] “Spatial4j,” 2020. [En línea]. Disponible en: <https://github.com/locationtech/spatial4j>
- [16] “Especificación de GeoJSON,” 2020. [En línea]. Disponible en: <https://geojson.org/>

- [17] “Spring Boot,” 2020. [En línea]. Disponible en: <https://spring.io/projects/spring-boot/>
- [18] “Querydsl,” 2020. [En línea]. Disponible en: <http://www.querydsl.com/>
- [19] “Java JWT: JSON Web Token for Java and Android,” 2020. [En línea]. Disponible en: <https://github.com/jwtkt/jjwt>
- [20] “The Parser interface,” 2020. [En línea]. Disponible en: <https://tika.apache.org/>
- [21] “OpenLayers LayerSwitcher,” 2020. [En línea]. Disponible en: <https://github.com/walkermatt/ol-layerswitcher>
- [22] Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc, “Single-page application,” 2020. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Single-page_application
- [23] “React Bootstrap,” 2020. [En línea]. Disponible en: <https://react-bootstrap.github.io/>
- [24] “react-bootstrap-table2,” 2020. [En línea]. Disponible en: <https://react-bootstrap-table.github.io/react-bootstrap-table2/>
- [25] “Redux,” 2020. [En línea]. Disponible en: <https://redux.js.org/>
- [26] “React Redux,” 2020. [En línea]. Disponible en: <https://react-redux.js.org/>
- [27] “React Router,” 2020. [En línea]. Disponible en: <https://github.com/ReactTraining/react-router>
- [28] “React Intl,” 2020. [En línea]. Disponible en: <https://github.com/stripe/react-intl>
- [29] “Draft.js,” 2020. [En línea]. Disponible en: <https://draftjs.org/>
- [30] “DraftJS: Import HTML to ContentState,” 2020. [En línea]. Disponible en: <https://github.com/sstur/draft-js-utils/tree/master/packages/draft-js-import-html>
- [31] “DraftJS: Export ContentState to HTML,” 2020. [En línea]. Disponible en: <https://github.com/sstur/draft-js-utils/tree/master/packages/draft-js-export-html>
- [32] “Formik,” 2020. [En línea]. Disponible en: <https://formik.org/>
- [33] “Yup,” 2020. [En línea]. Disponible en: <https://github.com/jquense/yup>
- [34] “jQuery,” 2020. [En línea]. Disponible en: <https://jquery.com/>
- [35] “Lodash,” 2020. [En línea]. Disponible en: <https://lodash.com/>

- [36] “Axios,” 2020. [En línea]. Disponible en: <https://github.com/axios/axios>
- [37] “Node.js,” 2020. [En línea]. Disponible en: <https://nodejs.org>
- [38] “React Responsive Carousel,” 2020. [En línea]. Disponible en: <https://react-responsive-carousel.js.org/>
- [39] “JUnit 5,” 2020. [En línea]. Disponible en: <https://junit.org/junit5/>
- [40] “EclEmma,” 2020. [En línea]. Disponible en: <https://www.eclEmma.org/>
- [41] “JaCoCo,” 2020. [En línea]. Disponible en: <https://www.jacoco.org/jacoco/>
- [42] “Cypress,” 2020. [En línea]. Disponible en: <https://www.cypress.io/>
- [43] “frontend maven plugin,” 2020. [En línea]. Disponible en: <https://github.com/eirslett/frontend-maven-plugin>
- [44] Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc, “npm,” 2020. [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))
- [45] “Redmine,” 2020. [En línea]. Disponible en: <https://www.redmine.org/>
- [46] “QGIS,” 2020. [En línea]. Disponible en: <https://www.qgis.org/>
- [47] “Eclipse IDE,” 2020. [En línea]. Disponible en: <https://www.eclipse.org/eclipseide/>
- [48] “Visual Studio Code,” 2020. [En línea]. Disponible en: <https://code.visualstudio.com/>
- [49] “draw io,” 2020. [En línea]. Disponible en: <https://drawio-app.com/>
- [50] “Overleaf,” 2020. [En línea]. Disponible en: <https://es.overleaf.com>
- [51] “Zuul,” 2020. [En línea]. Disponible en: <https://github.com/Netflix/zuul>
- [52] Open Geospatial Consortium, “Opengis® Web Map Service Interface Standard,” 2020. [En línea]. Disponible en: <https://www.ogc.org/standards/wms>
- [53] Open Geospatial Consortium, “Web Feature Service,” 2020. [En línea]. Disponible en: <https://www.ogc.org/standards/wfs>
- [54] “Core J2EE Patterns - Data Access Object,” 2020. [En línea]. Disponible en: <https://www.oracle.com/java/technologies/dataaccessobject.html>
- [55] Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc, “Transferencia de Estado Representacional,” 2020. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional

- [56] “Spring Data,” 2020. [En línea]. Disponible en: <https://spring.io/projects/spring-data>
- [57] “JSR 338: Java™ Persistence 2.2,” 2020. [En línea]. Disponible en: <https://jcp.org/en/jsr/detail?id=338>
- [58] Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc, “Facade (patrón de diseño),” 2020. [En línea]. Disponible en: [https://gl.wikipedia.org/wiki/Facade_\(patr%C3%B3n_de_dese%C3%B1o\)](https://gl.wikipedia.org/wiki/Facade_(patr%C3%B3n_de_dese%C3%B1o))
- [59] Techopedia, “End-to-end Test,” 2020. [En línea]. Disponible en: <https://www.techopedia.com/definition/7035/end-to-end-test>
- [60] “DOM,” 2020. [En línea]. Disponible en: https://es.wikipedia.org/wiki/Document_Object_Model
- [61] “Proyecto Inverclima,” 2020. [En línea]. Disponible en: <https://inverclima.usc.es/>
- [62] “TinyMCE,” 2020. [En línea]. Disponible en: <https://www.tiny.cloud/>