



10-3-2014

Observaterra 4.1

Documentación

Sergio Jiménez UO226261
Carlos Miguel UO218695
Jorge Alonso UO197637
Pelayo Gómez UO221205
Roque Carrizo UO217301
Daniel Tella UO218105

Contenido

Planteamiento de problema	3
Problema.....	3
Diagrama contextual	3
Metodología usada.....	4
Identificación de los Interesados (Stakeholders).....	4
Descripción de Stakeholders	4
Usuarios/Clientes	4
Directores del proyecto.....	4
Desarrolladores.....	4
Lista actualizada de Stakeholders	5
Atributos de calidad.....	6
Disponibilidad	6
Modificabilidad	6
Rendimiento	6
Seguridad.....	6
Testabilidad	6
Usabilidad	6
Primer acercamiento a la solución.....	7
Riesgos	7
Solución a los mismos	8
Repercusiones	8
Atributos de calidad	9
Atributos de calidad e interesados.....	10
Funcionamiento	12
Funcionamiento de la gestión de usuarios	12
Funcionamiento del control de datos	12
Escenarios de calidad.....	13
Vistas.....	15
System view.....	15
Presentación principal.....	15
Catálogo de elementos	15
Elementos.....	15
Interfaces/Puertos	15
Comportamiento.....	16
Justificación de las decisiones	16

Observaterra view	17
Presentación principal.....	17
Catálogo de elementos	18
Elementos.....	18
Comportamiento	18
Justificación de las decisiones	19
Package Model View	20
Justificación de las decisiones	20

Planteamiento de problema

Problema

La ONG Internacional Observaterra necesita una aplicación en la cual se facilite el acceso, lectura y escritura a los datos recogidos por las diferentes organizaciones que la forman.

Estos datos podrán ser consultados por todo aquel que lo desee aunque con diferentes niveles de privilegios y compartidos en caso de que así lo desee.

Algunas funcionalidades específicas será la posibilidad de comparar lecturas y la representación de ella en tablas o graficas según se quiera.

En caso de querer añadir nuevos datos a la aplicación el sistema deberá contemplar dos casos posibles:

- Datos añadidos por los interesados: se deberán registrar en la aplicación y su actividad y aportes estarán a disposición de los administradores

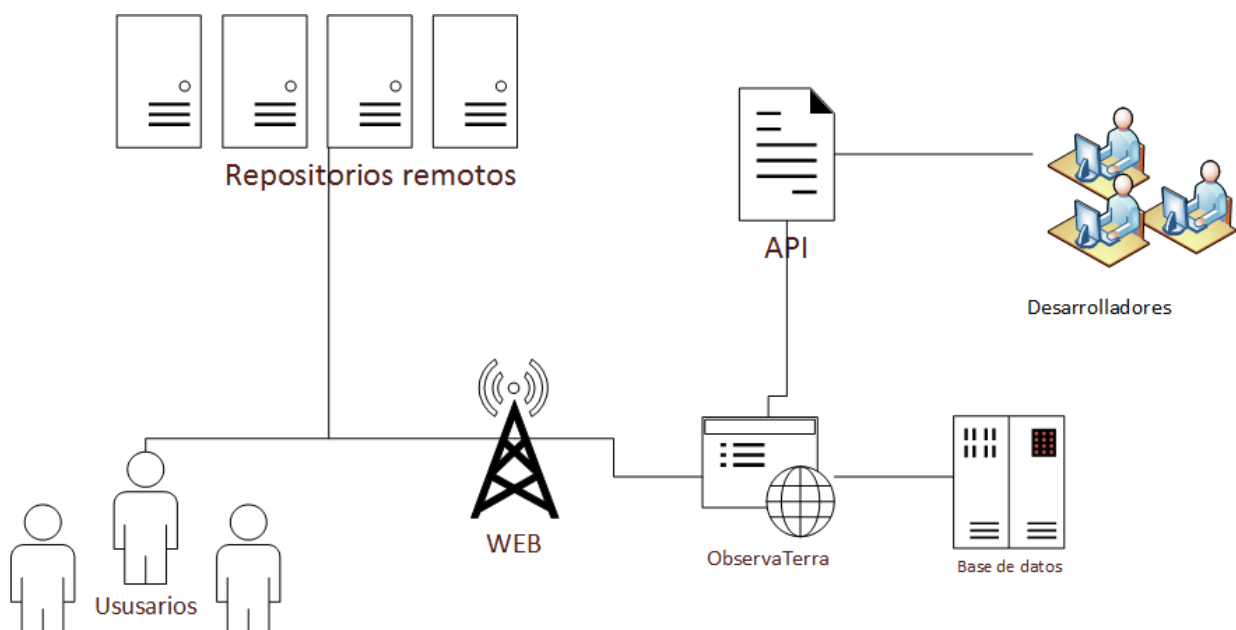
- Actualizaciones automáticas de información: Cada cierto tiempo aun no estipulado, el sistema lanzara peticiones de información a las organizaciones que nos faciliten sus datos de forma automatizada.

Teniendo en cuenta que las peticiones son automáticas, hay que garantizar que las actualizaciones se realizan en un plazo de tiempo razonable y no influye sobre el funcionamiento normal de la aplicación (perdidas de conexión, archivos corruptos, pérdida de paquetes,...).

En ambos casos nuestra aplicación será la encargada de leerlos y guardarlos de la forma más adecuada. Si los datos recibidos estuviesen en diferentes formatos, estos serán parseados a un formato genérico para todos ellos.

Para terminar, los datos alojados en nuestra aplicación deben estar a disposición de todas aquellas personas que quieran consultarlos o compartirlos en sus portales Web.

Diagrama contextual



Metodología usada.

Se va a realizar un estudio de arquitectura siguiendo el método de pair programming y big ball of mud.

Identificación de los Interesados (Stakeholders)

Los principales stakeholders del sistema:

- Usuarios/Clientes
- Directores del proyecto
- Desarrolladores

Descripción de Stakeholders

Usuarios/Clientes

En esta aplicación habrá diferentes tipos de clientes:

- Sin registrar
- Miembros de organizaciones
- Administradores

Todos ellos accederán y usarán la aplicación con diferentes privilegios y funcionalidades pero en general sus objetivos son los mismos:

- Que la aplicación sea usable y fácil de entender.
- Que no sea muy lenta, sin mucho tiempo de espera.
- Seguridad en el acceso.
- Recursos compartidos para su visualización en otros clientes

Directores del proyecto

En este caso son los profesores de la asignatura y deciden los requisitos y plazos para la aplicación.

Sus objetivos son:

- Que se pueda modificar fácilmente.
- Que funcione correctamente y muestre los resultados adecuados para cada petición.
- Que esté finalizada a tiempo.

Desarrolladores

El equipo que diseñará e implementará la aplicación. Sus objetivos son:

- Que la aplicación sea fácil de mantener.
- Que el coste de desarrollo no sea demasiado elevado.
- Facilidad para añadir nuevos sistemas de recolección de datos.
- Facilidad para la lectura y utilización de la API

Lista actualizada de Stakeholders

Código	Stakeholder	Intereses
ST-01	Usuarios/Clientes	Aplicación sea usable y fácil de entender
		No muy lenta, sin mucho tiempo de espera
		Seguridad en el acceso
ST-02	Directores del proyecto	Fácilmente modificable
		Funcionamiento correcto
		Entrega dentro de plazo
ST-03	Desarrolladores	Facilidad para mantener la aplicación
		Coste de desarrollo no muy elevado
		Interactuación con la aplicación mediante API

Atributos de calidad

Los atributos de calidad identificados para esta aplicación son:

Disponibilidad

- Por ahora sólo se requiere que la aplicación esté disponible para comprobar su funcionamiento, no es una prioridad.

Modificabilidad

- Se deben poder incorporar nuevas fuentes de datos o modificar las existentes con facilidad.

Rendimiento

- Los datos deben descargarse antes de que deban empezar a descargarse de nuevo

Seguridad

- Se requiere seguridad para las cuentas de usuario y el acceso a la aplicación.
- También seguridad en el almacenamiento de los datos y asegurar su integridad.
- Garantizar que los clientes externos no tendrán acceso a mas información que la suministrada por nosotros

Testabilidad

- El funcionamiento de la aplicación se debe probar con facilidad.

Usabilidad

- La aplicación debe ser amigable con el usuario y fácil de entender y usar.
- La internacionalización es necesaria, teniendo en cuenta que los datos pueden estar en diferentes idiomas también.
- Los datos almacenados deben estar accesibles para compartirlos cuando sea desee.

Primer acercamiento a la solución

La solución aportada es una aplicación web basada en el modelo vista-controlador que permitirá a los usuarios acceder a través del portal ObservaTerra.

Riesgos

El sistema elegido conlleva algunos riesgos que es necesario identificar:

1. **Información por parte de los usuarios.** Debido a que los usuarios van a poder compartir información en el sitio web va a ser necesario realizar comprobaciones de los datos que se suben así como de las noticias o documentos y de los enlaces que estos puedan compartir.
2. **Demostrar la validez de los datos.** Debido a que los datos van a ser recopilados de varias fuentes e incluso de los propios usuarios va a ser necesario mostrar al usuario que los datos que visualiza son datos reales y no ficticios.
3. **Actualización de los datos.** A la hora de realizar las actualizaciones de datos cuando se encuentren nuevos en las fuentes de información va a ser necesario controlar que el sistema no se sature y conlleve un no correcto funcionamiento del sitio web y por supuesto de comprobar que se mantiene la integridad de los datos.
4. **Reutilización de Datos.** Se quiere que los datos alojados en nuestro sistema estén a disposición de aquellos usuarios que quieran compartirlos en sus páginas o aplicaciones en todo momento.

Solución a los mismos

1. **Información por parte de los usuarios.** Se restringirá el sistema de subida de datos a ciertos formatos. Además se comprobará que estos son lo que el usuario dice que son y no serán publicados hasta que sean corroborados. Para el tema de compartir documentos, noticias y enlaces es posible que se habilite una zona especial para ello donde el sitio web no se hace responsable de lo que los usuarios puedan compartir, por ejemplo, que los enlaces lleven a sitios web con contenido malicioso o no deseado.
2. **Demostrar la validez de los datos.** Para demostrar al usuario que los datos son reales y correctos se indicará siempre la fuente de estos y se corroborarán con expertos y con varias organizaciones.
3. **Actualización de los datos.** Las actualizaciones de datos se realizarán de manera totalmente externa al sitio de tal manera que este no sea afectado por ellas. Es posible que se fije un horario para realizarlas o incluso si es necesario se cerraría el sitio durante el periodo de actualización para evitar su no correcto funcionamiento.
4. **Reutilización de Datos.** La solución más común al problema es la creación y puesta en funcionamiento de una API (Application Programming Interface) que permita al cliente que lo desee comunicarse con nuestra aplicación a través de los servicios que nosotros le cedamos siendo el resto del contenido totalmente opaco para él.

Repercusiones

La solución al sistema y los riesgos encontrados han hecho que se incorporen nuevos escenarios de calidad y que se incremente el gasto de tiempo y de esfuerzo a la hora de realizar y controlar la seguridad del sitio web.

Atributos de calidad

Código	Descripción	Tipo de Atributo
AT001	Se deben poder incorporar nuevas fuentes de datos o modificar las existentes con facilidad	Modificabilidad
AT002	Los datos deben descargarse antes de que deban empezar a descargarse de nuevo	Rendimiento
AT003	Se requiere seguridad para las cuentas de usuario y el acceso a la aplicación	Seguridad
AT004	Seguridad en el almacenamiento de los datos y asegurar su integridad	Seguridad
AT005	El funcionamiento de la aplicación se debe probar con facilidad	Testabilidad
AT006	La aplicación debe ser amigable con el usuario y fácil de entender y usar	Usabilidad
AT007	La internacionalización es necesaria, teniendo en cuenta que los datos pueden estar en diferentes idiomas también	Usabilidad
AT008	La aplicación debe ser fácil de mantener para los desarrolladores	Mantenibilidad
AT009	Los datos almacenados estarán disponibles para otros clientes Web (API)	Usabilidad

Atributos de calidad e interesados

	ST-01	ST-02	ST-03
AT001		X	
AT002		X	
AT003	X		
AT004	X	X	
AT005			X
AT006	X		
AT007	X		
AT008			X
AT009	X		X

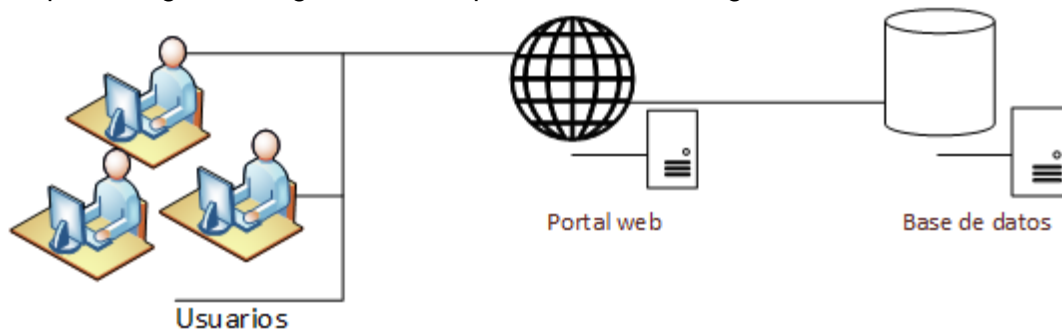
Descripción de negocio de la solución

Para describir la solución se emplean diagramas junto a un texto explicativo.

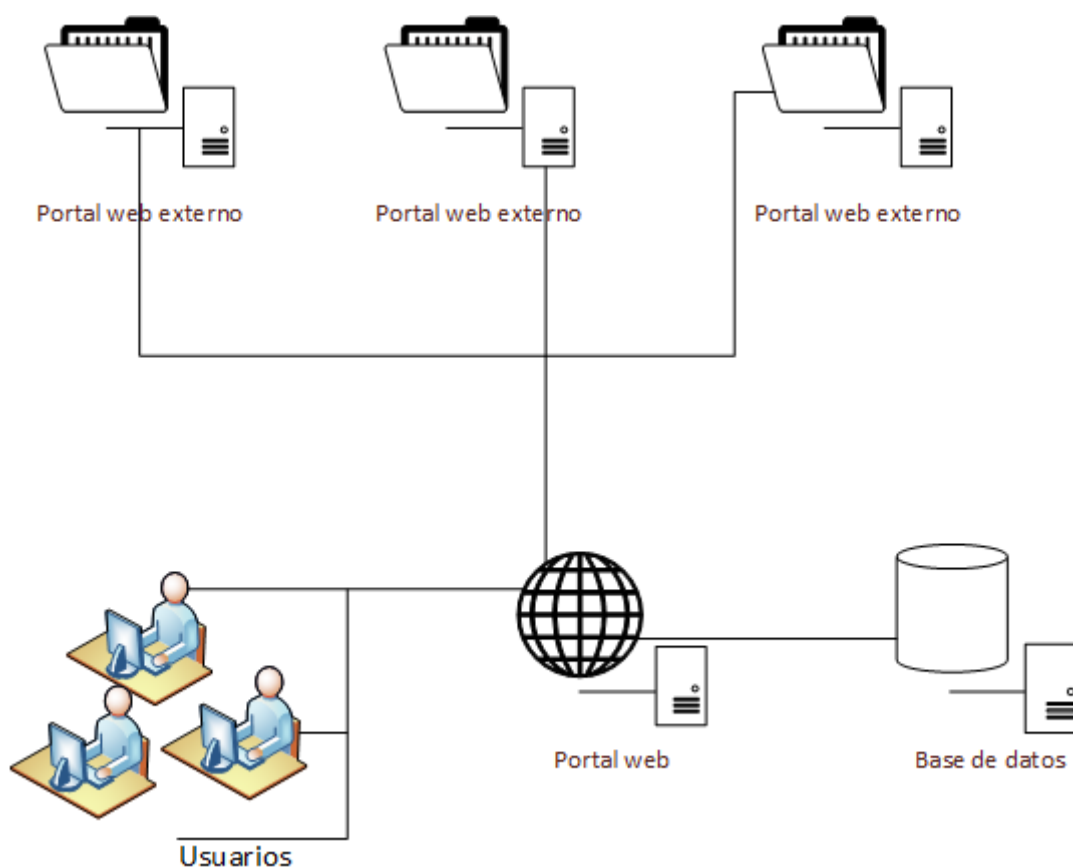
El modelo se basa en un portal web, en el cual un usuario que previamente se haya registrado y logueado podrá subir, compartir y comparar datos.

Aun sin registrarse u loguearse, un usuario gest podrá comparar datos pero no realizar otras operaciones.

En primer lugar un diagrama en el que se muestra el logueo de varios usuarios:



En segundo lugar se muestra el tratamiento del flujo de información, Los datos se transmiten hacia el portal desde otras web como la ONU o desde el portal a otros portales via API a los propios usuarios:



Solución

En nuestra arquitectura nos encontramos ante dos sistemas:

- Gestión de usuarios
 - Los usuarios registrados mediante la red comunicarse al portal web, que este verifica a través de la base de datos si el usuario existe y que nivel de privilegios posee.
 - Posee una ventana visual donde se muestra información pública y da la posibilidad de logueo de usuarios registrados, así como permisos especiales como el de permitir subir datos o compartirlos más adelante.
- Control de datos
 - Controla los datos que sube un usuario, así como establecer que tiene un formato correcto.
 - Obtiene información de otros portales.
 - Cede información que ha recogido a usuarios y otros portales.
 - Compara la información adquirida y la convierte de formato (parseando) en caso de necesidad.

Funcionamiento

Funcionamiento de la gestión de usuarios

El usuario no registrado puede únicamente visualizar el contenido de los datos sobre los que posea permiso de lectura.

Cada usuario registrado permite ampliar, comparar y modificar el repositorio de datos que se pretenden gestionar posteriormente en la aplicación web. Previamente ha de pasar por una ventana de logueo si desea realizar todas esas operaciones, y por supuesto también tiene el mismo permiso de lectura que posee un usuario no registrado agregado a los anteriores permisos mencionados (modificar, comparar y agregar contenido).

La idea del sistema es que este disponible siempre que un usuario lo requiera intentando aproximarnos a un sistema 24/7.

Funcionamiento del control de datos

Recoge la información pasada por los usuarios o por otros portales web, posteriormente la parsea en un formato estándar para visualizar en el sistema (tal como en HTML, XML, etc).

La información tomada se puede visualizar y comparar con otra ya recogida previamente para que un usuario pueda realizar estudios sobre esta, tales como comparación de tasas de natalidad de distintos años o información de las votaciones de un partido político por ejemplo. Además aquella información del repositorio del sistema se ha de permitir representar gráficamente mediante diagramas que también facilitarán al usuario la tarea de comparación de datos.

También será posible ofrecer nuestra información a quienes lo deseen. Esto incluye tener una copia de la misma o acceso a ella para mostrarlos como ellos prefieran

Por último la información ha de actualizarse cada cierto tiempo, para que en caso de un cambio en los datos, estos estén al día y de garantizar la veracidad de la misma para evitar lecturas erróneas o intentos de fraude.

Escenarios de calidad

Escenario nº	Fuente de estímulo	Estímulo	Entorno	Artefacto	Respuesta	Medición de la respuesta	Atributo de calidad afectado
1	Incrementar la cantidad de datos en el sistema	Obtener nuevos datos	Comunicación con el servidor	Gestor de peticiones remotas	Tiempo efectivo de descarga de datos	No precisa permisos para añadir datos	AT001
2	Actualizar o modificar las fuentes de datos existentes	Actualización de fuentes de datos	Comunicación con el servidor	Gestor de peticiones remotas			AT001
3	Descarga de datos	Descarga de datos vía web	Explotación	Gestor de peticiones remotas	Tiempo de descarga	Tiempo <= 60'	AT002
4	Seguridad para los usuarios	Conexiones seguras para usuarios	Explotación	Gestor de sesión	Inicio de sesión		AT003
5	Discriminación de tipos de usuarios	Diferir entre registrados y o no registrados	Explotación	Gestor de sesión	Inicio de sesión según rol	Rol correcto 0% fallo	AT003
6	Integridad de datos para los usuarios	Integridad de datos	Explotación	Gestor de datos	Datos íntegros y coherentes para el usuario	Obtener siempre los mismos datos al leer dos o más veces 0%fallo	AT004
7	Estabilidad para para aplicación	Estabilidad	Desarrollo	Sistemas y equipos de desarrollo	Aplicación robusta	0% de fallos inesperados no controlados	AT005
8	Aplicación fácil de usar	usabilidad	Explotación	Controlador de peticiones de usuario	Interfaz usable	<15% de rechazo	AT006
9	Aplicación accesible	Accesibilidad	Explotación	Controlador de peticiones de usuario	Interfaz accesible	Chequeado desde W3C cumpliendo más de AA	AT006
10	Aplicación multilingüe	Usuarios de diferentes nacionalidades necesitan traducción de	Explotación	Controlador de peticiones de usuario	Interfaz multilingüe	Idiomas: español inglés, ampliable posteriorme	AT007

		contenido				nte por la comunidad	
11	Aplicación mantenible	Necesidad de mantener la aplicación en el futuro	Desarrollo	ObservaTerra	Aplicación mantenible	Minimizar tiempos de corrección de errores	AT008
12	Datos compartidos	Ofrecer nuestros datos a quienes lo requieran	Explotacion	ObservaTerra	API	Siempre que el sistema esté disponible	AT009

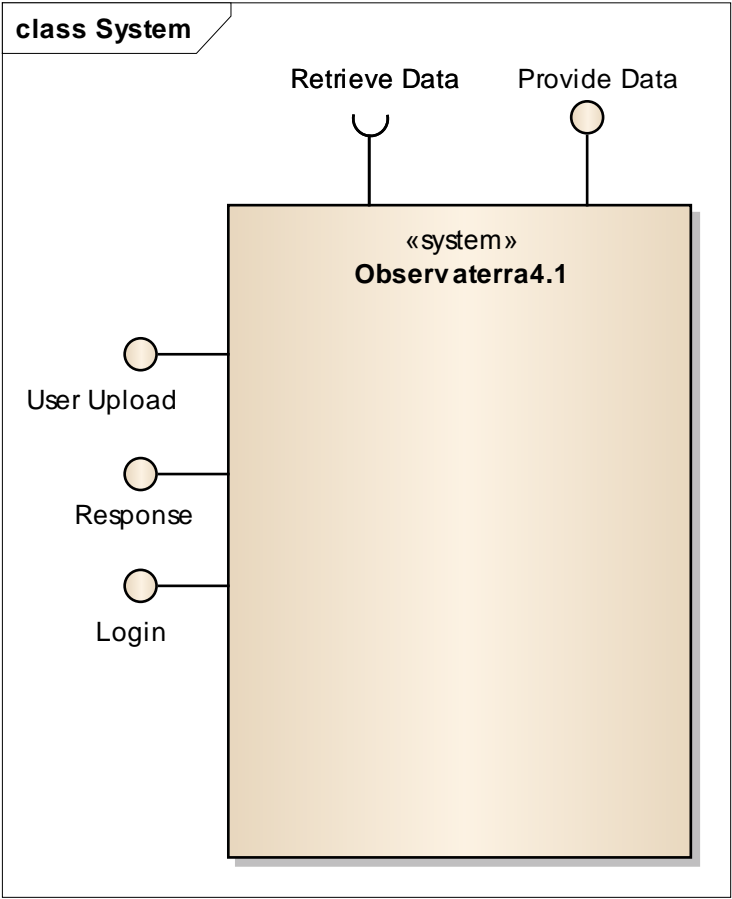
Vistas

En los próximos párrafos se describirán algunas de las vistas identificadas

System view

La vista de sistema describe los dos subsistemas en interacción

Presentación principal



Catálogo de elementos

Elementos

Interfaces/Puertos

ObservaTerra 4.1

Interface	Tipo	Tecnología	Propiedades
Retrieve data	Proveida	Web Service	Descarga los datos de url externas
Provide data	Requerida	Web Service Rest	Permite a otros usuarios usar nuestros datos (API)
User Upload	Proveida	Web Service	Permite a los usuarios registrados subir datos.
Response	Proveida	Web Service	Retorna al usuario aquellos datos que pida desde el portal web

Comportamiento

Observaterra es el portal web desde el cual un usuario puede logearse solicitar datos y enviarlos. Las interfaces UserUpload, Login y Respose son desde las cuales el usuario se loguea, recibe las respuestas del portal web y desde UserUpload se le facilita la caga de datos personales al portal.

La interfaz Provide data permite a otros desarrolladores acceder a los datos del portal web, por ultimo Retrieve data obtiene los datos de otras web.

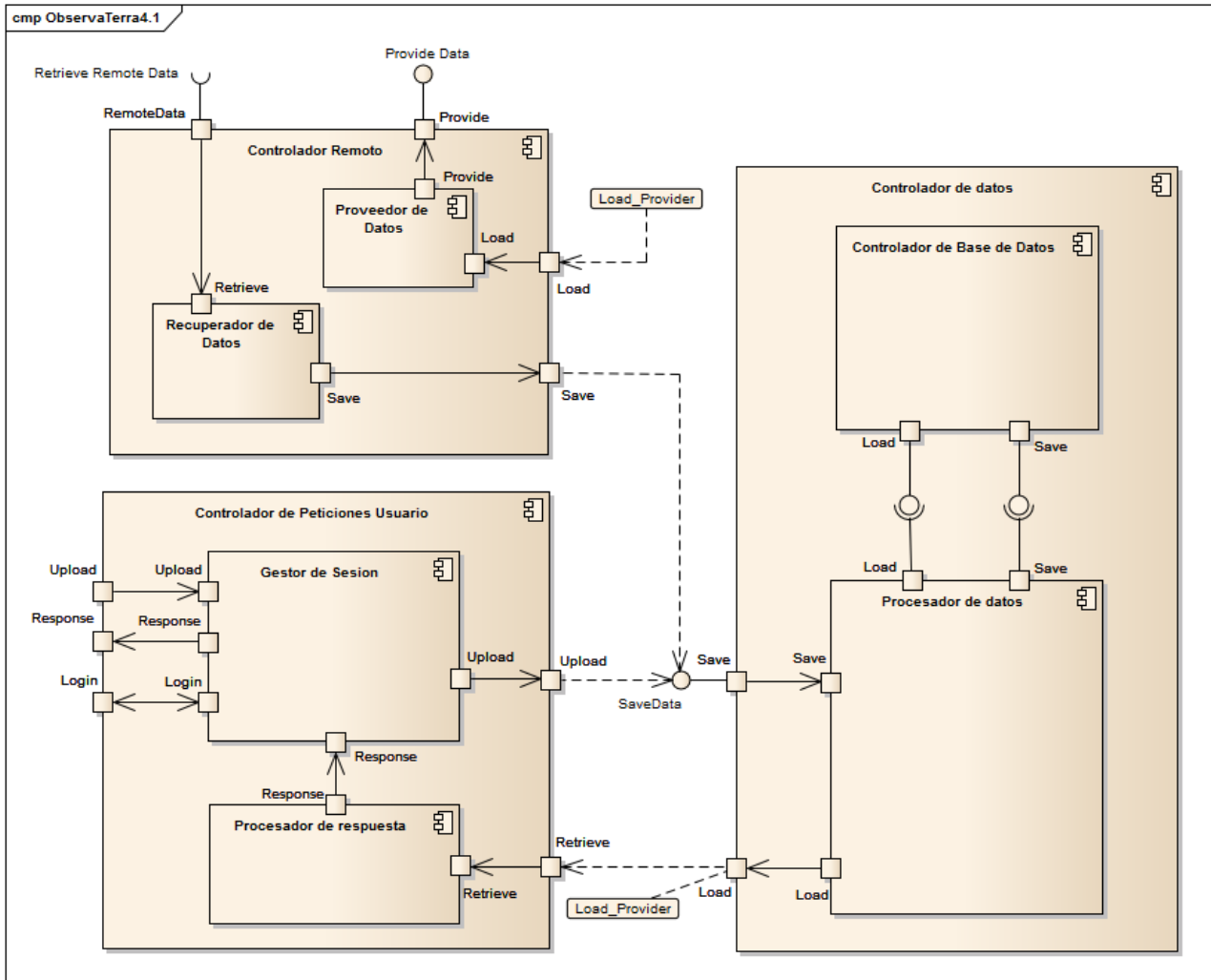
Justificación de las decisiones

Escenario	Atributos de calidad	Justificación
4 y 5	AT003	Crear la interfaz login
1 y 3	AT001, AT002	Crear las interfaces user upload y retrieve data

Observaterra view

La vista de *Observaterra* muestra el primer nivel de descripción de sus componentes

Presentación principal



Catálogo de elementos

Elementos

Elemento	Propiedades
Controlador remoto	Subsistema que controla todas las peticiones y solicitudes de datos con otros sistemas.
Proveedor de datos	Módulo que carga los datos de la bd para servirlos al usuario desde la api.
Recuperador de datos	Módulo que regularmente realiza peticiones a fuentes externas y dirige los datos recibidos hacia persistencia. Gestiona las fuentes externas.
Controlador de peticiones de usuario	Subsistema que controla todas las comunicaciones con los usuarios por el portal web.
Gestor de sesión	Módulo que regula las peticiones y los usuarios del sistema. Control de cuentas y acceso.
Procesador de respuesta	Módulo que procesa los datos y la petición y genera la respuesta adecuada.
Controlador de datos	Subsistema que controla la persistencia de los datos.
Controlador de base de datos	Módulo que gestiona la base de datos.

Comportamiento

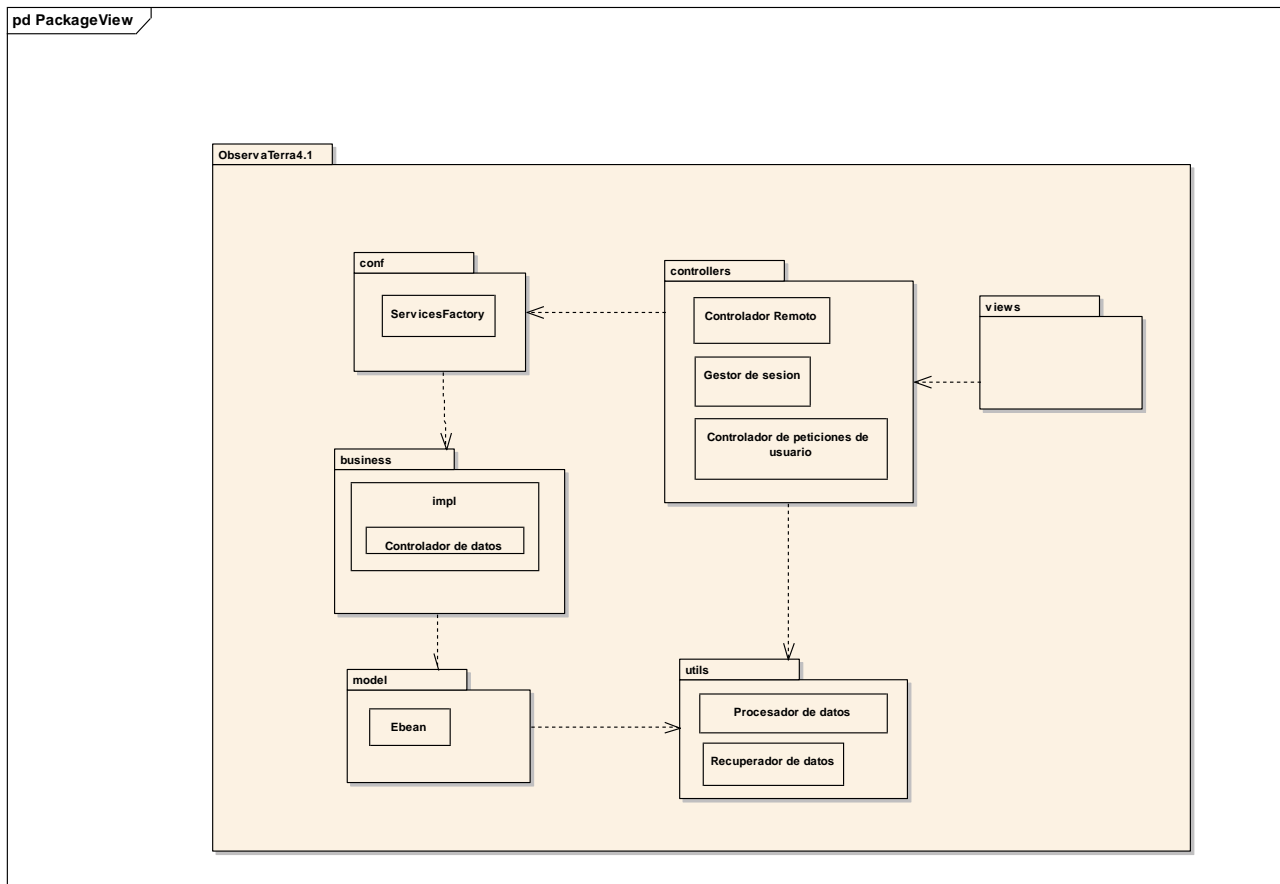
Nuestro sistema cuenta con un módulo de logueo para diferir y proteger los usuarios llamado controlador de peticiones de usuario, también contamos con un módulo de peticiones remotas encargado de la obtención de los datos en las otras webs, por último, el controlador de datos es el encargado de almacenar y gestionar tanto los datos enviados por los usuarios como los descargados por el controlador remoto.

Justificación de las decisiones

Escenario	Atributos de calidad	Justificación
2	AT001	Creación del <u>controlador de datos</u> pues se necesita un módulo que permita actualizar las fuentes de datos
6	AT004	Controlador de datos, concretamente el <u>controlador de base de datos</u> pues se precisa integridad de los datos
7	AT005	<u>Controlador de peticiones de usuario</u> ha sido creado para facilitar la estabilidad en la aplicación pues difiere los usuarios
8 y 9	AT006	<u>Procesador de respuesta</u> permite visualizar la información de una forma accesible y usable
10	AT007	Controlador de peticiones de usuario y concretamente en el <u>procesador de respuestas</u> que permitirá varios lenguajes
12	AT009	<u>Proveedor de datos</u> que permite servir los datos de la bd desde la api

Package Model View

La vista de paquete muestra información importante para el equipo de desarrollo del sistema: el empaquetamiento de los componentes.



Justificación de las decisiones

Escenario	Atributos de calidad	Justificación
7	AT005	La división de funcionalidad por paquetes permite que la aplicación se más estable y fácil de mantener
11	AT008	La división en paquetes según funcionalidad permite mantener los módulos concretos