

Escuela de Ingeniería Informática, Univ. Oviedo

4 de mayo de 2016

Sistemas de Gestión de Usuarios y Votaciones

Aquilino A. Juan Fuente

José Emilio Labra Gayo

Begoña Cristina Pelayo García-Bustelo

Jordán Pascual Espada

Vicente García Díaz

*Arquitectura Software para GestUsers, VotingSyst y Counting and Publication System. Descripción del entregable final(2016)*

Descripción de las prácticas primera y segunda a realizar por los grupos de trabajo de la asignatura de Arquitectura del Software durante el curso 2015-16.

**Grado de Ingeniería Informática del Software**

|  |  |
| --- | --- |
| G:\Mis Documentos\Documentos\Universidad\Asignatures\Asig-Grado\Arquitectura del Software\Teoría\logo.png | **G:\Mis Documentos\Documentos\Universidad\Asignatures\Asig-Grado\Arquitectura del Software\Logos\logo-color.png** |

**GestUsers: Sistema de Gestión de Usuarios**

**Versión:** 1

**Autores**: **D. Aquilino Adolfo Juan Fuente**

Doctor Ingeniero en Informática

Profesor del Departamento de Informática de la Universidad de Oviedo

**D. José Emilio Labra Gayo**

Doctor Ingeniero en Informática

Profesor del Departamento de Informática de la Universidad de Oviedo

**Da. Begoña Cristina Pelayo García-Bustelo**

Doctor Ingeniero en Informática

Profesora del Departamento de Informática de la Universidad de Oviedo

**Fecha**: 01 de enero de 2016

**VotingSyst: Sistema de Votaciones**

**Versión:** 2

**Autores:** **Ricardo Suárez Rodríguez**

Alumno de Ingeniería del Software de la Universidad de Oviedo

**Iván Modroño Álvarez**

Alumno de Ingeniería del Software de la Universidad de Oviedo

**Víctor Castaño Gutiérrez**

Alumno de Ingeniería del Software de la Universidad de Oviedo

**Lucas García Fernández**

Alumno de Ingeniería del Software de la Universidad de Oviedo

**Adrián Jiménez Villareal**

Alumno de Ingeniería del Software de la Universidad de Oviedo

**Fecha**: 30 de abril de 2016

Tabla de contenido

[1 Introducción y Objetivos 7](#_Toc450004647)

[2 Requisitos 8](#_Toc450004648)

[2.1 Censuses 8](#_Toc450004649)

[2.2 VoterInfo 8](#_Toc450004650)

[2.3 Sistema de votación 9](#_Toc450004651)

[2.4 Sistema de recuento y publicación. 9](#_Toc450004652)

[3 Metodología usada 11](#_Toc450004653)

[4 Identificación de los Interesados (Stakeholders) 12](#_Toc450004654)

[4.1 Alumnos que realizan la práctica 12](#_Toc450004655)

[4.2 Administrador del Sistema 12](#_Toc450004656)

[4.3 Votantes 12](#_Toc450004657)

[4.4 Desarrolladores del Sistema Electoral 13](#_Toc450004658)

[4.5 Profesores de la asignatura 13](#_Toc450004659)

[4.6 Junta Electoral 13](#_Toc450004660)

[5 Atributos de calidad 14](#_Toc450004661)

[5.1 Lista de atributos de calidad 15](#_Toc450004662)

[5.2 Atributos de calidad e Interesados 17](#_Toc450004663)

[6 Restricciones 19](#_Toc450004664)

[6.1 Restricciones técnicas 19](#_Toc450004665)

[6.2 Restricciones organizativas 19](#_Toc450004666)

[7 Ámbito del sistema y contexto 20](#_Toc450004667)

[8 Escenarios de calidad 24](#_Toc450004668)

[9 Vistas 30](#_Toc450004669)

[9.1 Contexto 30](#_Toc450004670)

[9.1.1 Presentación principal 30](#_Toc450004671)

[9.1.2 Catálogo de elementos 32](#_Toc450004672)

[9.1.3 Justificación de las decisiones 35](#_Toc450004673)

[9.2 Census Reader 35](#_Toc450004674)

[9.2.1 Presentación principal 35](#_Toc450004675)

[9.2.2 Catálogo de elementos 35](#_Toc450004676)

[9.2.3 Diagrama contextual 37](#_Toc450004677)

[9.2.4 Justificación de las decisiones 37](#_Toc450004678)

[9.3 VoterInfo 38](#_Toc450004679)

[9.3.1 Presentación principal 38](#_Toc450004680)

[9.3.2 Catálogo de elementos 38](#_Toc450004681)

[9.3.3 Diagrama contextual 40](#_Toc450004682)

[9.3.4 Justificación de las decisiones 40](#_Toc450004683)

[9.4 VotingSyst 40](#_Toc450004684)

[9.4.1 Presentación principal 40](#_Toc450004685)

[9.4.2 Catálogo de elementos 41](#_Toc450004686)

[9.4.3 Diagrama contextual 43](#_Toc450004687)

[9.4.4 Diagrama de paquetes 43](#_Toc450004688)

[9.4.5 Justificación de las decisiones 43](#_Toc450004689)

[9.5 Recount 44](#_Toc450004690)

[9.5.1 Presentación principal 44](#_Toc450004691)

[9.5.2 Catálogo de elementos 45](#_Toc450004692)

[9.5.3 Diagrama contextual 46](#_Toc450004693)

[9.5.4 Diagrama de paquetes 46](#_Toc450004694)

[9.5.5 Justificación de las decisiones 47](#_Toc450004695)

[9.6 Instantiation 47](#_Toc450004696)

[9.6.1 Presentación principal 48](#_Toc450004697)

[9.6.2 Catálogo de elementos 48](#_Toc450004698)

[9.6.3 Diagrama contextual 49](#_Toc450004699)

[9.6.4 Diagrama de paquetes 49](#_Toc450004700)

[9.6.5 Justificación de las decisiones 49](#_Toc450004701)

[10 Bibliografía 51](#_Toc450004702)

# Introducción y Objetivos

El objetivo de este documento es definir la estructura de una arquitectura para gestión de usuarios que pueda ser reutilizada. Aunque el sistema aquí descrito tiene funcionalidad propia, el objetivo es que pueda integrarse como parte de un sistema general de voto electrónico.

La arquitectura aquí descrita forma parte del segundo entregable (incluye la arquitectura del primero) de la asignatura Arquitectura del Software, impartida en el grado de Ingeniería Informática del Software, Escuela de Ingeniería Informática, Universidad de Oviedo por los autores de este documento.

El sistema se ha descompuesto en dos partes: Sistema de votación para gestionar el proceso electoral y de la realización de las votaciones, y Sistema de Recuento y publicación, el cual realizará el recuento de los votos y podrá ir mostrando los resultados, cuya documentación e implementación queda a cargo del otro equipo (uno de los stakeholders). Todo esto teniendo en cuenta lo ya implementado hasta la fecha (versión 1).

En los próximos entregables, ambos sistemas deben casar a la perfección para que todo el trabajo realizado por los diferentes equipos quede unificado.

# Requisitos

La gestión de usuarios se hará siguiendo un esquema de dos módulos:

* Censuses: Carga de Censos
* VoterInfo: Comprueba que un usuario puede votar.

El proyecto de voto electrónico se descompondrá en dos módulos:

* Sistema de Votación: Gestión del proceso electoral y realización de las votaciones.
* Sistema de Recuento de Votos: Realizar cálculos oportunos una vez pasadas las elecciones.

## Censuses

El Administrador del sistema debe poder introducir los censos electorales. Dichos censos serán entregados por cada ayuntamiento con la lista de votantes y los datos del colegio electoral en el que deberán votar.

La Introducción de los censos se realizará a partir de ficheros Excel formados por filas de votantes, cada uno con la siguiente información:

* Nombre
* Email
* NIF
* Código de Colegio Electoral

Durante la importación del Censo se creará un usuario y una clave que le permita acceder al sistema para comprobar que está dado de alta, ver el colegio electoral en el que votará y durante el día de las elecciones, le permitirá emitir su voto.

Una vez importando un fichero Excel conteniendo un censo, se emitirán cartas para cada usuario comunicándole que ha sido añadido al Censo Electoral, su usuario y su clave de acceso. Las cartas serán emitidas como ficheros de texto.

(Opcional) El sistema permite emitir las cartas en otros formatos como Word ó PDF.

(Opcional) Si el fichero viene con errores, se detectarían y se enviarían los datos a un fichero de LOG para su posterior tratamiento.

(Opcional) El analizador de los datos de entrada debe ser configurable, ya que podrían venir los datos en diferentes formatos y no sólo en Excel. El sistema permite que en el futuro se amplíe de manera sencilla.

## VoterInfo

Los usuarios pueden acceder al sistema para comprobar que han sido dados de alta, a partir de la información de la carta recibida. Para ello se creará un servicio Web sencillo que tome como parámetros codificados en una llamada POST el nombre de usuario y la clave,y devuelva información sobre el código de colegio electoral del votante en caso correcto o informe del error en case incorrecto. Tanto los parámetros como la respuesta se enviarán en formato JSON.

## Sistema de votación

- Cada colegio electoral tiene asociado un número a partir del cual se accede a la información del mismo: circunscripción, comunidad autónoma, etc.

- El sistema de votación asocia cada voto a un colegio electoral (identificado por el número de colegio electoral). Cada colegio electoral formará parte de una circunscripción dentro de una comunidad autónoma.

(Opcional) Se pueden añadir otras informaciones sobre los colegios electorales que ofrezcan datos más reales.

(Opcional) Se puede crear un sistema de votación genérico en el que los datos de colegio electoral, comunidad, etc. sean parámetros del sistema. De esta forma, el sistema podría utilizarse en otros ámbitos (otros países y otros tipos de elecciones)

Este sistema se encargará de gestionar el proceso electoral y de la realización de las votaciones. La gestión del proceso electoral será realizada por la Junta Electoral y permitirá configurar los diferentes parámetros, como el día y horas de elecciones, opciones de voto, colegios electorales, etc. La realización de las votaciones se realizará durante un periodo de tiempo concreto (el día y horas estipuladas por la Junta Electoral).

Durante ese tiempo, los usuarios podrán emitir su voto en el sistema. Los votos serán registrados por el sistema en una base de datos, que será utilizada posteriormente por el sistema de recuento.

Como proyecto piloto, el sistema permitirá crear elecciones de tipo referéndum en las que se elige una opción entre varias. Sin embargo, el sistema debe ser extensible para poder aceptar otros tipos de elecciones.

- El sistema de votación almacenará los datos en una base de datos, los cuales serán leídos por el sistema de recuento y publicación. La tecnología a utilizar y la estructura de la base de datos deberá ser acordada por ambos equipos.

- El voto del usuario debe ser privado y tras emitirlo no podrá existir ninguna relación entre el votante y el voto.

- El diseño del sistema debe permitir implementar los requisitos opcionales de una forma sencilla, modificando el menor número de módulos posibles en pos de un menor acoplamiento.

- El sistema debe impedir votar fuera del tiempo estipulado por la Junta Electoral para ese fin.

- El sistema debe respetar la premisa de una persona, un voto.

## Sistema de recuento y publicación.

Una vez cerrado el tiempo de voto, el sistema accederá a la base de datos para extraer la información de los votos recogida por el sistema de votación (2.3).

Con los datos de los votos, se irá actualizando en tiempo real, un servicio web que muestre las estadísticas de la votación.

A medida que los colegios electorales vayan arrojando más información a la base de datos sobre el escrutinio, el sistema tendrá que ir repitiendo las operaciones para incluir dicha información, y actualizando los resultados.

Como resultado online, el sistema reflejará el porcentaje de votos escrutados (votos calculados y procesados sobre el total de usuarios registrados en el sistema) además de porcentaje de votos emitidos a cualquiera de las opciones disponibles.

# Metodología usada

Se va a realizar un estudio de arquitectura siguiendo el método de ADD (Atribute-Driven Design) (Bass, Clements, & Kazman, 2003) y la norma del SEI (ANSI/IEEE 1471, 2000).

La documentación sigue el esquema propuesto en la guía de aprendizaje de la asignatura y también se han tomado algunas secciones siguiendo las plantillas propuestas en arc42 (<http://arc42.org/>). Las plantillas están tanto en español como en inglés y alemán.

Existe un proyecto que usa esas plantillas para documentar una arquitectura de software sencilla sobre una aplicación de gestión de rutas de bicicleta. La documentación está disponible en la Web. Se Pude ver aquí:

<http://biking.michael-simons.eu/docs/index.html>

# Identificación de los Interesados (Stakeholders)

En este caso los interesados son:

1. Alumnos que realizan la práctica
2. Administrador del Sistema
3. Votantes
4. Desarrolladores del sistema electoral
5. Profesores de la asignatura
6. Junta Electoral

Así pues, la lista de Stakeholders (interesados) queda:

| Código | Stakeholder | Intereses (Módulos) |
| --- | --- | --- |
| ST-01 | Alumnos que realizan la práctica | Todos |
| ST-02 | Administrador del Sistema | Carga de Censos |
| ST-03 | Votantes | Gestor de Votaciones |
| ST-04 | Desarrolladores Sistema electoral | Gestor de Votaciones |
| ST-05 | Profesores de la asignatura | Todos |
| ST-06 | Junta Electoral | Gestor de Votaciones y Gestor de Configuración |

Tabla 1. Lista de Stakeholders e intereses

Posteriormente se pasa a describir en más detalle cada uno.

## Alumnos que realizan la práctica

Se trata de los equipos de desarrollo.

Entre sus objetivos están:

* Utilizar tecnologías y metodologías conocidas, minimizando los riesgos relacionados con el aprendizaje de las nuevas.
* Aprender técnicas de desarrollo de software colaborativo y profesional.
* Utilización de tecnologías similares a las del grupo con quien deberán integrarse posteriormente para evitar incompatibilidades.

## Administrador del Sistema

Es la persona que carga los Censos.

Entre sus objetivos están:

* Tecnologías sencillas de los ficheros de entrada.
* Ficheros que puedan leerse por los humanos.
* Ser capaz de automatizar el proceso de carga de censos
* Ser capaz de depurar el proceso de carga en caso de errores

## Votantes

Son los usuarios finales del sistema.

Entre sus objetivos están:

* Sencillez de acceso a los datos.
* Ser capaz de votar desde su casa de una forma segura.
* Ser capaz de consultar el estado del sistema
* Ser capaz de cambiar su información en el sistema, por ejemplo, la clave (Opcional)

## Desarrolladores del Sistema Electoral

Está formado por el equipo que desarrollará el Sistema Electoral. Entre sus objetivos están:

* Disponer de una forma sencilla de comprobar si un usuario puede o no votar
* Utilizar tecnologías fáciles de usar e interoperables con otros sistemas.
* Poder configurar los diferentes parámetros de forma sencilla, entre ellos el día y la hora de las elecciones.

## Profesores de la asignatura

Son los responsables de los resultados de la práctica.

Entre sus objetivos están:

* Proponer tecnologías que ayuden a los estudiantes a adquirir habilidades relacionadas con Arquitectura del Software mediante el desarrollo de un proyecto práctico.
* Mostrar a los estudiantes un ejemplo de documentación de arquitectura.

## Junta Electoral

Es la encargada de la gestión del proceso electoral.

Entre sus objetivos están:

* Poder configurar los diferentes parámetros de forma sencilla y eficiente, entre ellos el día y la hora de las elecciones, las opciones de voto, los colegios electorales, etc.

# Atributos de calidad

Para el sistema se han identificado los siguientes atributos de calidad:

* **Disponibilidad**
  + Disponibilidad del sistema para procesar datos 24x7.
* **Modificabilidad**
  + Facilidad de cambio de diversas partes de la aplicación: Cambiar el parser de entrada de censos.
  + Facilidad de cambio de diversas partes de la aplicación: Añadir un fichero de log
  + Facilidad para modificar partes de la aplicación: otros formatos de salida para las cartas personales
  + Facilidad de cambio de diversas partes de la aplicación: Permitir el cambio de contraseña.
  + Facilidad para modificar partes de la aplicación: otros formatos mediante negociación de contenido
  + Facilidad para incluir otras informaciones sobre los colegios electorales que ofrezcan datos más reales.
  + Facilidad para utilizar el sistema en otros países u otro tipo de elecciones.
* **Rendimiento**
  + El rendimiento del proceso de carga del censo es razonable
  + La consulta de información de un usuario por el servicio web debe ser rápida
  + El proceso de votación debe realizarse de forma rápida para evitar la frustración del usuario.
* **Seguridad**
  + Garantizar la confidencialidad de los censos.
  + Garantizar la privacidad del voto de cada usuario.
  + Garantizar que el sistema sólo permita emitir votos durante el período establecido por la Junta Electoral.
  + Garantizar que un votante sólo pueda ejercer su derecho a voto una vez, independientemente del medio elegido para hacerlo: voto presencial o electrónico.
* **Testabilidad**
  + Debe ser posible chequear automáticamente que los datos del censo se cargan adecuadamente
  + Debe ser posible chequear que el servicio Web se comporta de forma adecuada
  + Debe ser posible chequear que el sistema de voto se comporta de forma adecuada y que los votos se guardan de forma correcta.
  + Debe ser posible chequear que el número de votos emitidos coincida con el número de votantes que han ejercido el derecho a voto.
* **Usabilidad**
  + El sistema de carga de datos debe poder ser usado por usuarios administradores de sistema familiarizados con herramientas tipo Unix.
  + El sistema de votación debe poder permitir configurar los parámetros de día y hora de las elecciones, opciones de voto, colegios electorales, etc… de una forma cómoda.
  + El proceso de votación debe ser cómodo y fácilmente manejable para el usuario.
  + Uso de estándares web y los principios del diseño web responsable.
  + Los datos expuestos en el servicio web por el sistema de recuento y publicación (2.4) deben poder ser comprendidos por cualquier tipo de usuario que acceda al sistema.
* **Interoperabilidad**
  + El presente sistema será usado por el Sistema de Voto Electrónico, el cual delegará en el sistema actual la gestión de usuarios. El subsistema VoterInfo deberá ser utilizado por un proceso automático para consultar el estado de los usuarios que quieran votar.
* **Simplicidad**
  + Los cuatro sub-sistemas deberán ser simples y fáciles de desarrollar
* **Desplegabilidad**
  + El sistema debe ser fácilmente desplegable, especialmente en un servidor en la nube.
  + El sistema debe poder ser desplegado e inicializado con la información para funcionar con un tipo de votación en concreto.
* **Escalabilidad**
  + El software no debería cambiar al aumentar la escala del sistema.
  + El sistema debe poder realizar los cálculos pertinentes para sea cual sea el tipo de votación.
* **Pertinencia**
  + Garantía que los usuarios reciban los servicios que requieren

## Lista de atributos de calidad

| **Código** | **Descripción** | | **Tipo de Atributo** | | | **Módulo afectado** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **AT001** | Disponibilidad del sistema para procesar datos 24x7 | | Disponibilidad | | | VoterInfo | | |
| **AT002** | Facilidad de cambio de diversas partes de la aplicación: Cambiar el parser de entrada de censos. | | Modificabilidad | | | Censuses | | |
| **AT003** | Facilidad de cambio de diversas partes de la aplicación: Añadir un fichero de log. | | Modificabilidad | | | Censuses | | |
| **AT004** | Facilidad para modificar partes de la aplicación:  Añadir otros formatos de salida para las cartas personalizadas | | Modificabilidad | | | Censuses | | |
| **AT005** | Facilidad de cambio de diversas partes de la aplicación: Permitir el cambio de contraseña. | | Modificabilidad | | | VoterInfo | | |
| **AT006** | Facilidad para cambiar partes de la aplicación: procesar y devolver información en otros formatos mediante negociación de contenido. | | Modificabilidad | | | VoterInfo | | |
| **AT007** | Facilidad para incluir otras informaciones sobre los colegios electorales que ofrezcan datos más reales | | Modificabilidad | | | VotingSyst | | |
| **AT008** | Facilidad para utilizar la aplicación en otros países u otro tipo de elecciones. | | Modificabilidad | | | VotingSyst | | |
| **AT009** | El rendimiento del proceso de carga de datos en el censo es razonable (no demasiado lento, pero tampoco crítico) | | Rendimiento | | | Censuses | | |
| **AT010** | El sistema debe garantizar la confidencialidad de los datos de los usuarios | | Seguridad | | | Censuses y VoterInfo | | |
| **AT011** | El sistema debe garantizar la privacidad del voto de los usuarios. | | Seguridad | | | VotingSyst y VoteCounting | | |
| **AT012** | El sistema debe garantizar que sólo se permita emitir votos durante el período establecido por la Junta Electoral | | Seguridad | | | VotingSyst | | |
| **AT013** | El sistema debe garantizar que un votante sólo pueda votar una vez, independientemente del medio elegido para hacerlo | | Seguridad | | | VotingSyst | | |
| **AT014** | Debe ser posible chequear que el servicio web se comporta adecuadamente | | Testabilidad | | | VoterInfo | | |
| **AT015** | Debe ser posible chequear el comportamiento del sistema de carga de datos | | Testabilidad | | | Censuses | | |
| **AT016** | Debe ser posible chequear que el sistema de voto se comporta de forma adecuada y que los votos se guardan de forma correcta | | Testabilidad | | | VotingSyst | | |
| **AT017** | Debe ser posible chequear que el número de votos emitidos coincide con el número de votantes que han votado | | Testabilidad | | | VotingSyst | | |
| **AT018** | El sistema debe poder ser usado por administradores de sistemas familiarizados con herramientas tipo Unix | | Usabilidad | | | Censuses | | |
| **AT019** | El sistema de votación debe poder permitir configurar los parámetros de día y hora de las elecciones, opciones de voto, colegios electorales, etc… de una forma cómoda | | Usabilidad | | | VotingSyst | | |
| **AT020** | El proceso de votación debe ser cómodo y fácilmente manejable para el usuario | | Usabilidad | | | Votingsyst | | |
| **AT021** | Uso de estándares web y los principios del diseño web responsable | | Usabilidad | | | VoteCounting | | |
| **AT022** | Cualquier usuario no familiarizado con gráficas y estadísticas debe poder comprender los datos sobre el escrutinio expuestos por el Sistema de recuento de votos. | | Usabilidad | | | VoteCounting | | |
| **AT023** | El servicio Web debe poder ser utilizado por procesos automáticos que consulten el estado de un usuario | | Interoperabilidad | | | VoterInfo | | |
| **AT024** | El sistema debe ser sencillo y fácil de implementar | | Simplicidad | | | Censuses y VoterInfo | | |
| **AT025** | El sistema debe ser fácilmente desplegable | | Desplegabilidad | | | Censuses y VoterInfo | | |
| **AT026** | Se le podrá indicar al sistema de recuento la información sobre el sistema de voto en curso. A raíz de dicha información calculará y mostrará los datos de una forma u otra. | | Desplegabilidad | | | VoteCounting y Voting System | | |
| **AT027** | | | El software no debe cambiar al aumentar la escala del sistema: debe ser capaz de atender un gran número de peticiones entrantes simultáneas. | | Escalabilidad | | Sistema de recuento y publicación | | | |
|  |  | |  | | |  | | |
| **AT028** | | El software debe poder ser fácilmente adaptable a otros sistemas de votación: cualquier tipo de representación de resultados y/o con más o menos formas de recuento de votos. | | Escalabilidad | | | | Sistema de recuento y votación | | |
|  |  | |  | | |  | | |
| **AT029** | | | | Garantizar que el sistema haga lo que el usuario requiere. | | Pertinencia | Sistema de recuento y votación | | | |
| **AT030** | | | |  | |  |  | | | |
|  | |  |  | | |  | | |

Tabla 2. Lista de atributos de calidad y tipos

## Atributos de calidad e Interesados

Los diferentes atributos de calidad son de interés para alguno de los Stakeholders. La siguiente tabla muestra la lista de intereses para el proyecto actual:

| **Atributos**  **vs**  **Interesados** | **ST-01** | **ST-02** | **ST-03** | **ST-04** | **ST-05** | **ST-06** |
| --- | --- | --- | --- | --- | --- | --- |
| **AT001** | X |  | X | X | X | X |
| **AT002** | X | X |  |  | X |  |
| **AT003** | X | X |  |  | X |  |
| **AT004** | X | X |  |  | X |  |
| **AT005** | X |  | X |  | X |  |
| **AT006** | X |  | X | X | X |  |
| **AT007** | X | X |  |  | X | X |
| **AT008** | X | X | X |  | X | X |
| **AT009** | X | X |  |  | X |  |
| **AT010** | X | X |  |  | X |  |
| **AT011** | X |  | X | X | X |  |
| **AT012** | X | X | X |  | X | X |
| **AT013** | X | X | X |  | X |  |
| **AT014** | X | X |  |  | X |  |
| **AT015** | X | X | X | X | X |  |
| **AT016** | X | X | X |  | X | X |
| **AT017** | X | X | X |  | X | X |
| **AT018** | X | X |  |  | X |  |
| **AT019** | X |  |  |  | X | X |
| **AT020** | X |  | X | X | X | X |
| **AT021** | X |  |  | X | X |  |
| **AT022** | X | X | X | X | X | X |
| **AT023** | X |  |  | X | X | X |
| **AT024** | X |  |  | X | X |  |
| **AT025** | X | X |  |  | X |  |
| **AT026** | X | X |  | X | X | X |
| **AT027** | X | X | X |  | X |  |
| **AT028** | X |  |  | X | X | X |
| **AT029** | X |  | X |  | X |  |
|  |  |  |  |  |  |  |

Tabla 3. Lista de intereses de los *stakeholders*

# Restricciones

Para realizar esta aplicación existen las siguientes restricciones

## Restricciones técnicas

| **Código** | **Restricción** | **Motivación** |
| --- | --- | --- |
| **TC001** | El lenguaje de programación será Java | Se asume que el equipo de desarrollo (ST001) tiene conocimientos de Java |
| **TC002** | Se utilizará una base de datos relacional para almacenar los datos | Se asume que el equipo de desarrollo (ST001) tiene conocimientos de bases de datos relacionales y existen múltiples librerías para trabajar con bases de datos relacionales desde Java |
| **TC003** | El servicio Web estará basado en estilo REST | El estilo REST es fácil de implementar y consumir. |
| **TC004** | Los datos de entrada vienen en formato Excel | Excel es un formato de datos bastante popular y existen varias librerías Java para procesar ficheros Excel |
| **TC005** | El formato de salida de las cartas personalizadas será texto plano | Con el fin de facilitar la implementación se propone generar cartas personalizadas mediante texto plano. El equipo de desarrollo puede opcionalmente implementar otros formatos |
| **TC007** | Pruebas automáticas y desarrollo basado en pruebas | Las pruebas deberán ser ejecutables automáticamente. Se propone un desarrollo basado en pruebas así como la utilización de técnicas de integración continua. |
| **TC008** | El servicio Web se implementará mediante el *framework* Spring Boot | El framework Spring Boot se basa en Spring, que es un framework Java muy popular en la industria. Existen muchos ejemplos y material de ayuda para facilitar el aprendizaje por parte de los estudiantes. |

Tabla 4. Restricciones técnicas

## Restricciones organizativas

| **Código** | **Restricción** | **Motivación** |
| --- | --- | --- |
| **OC001** | Cada sub-sistema será implementado por un equipo pequeño de estudiantes. | El tamaño de los equipos será de unos 3 ó 4 estudiantes con el fin de que los estudiantes puedan aprender a desarrollar software de forma colaborativa mediante un proyecto simple. |
| **OC002** | La estructura de la base de datos será la misma para los 2 sub-sistemas | El pegamento entre los 2 sub-sistemas es la base de datos, cuya estructura debe ser acordada por los 2 equipos. |
| **OC003** | El código fuente será gestionado mediante el sistema control de versiones Git en un repositorio público en github | Los sistemas de control de versiones son utilizados por la mayoría de las empresas de desarrollo de software. Github ofrece un software de gestión de proyectos muy potente |

Tabla 5. Restricciones organizativas

# Ámbito del sistema y contexto

Para describir la solución se utilizarán diagramas contextuales y texto.

La aplicación está partida en dos procesos:

* Censuses: Se encarga de la carga de los censos.
* Voters: Se encarga de las comprobaciones de los votantes.
* VotingSyst: Se encarga de gestionar el proceso electoral y de la realización de las votaciones.
* RecountingSyst: Recuenta los votos según el sistema de votación y los publica.

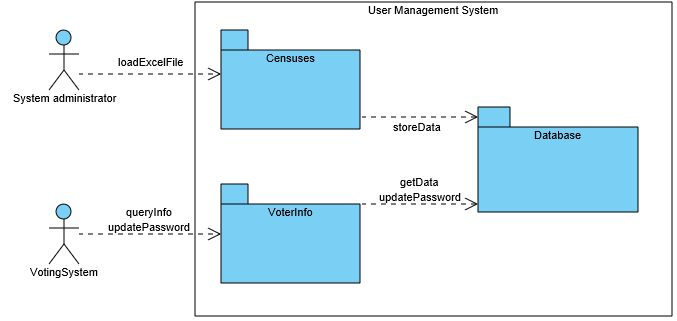
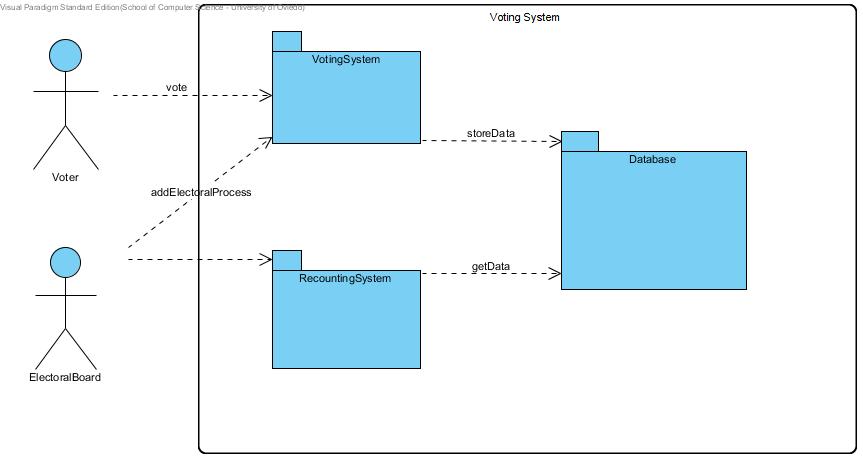


Figura 1. Contexto de negocio del sistema de gestión de usuarios



**Figura 2. Contexto de negocio del sistema de votación.**

A continuación, se incluye un diagrama de flujo de datos del sistema de gestión de usuarios:

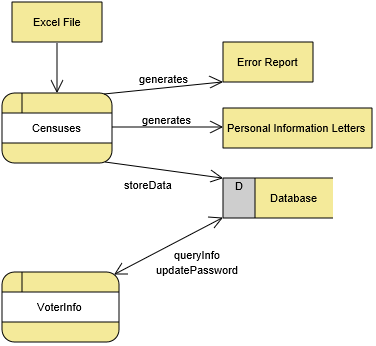
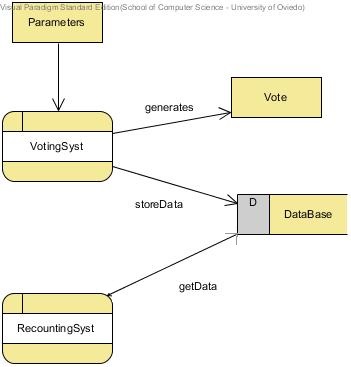
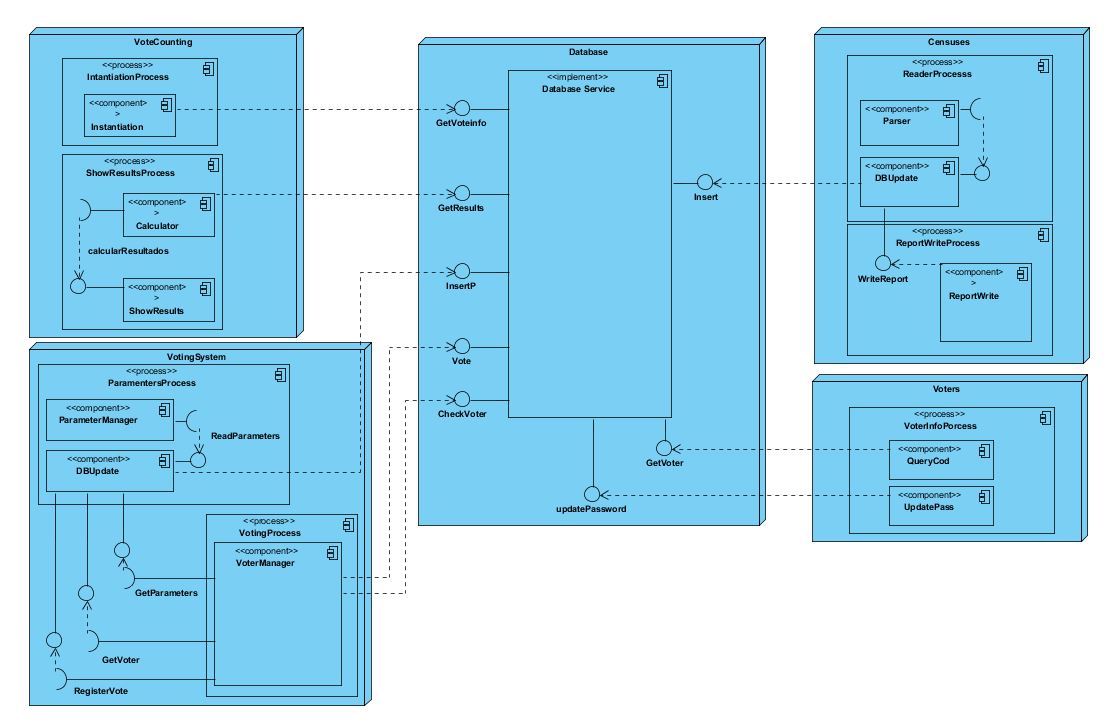


Figura 3. Diagrama de flujo de datos del sistema de gestión de usuarios

A continuación se incluye un diagrama de flujo de datos del sistema de votación:



**Figura 4. Diagrama de flujo de datos del sistema de votación.**

A continuación se incluye un diagrama de despliegue de la aplicación:

# Escenarios de calidad

Con toda la información anterior se procederá a definir los escenarios de calidad que influencian esta arquitectura.

En las próximas páginas se muestra una tabla con la lista de escenarios identificados.

| Escenario Nº | | Fuente de estímulo | | Estímulo | | Entorno | Artefacto | | | Respuesta | | Medición de la respuesta | Atributo de calidad afectado | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | | Sistema de votación | | Realiza una petición sobre un usuario | | Tiempo de ejecución | VoterAccess | | | El sistema de votación recibe la respuesta adecuada | | La información adecuada es recibida | **AT001** | | | |
| 2 | | Desarrollador | | Se introduce un nuevo Parser | | Desarrollo | Parser | | | La modificación es introducida adecuadamente | | El sistema es compilado y pasa todas las pruebas | **AT002** | | | |
| 3 | | Desarrollador | | Se implementa la generación de informes de error | | Desarrollo | ReportWriter, DBUpdate y Parser | | | La opción es implementada con cambios mínimos que solamente afectan al módulo de generación de informes | | Menos de un día de trabajo | **AT003** | | | |
| 4 | | Desarrollador | | A new output format is added | | Desarrollo | VoterAccess and DBManagement | | | The new output format is included with minimal changes to existing code. | | Menos de un día de trabajo | **AT004** | | | |
| 5 | | Desarrollador | | The option to change user's password is introduced | | Desarrollo | VoterAccess and DBManagement | | | The password of a user is successfully changes | | Menos de un día de trabajo | **AT005** | | | |
| 6 | | Desarrollador | | A new format is added to the web service | | Desarrollo | VoterAccess | | | The new format is implemented | | Menos de dos día de trabajo | **AT006** | | | |
| 7 | | Junta Electoral | | Sistema de votación | | Tiempo de ejecución | VotingManager y DBUpdate | | | Cargar los datos electorales sin errores | | La información adecuada es recibida | **AT007** | | | |
| 8 | | Desarrollador | | Sistema de votación | | Desarrollo | ParameterManager | | | Parametrizar los datos del colegio, comunidad, etc… para conseguir un sistema genérico reutilizable en otros países | | Menos de un día de trabajo | **AT008** | | | |
| 9 | | Administrador del Sistema | | Load an Excel file into the System (DB) | | Tiempo de ejecución | Parser, DBUpdate and ReportWriter | | | Loading an excel file without errors is done in a reasonable time. | | < 1 segundo por cada 10 votantes | **AT009** | | | |
| 10 | | Desarrollador | | Load an Excel file into the system (DB) | | Desarrollo/  Tiempo de ejecución | Parser, DBUpdate and ReportWriter (Optional) | | | Loading data should be done in a safe way | | No es posible acceder a los datos personales de los usuarios salvo el administrador del sistema, que tampoco puede acceder a las contraseñas. | **AT010** | | | |
| 11 | | Desarrollador | | Almacenar los votos en el Sistema (BD) | | Desarrollo | DBUpdate | | | El sistema debe desligar el voto del votante para garantizar la privacidad | | Debe quedar registrado que el votante votó y el voto debe ser contabilizado, pero sin relación entre ellos de ningún tipo | **AT011** | | | |
| 12 | | Desarrollador/ Voters | | Almacenar los votos en el Sistema (BD) | | Desarrollo/  Tiempo de ejecución | VotingManager , ParameterManager y DBUpdate | | | El Sistema únicamente registrará votos en el horario estipulado en los parámetros | | No es posible votar fuera del horario estipulado en la configuración | **AT012** | | | |
| 13 | | Desarrollador | | Almacenar los votos en el Sistema (BD) | | Desarrollo/  Tiempo de ejecución | VotingManager y DBUpdate | | | El Sistema debe garantizar que cada votante solo pueda ejercer su derecho a voto una única vez | | No es posible votar más de una vez independientemente del método elegido: presencial o electrónico | **AT013** | | | |
| 14 | | Voters | | Accede a la aplicación | | Tiempo de ejecución | VoterAccess | | | A user can get access to his data but not to other user's data | | El acceso a los datos se permite solamente cuando la información de email/contraseña son correctas. | **AT014** | | | |
| 15 | | Administrador del Sistema | | Carga un fichero Excel en la base de datos | | Tiempo de ejecución | Parser, DBUpdate y ReportWriter | | | El proceso de carga se realiza de una forma fiable y es posible chequear que los datos han sido cargados adecuadamente. | | No hay errores en la base de datos ni registros duplicados. Ningún votante tiene menos información que la requerida. | **AT015** | | | |
| 16 | | Desarrolladores del Sistema Electoral/  Voters | | Almacenar los votos en el Sistema (BD) | | Desarrollo/  Tiempo de ejecución | VotingManager y DBUpdate | | | El proceso de carga de los votos en la base de datos se realiza de una forma fiable y es posible chequear si ha sido correcto o no | | No hay errores en la base de datos. El voto ha sido contabilizado una vez | **AT016** | | | |
| 17 | | Desarrolladores del Sistema Electoral/  Junta Electoral | | Almacenar los votos en el Sistema (BD) | | Tiempo de ejecución | VotingManager y DBUpdate | | | Debe ser posible comprobar que el número de votantes coincide con el número de votos | | No hay errores en la base de datos. No puede haber discrepancia entre el número de votantes y el número de votos emitidos | **AT017** | | | |
| 18 | | Administrador del Sistema | | Carga un fichero Excel en la base de datos | | Tiempo de ejecución | Parser, DBUpdate y ReportWriter | | | El proceso de carga se comporta de una forma habitual y las opciones son fáciles de comprender | | El sistema muestra ayuda si el usuario la solicita. Los mensajes de error y otra información son comprensibles por personal técnico | **AT018** | | | |
| 19 | | Junta Electoral | | Paso de parámetros del sistema de votación | | Tiempo de ejecución | DBUpdate y ParameterManager | | | La configuración del sistema de votación debe poder realizarse de una manera sencilla | | No hay errores en la base de datos. El sistema muestra ayuda si el usuario la solicita. Los mensajes de error y otra información son comprensibles | **AT019** | | | |
| 20 | | Desarrolladores del Sistema Electoral/  Voter | | Procedimiento de votación | | Tiempo de ejecución | VotingManager y DBUpdate | | | El procedimiento a seguir para registrar los votos debe ser cómodo y manejable | | El sistema muestra ayuda si el usuario la solicita. Los mensajes de error y otra información son comprensibles | **AT020** | | | |
| 21 | | Desarrolladores del Sistema Electoral / Profesores de la asignatura | | Accede al servicio Web | | Tiempo de ejecución | Voting System, Counting and Publication System | | | El sistema muestra las gráficas según los estándares | | El sistema cumple con los estándares de accesibilidad. | **AT021** | | | |
| 22 | | Votantes | | Acceso al servicio web | | Tiempo de ejecución | Voting Sysemt, Counting and Publication System | | | La navegación por el servicio web es amigable e intuitiva para el usuario. | | La frustración del usuario a la hora de hacer uso del sistema es nula. | **AT022** | | | |
| 23 | | Sistema de votación | | Accede al servicio Web | | Tiempo de ejecución | VoterAccess | | | El sistema de votación solicita información sobre un usuario pasando una combinación de email y contraseña | | Se envía respuesta 200 OK si la combinación aparece en el sistema o error en caso contrario | **AT023** | | | |
| 24 | | Desarrollador | | Implementa el sistema | | Desarrollo | VoterInfo  Censuses | | | Los desarrolladores pueden implementar el sistema | | El sistema puede implementarse en 2 semanas | **AT024** | | | |
| 25 | | Administrador del Sistema | | Despliega el sistema | | Despliegue | Censuses, VoterInfo | | | El sistema es desplegado en un entorno de producción | | El sistema puede desplegarse en menos de una hora | **AT025** | | | |
| 26 | | Administrador del sistema | | Despliegue del sistema | | Despliegue | Counting and Publication System | | | El sistema se inicia según los datos de la votación propocionados | | El sistema es capaz de identificar esos datos y hace tanto los cálculos, como la representación de datos correcta para la forma de votación en curso. | **AT026** | | | |
| 27 | | Voters | | Llega un número alto de peticiones simultáneas al sistema | | Tiempo de ejecución | | | Voting System, Counting and Publication System | El sistema se mantiene activo y no se satura | | El sistema es capaz de recibir 1000 peticiones simultáneas sin colapsarse | | | **AT027** | |
|  | |  | |  | |  |  | | |  | |  |  | | | |
| 28 | | Desarrollador | | Se introduce una nueva forma de procesar los datos de una votación | | Desarrollo | | | Counting and Publication System | La nueva forma de proceso de datos es implementada | | El número de módulos a implementar/modificar es mínimo y no implicará más de 5 días | | | **AT028** |
|  | |  | |  | |  |  | | |  | |  |  | | | |
| 29 | | Voters | | Accede a la aplicación | | Tiempo de ejecución | | | Voting System | El sistema muestra los datos de los resultados de las votaciones | | No hay errores en la representación de los datos ni carencias de información de los mismos | | | **AT029** | |
|  | |  | |  | |  |  | | |  | |  |  | | | |

Tabla 6. Lista de escenarios de calidad

# Vistas

En los próximos párrafos se describirán algunas de las vistas identificadas y se documentarán de acuerdo a las instrucciones definidas en la guía de aprendizaje.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Vista | | Stakeholders | | Atributos de calidad | | Escenarios | |
| Context | | ST-01, ST-02, ST-03, ST-04, ST-05 | | AT018, AT022 | | 18, 22 | |
| Censuses | | ST-01, ST-02, ST-04, ST-05 | | AT002, AT003, AT004, AT009, AT010 y AT015, AT018, AT022 | | 2, 3, 4, 9, 10, 15, 18, 22 | |
| VoterInfo | | ST-01, ST-03, ST-04, ST-05 | | AT001, AT005, AT006, AT010, AT014, AT021, AT022 | | 1, 5, 6, 8, 14, 21, 22 | |
| VotingSyst | | ST-01, ST-02, ST-03, ST-04, ST-05, ST-06 | | AT007, AT008, AT011, AT012, AT013, AT016, AT017, AT019, AT020 | | 7, 8, 11, 12, 13, 16, 17, 19, 20 | |
| Recuento | | ST-01, ST-02, ST-06 | | AT009, AT011, AT022, AT027, AT028, AT029 | | 9, 11, 22, 27, 28, 29 | |
| Instanciación del Sistema de Recuento | | ST-01, ST-04, ST-05 | | AT026 | | 26 | |
|  | |  | |  | |  | |

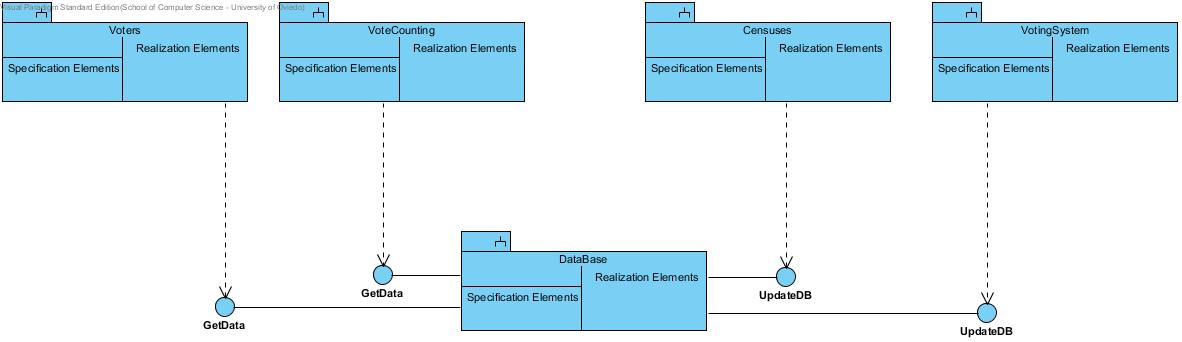
En los catálogos y en las vistas se han definido tanto los elementos obligatorios como opcionales que nuestro equipo va a implementar.

## Contexto

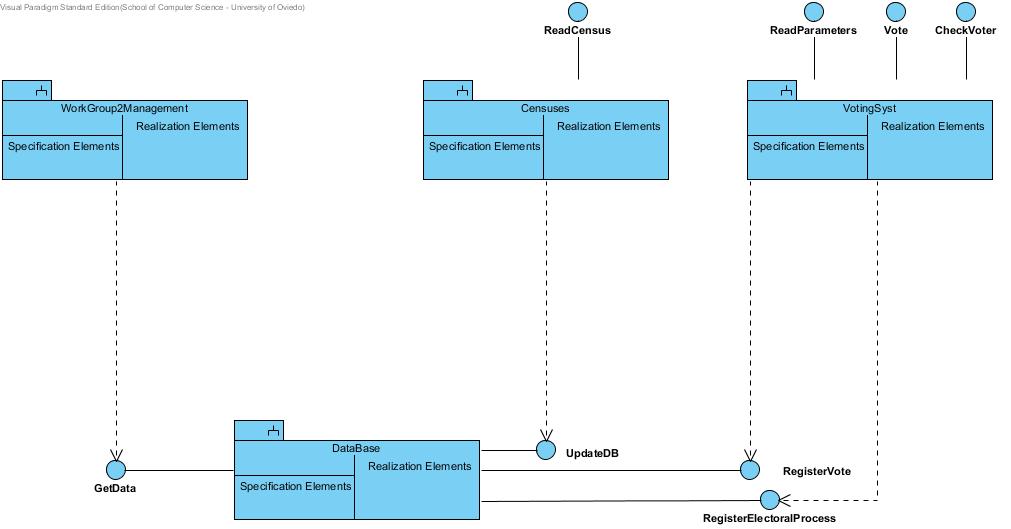
La vista de sistema describe los dos subsistemas en interacción así como sus interfaces.

### Presentación principal

Dividimos la presentación principal en dos diagramas basándonos en los subsistemas para simplificar la visualización.

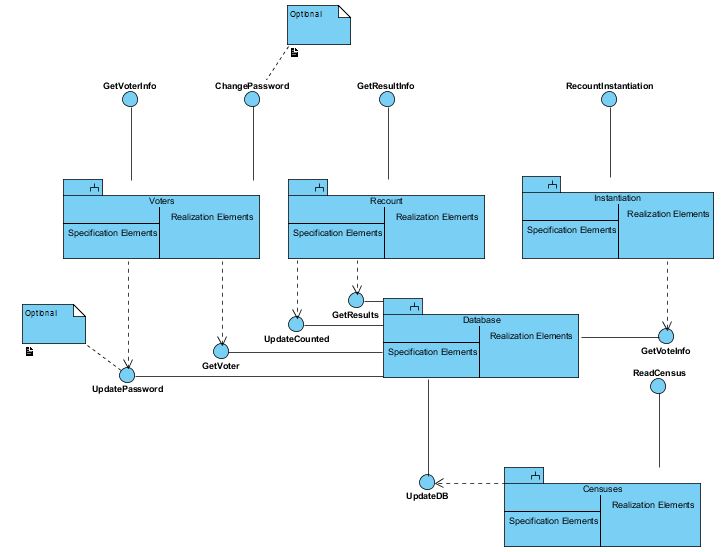


#### Presentación del sub-sistema “Voting System” y “Censuses”.



**Figura 6. Context view (versión 2)**

#### Presentación del sub-sistema “Voters” y “Counting and Publication System”.



### Catálogo de elementos

#### Elementos

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento | | Propiedades | |
| Censuses | | Se encarga de la introducción de los censos en el sistema. Lee un fichero con los datos de los votantes. Crea las calves. Crea las cartas para los usuarios dados de alta. | |
| Voters | | Es el módulo usado por los votantes para comprobar que han sido dados de alta y opcionalmente para hace el cambio de clave. | |
| DataBase | | Este módulo encapsula los accesos a la base de datos. | |
| VotingSyst | | Es el módulo encargado de crear los distintos tipos de elecciones y registrar el voto correctamente. | |
| Recount | | Este módulo es el encargado de realizar constantes accesos a la Base de Datos para calcular los votos nuevos que entren en el sistema. | |
| Instantiation | | Módulo encargado de obtener los datos de la votación de DataBase e instanciar el sistema de Recuento en función de dicha información. | |
|  | |  | |

#### Relaciones

Los datos de los votantes se introducen en el sistema a través de las interface *ReadCensus* del módulo *Censuses*. Para cada usuario, se crea una clave y emite una carta con todos los datos del usuario.

Este los envía a la base de datos a través de la interfaz *UpdateDB* del módulo DataBase.

El módulo *Voters* (versión 1, ver Figura 5) permite al usuario entrar en sesión a través del servicio web *GetVoterInfo* para comprobar sus datos. Para ello *Voters* pide los datos al módulo *DataBase* a través de la interfaz *GetVoter*.

La interface *ChangePassword (*versión 1, ver Figura 5*)* permite al usuario cambiar su clave, para ello el módulo *Voters* solicita al módulo *DataBase* el cambio de clave a través de la clase *UpdatePasswd*.

El módulo *VotingSyst* permite que la Junta Electoral establezca los parámetros de configuración del sistema de votación a través del servicio web *ReadParameters*. Para ello, *VotingSyst* envía los datos al módulo *DataBase* a través de la interfaz *RegisterElectoralProcess*.

Además, *VotingSyst* permite a los votantes y a los desarrolladores del sistema electoral entrar en sesión a través del servicio web *Vote* para emitir el voto electrónico, en el caso de los primeros, o para registrar a los votantes y las actas de las mesas de los colegios electorales (o cualesquiera sean los parámetros elegidos por la Junta Electoral) una vez finalizado el período de votaciones, en el caso de los segundos. Para ello, *Vote* envía los datos al módulo *DataBase* a través de la interfaz *RegisterVote*.

Por último, *VotingSyst* también permite a los desarrolladores del sistema electoral entrar en sesión a través del servicio web *CheckVoter* para consultar si un votante ha ejercido o no su derecho a voto. Para ello, *Vote* pide los datos del votante al módulo *DataBase* a través de la interfaz *GetVoter*.

#### Interfaces / Puertos

##### Censuses

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| ReadCensus | Interface | Invocación mediante línea de comandos | Se invocará a partir de la aplicación principal como un programa en consola |

##### Voters

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetVoterInfo | Interface | Servicio Web | Este servicio se invocará a través de una petición HTTP |

##### DataBase

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetVoter | Interface | Invocación a Método | Devuelve los datos de un votante. |
| UpdateDB | Interface | Invocación a Método | Inserta en la base de datos los datos de un votante, incluida su contraseña |
| UpdatePasswd | Interface | Invocación a Método | Actualiza la clave del usuario en la base de datos. |
| RegisterElectoralProcess | Interface | Invocación a Método | Inserta en la base de datos los parámetros de configuración del sistema electoral |
| RegisterVote | Interface | Invocación a Método | Inserta en la base de datos un voto o un conjunto de votos |

##### VotingSyst

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| ReadParameters | Interface | Servicio Web | Establece los parámetros de configuración del sistema electoral a través de una petición HTTP |
| CheckVoter | Interface | Servicio Web | Permite averiguar si un votante ha ejercido su derecho a voto a través de una petición HTTP |
| Vote | Interface | Servicio Web | Registra un voto o un conjunto de votos a través de una petición HTTP. |

##### Recount

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interface | Tipo |  | Tecnología | Propiedades |
| GetResultInfo | Interface |  | Servicio Web | El servicio es invocado a través de una petición HTTP y en la respuesta devuelve los resultados de la votación ya calculados |

##### 9.1.2.3.6 Instantiation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interface | Tipo |  | Tecnología | Propiedades |
| RecountInstantation | Interface |  | Invocación mediante línea de comandos | Se invocará como un programa de consola. |

#### Comportamiento

##### Censuses

Ver el párrafo 9.2.2.3.4.

El subsistema que crea las cartas implementa un patrón *Adapter* que permite cartas en el futuro en diferentes formatos (Word, ODT, PDF, RTF, ETC.).

Si el fichero viniera con errores, se detectan y se envían los datos a un fichero de LOG para su posterior tratamiento.

El *parser* de los datos de entrada es configurable mediante un patrón *Adapter*, para permitir datos en diferentes formatos (Excel, TXT, etc.).

##### Voters

Permite a los usuarios poder acceder al sistema para comprobar que han sido dados de alta, usando la información recibida en la carta. Los usuarios podrían no acceder directamente mediante un navegador Web, sino a través de un sistema de voto externo que invoca el módulo como un servicio Web.

##### DataBase

Todas las operaciones que realiza este módulo estarán integradas dentro de un patrón *Facade* donde estarán las operaciones servidas para el acceso a base de datos. Encapsula todas las operaciones de base de datos así como la tecnología usar.

##### VotingSyst

Permite a la Junta Electoral acceder al sistema para registrar la configuración de la votación, a los usuarios ejercer su derecho a voto y a los desarrolladores del sistema electoral comprobar o marcar que un votante ha votado y registrar las actas de las mesas de los colegios electorales (o cualesquiera sean los parámetros establecidos en la configuración) una vez finalizado el período de votaciones.

Los stakeholders podrían no acceder directamente mediante un navegador Web, sino a través de un sistema externo que invoca el módulo como un servicio Web.

##### Recount

Las operaciones que realiza este módulo están integradas dentro de un patrón de diseño *Abstract Factory.* De esta manera cada operación de recuento de votos, dependiendo del tipo de votación (referéndum, generales… etc) son independientes unas de otras, e implicaría poca modificación de código incluir alguna nueva. Estas operaciones son llevadas a cabo cada vez que se consultan votos no calculados en la Base de datos. Tras procesar dichos votos, se marcan como ya calculados.

##### Instantiation

Las operaciones que realiza este módulo son tres.

* Obtiene los datos de la votación en curso de la base de datos
* Interpreta dichos datos
* En función de los datos interpretados instancia el sistema de recuento y publicación (para que funcione de manera adecuada a la votación en curso).

### Justificación de las decisiones

Ver 9.2.4, 9.3.4, 9.4.5, 9.5.5, 9.6.5

## Census Reader

La vista de *Census Reader* muestra el primer nivel de descripción de los componentes.

### Presentación principal

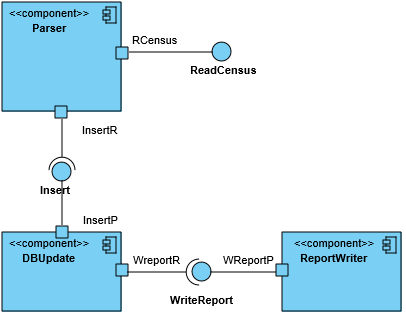


Figura 7. Vista Census Reader

### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Elemento | Propiedades |
| Parser | Lee los datos de entrada en Excel y los transforma en un contenedor de objetos que pueda ser recorrido para su inserción en la base de datos.  También crea el *usuario/password* del votante y la carta de comunicación.  Durante el diseño y la implementación hay que partir este componente en los subcomponentes necesarios para separar todos estos servicios y hacerlo de manera que se cumplan los atributos de calidad AT002, AT003, AT004 y AT007. |
| DBUpdate | Encapsula todas las operaciones de base de datos usando interfaces para permitir el acceso a la base de datos. |
| ReportWritter | Recibe cadenas de información con los datos del usuario que fue imposible de dar de alta y las razones de dicho fallo y los escribe en un fichero de texto secuencial, indicando toda la información necesaria para poder revisar visualmente los fallos. |

#### Relaciones

El componente *Parser* recibe el fichero de entrada en *Excel* y mediante un *parser* convierte éste en objetos. Añade a éstos objetos el email y el *password*, y lo añade a la base de datos utilizando el componente *DBUpdate*.

Si se producen errores en la carga de datos (DNI duplicados, campo DNI vacío, etc.) o si el componente *de la base de datos* devuelve un error, esta información se escribe en un fichero de *LOG* mediante la interface *WriteReport* y el componente Report*Writer*.

#### Interfaces / Puertos

##### Parser

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| ReadCensus | Interface | Invocación a Métodos | Lee el fichero de *Excel* con los datos del censo. |
| Rcensus | Port |  | Crea los subcomponentes del *parser* necesarios para procesar el fichero de entrada. |
| Insert | Interface (Requerida) | Invocación a Métodos | Llama a un método del componente *DBUpdate* para hacer la inserción en la base de datos. |
| InserR | Port |  | Verifica los datos y crea el objeto a enviar a *DBUpdate*. |

##### DBUpdate

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| Insert | Interface | Invocación a Métodos | Recibe un objeto con la información para insertar en la base de datos. |
| InsertP | Port |  | Verifica los datos de entrada y si falta algún atributo obligatorio genera el correspondiente error. |
| WriteReport | Interface (Requerida) | Invocación a Métodos | Llama a un método del componente *ReportWriter* para escribir una línea en el fichero de *log*. |
| WreportR | Port |  | Verifica los datos a escribir. |

##### ReportWriter

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| WriteReport | Interface | Invocación a Métodos | Recibe los datos para escribir en el fichero de *log*. |
| WreportP | Port |  | Añade a los datos la hora y la fecha. |

##### Parser

Introduce los censos en el sistema a partir de ficheros Excel formados por filas de votantes, cada una con la siguiente información (excepto la primera fila que contiene las cabeceras):

* Nombre (String)
* Email (String con una formato acorde a las convenciones de correo electrónico)
* NIF (String formado por dígitos seguidos de una letra de verificación)
* Código de Colegio Electoral (nº entero)

La invocación se hará mediante un programa *batch* ejecutado en línea de comando por el administrador del sistema. Durante la importación del Censo se creará un usuario cuyo nombre de usuario coincidirá con el correo electrónico y se generará una contraseña aleatoria. La combinación adecuada de email/contraseña permitirá al usuario entrar al sistema, acceder a su información y emitir su voto.

Este componente también creará cartas personales comunicando al usuario que ha sido añadido al Censo Electoral, e informando de su correo electrónico y su clave de acceso.

##### DBUpdate

Actualiza la base de datos. Ver 9.1.2.4.3.

##### ReportWriter

Guarda en un fichero de texto la información de los errores producidos en el proceso deconversión. La información básica a guardar es:

* Fecha
* Hora
* Fichero Excel de procedencia
* Descripción del error (con toda la información necesaria)

### Diagrama contextual

Ver 9.1.

### Justificación de las decisiones

Las decisiones que han llevado a este diseño son:

|  |  |  |
| --- | --- | --- |
| Escenario | Atributos de calidad | Justificación |
| 2 | AT002 | El acceso al *parser* mediante un patrón *Adapter* garantiza un cambio rápido de *parser* sin tocar el código ya realizado en otras partes de la aplicación. |
| 3 | AT003 | Prever una interfaz y un objeto que pueda estar vacío para el informe de errores (*WriteReport*) facilita la modificabilidad en caso de añadir esta funcionalidad posteriormente. |
| 5 | AT005 | La utilización de una base de datos relacional ofrecerá un acceso eficiente a la información de los usuarios |
| 6 | AT006 | Utilizar una base de datos con características de seguridad habilitadas podrá garantizar que los datos están aislados de accesos indebidos. El envío de la contraseña por carta evita que la información pueda ser leída por medios electrónicos. |
| 8 | AT008 | La utilización de una base de datos relacional con acceso mediante SQL puede permitir a los alumnos verificar que los datos han sido cargados adecuadamente |
| 10 | AT010 | La utilización de una aplicación *batch* que pueda ser ejecutada manualmente o configurada para su ejecución automatizada es una práctica común entre los administradores de sistemas |
| 14 | AT014 | Una aplicación *batch* independiente puede ser ejecutada directamente sin ninguna necesidad especial para su despliegue |

## VoterInfo

### Presentación principal

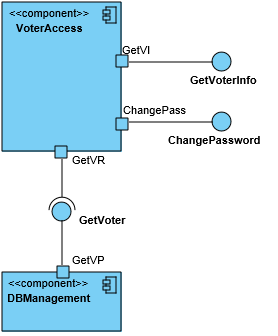


Figura 8. Vista de VoterInfo

### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Elemento | Propiedades |
| VoterAccess | Se accede a través de dos servicios web: *GetVoterInfo*, que permite al usuario acceder a sus datos en el sistema y (opcional) *ChangePassword* que permite al usuario cambiar su clave de acceso. |
| DBManagement | Se accede a través de dos interfaces: GetVoter, que devuelve los datos de un votante en la base de datos y (opcional) *UpdatePassword*, para actualizar un cambio de clave en la base de datos. |

#### Relaciones

El Sistema de Voto invoca *VoterAccess* utilizando una llamada a un servicio web que es procesada por *GetVoterInfo* (enviado *email/contraseña*) y éste accede a los datos encapsulados en DBManagement mediante la interface *GetVoter*. Si la combinación *email/contraseña* es correcta se devuelve la información del usuario en formato JSON.

El usuario (votante) accede a *VoterAccess* de forma manual al servicio web *ChangePassword* enviado *usuario/password/newPasswod*) y éste llama a la interface *UpdatePassword* para modificar la clave a través del componente *DBManagement*.

#### Interfaces / Puertos

##### VoterAccess

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetVoterInfo | Interface | Servicio Web | Permite el acceso a los datos de un votante mediante la combinación *email/contraseña*. |
| GetVI | Port |  | Hace la validación del usuario antes de pedir los datos. |
| ChangePassword | Interface | Servicio Web | Permite el cambio de clave a un votante mediante *email/contraseña/nuevaContraseña*. |
| ChangePass | Port |  | Hace la validación del usuario antes de solicitar el cambio de clave. |
| UpdatePassword | Interface (Requerida) | Invocación a Métodos | Solicita el cambio de clave para el usuario. |
| UpPssR | Port |  |  |
| GetVoter | Interface (Requerida) | Invocación a Métodos | Solicita la información para el usuario. |
| GetVR | Port |  |  |

##### DBManagement

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| UpdatePassword | Interface | Invocación a Métodos | Permite la solicitud de cambio de contraseña para el usuario. |
| UpPssR | Port |  |  |
| GetVoter | Interface | Invocación a Métodos | Permite la solicitud la de información para el usuario. |
| GetVP | Port |  |  |

#### Comportamiento

##### VoterAccess

Ver 9.3.2.2.

Implementa un servicio web REST para gestionar las peticiones de información sobre los usuarios. La petición principal será una petición HTTP POST que se realizará a la dirección:

<URIServicioWeb>/user

Donde URIServicioWeb representa la URI en la que está desplegado el servicio Web. La petición POST contiene datos JSON con la siguiente estructura:

{"login": email, "password": password}

En caso de que la combinación (email, password) aparezca en la base de datos, la respuesta será 200 OK con el cuerpo JSON de la forma:

{"name": name, "NIF": NIF, "email": email, "PollingStationCode": code}

En caso de que la combinación (email, password) no aparezca, la respuesta será 404 Not found.

Se puede implementar un interfaz HTML para que el servicio Web pueda también ser utilizado por personas a través de un navegador Web convencional.

El servicio Web puede ser extendido para permitir a los usuarios cambiar su password.

##### DBManagement

Encapsula todos los accesos a la base de datos.

### Diagrama contextual

Ver 9.1.

### Justificación de las decisiones

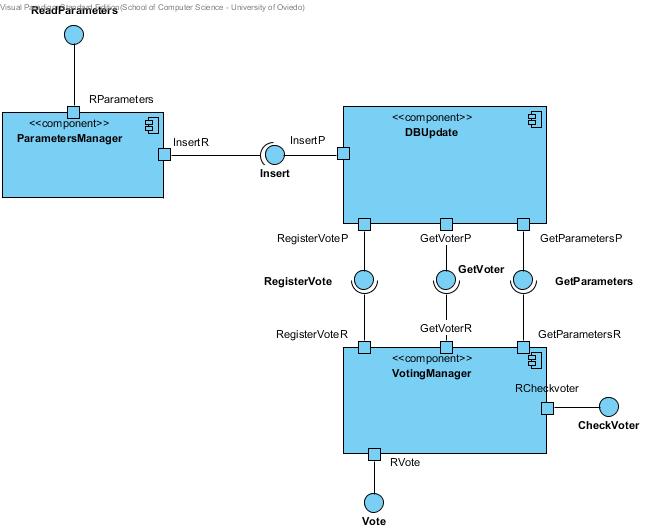
Las decisiones que han llevado a este diseño son:

| Escenario | Atributos de calidad | Justificación |
| --- | --- | --- |
| 1 | AT001 | La utilización de un servicio web REST se aprovecha de la tecnología HTTP y facilita el despliegue del sistema en infraestructuras de alta disponibilidad como pueden ser servidores Web, tanto locales como en la nube |
| 4 | AT005 | La encapsulación de las características del modelo que afectan a la base durante el desarrollo y la utilización de un framework basado en MVC facilitará el desarrollo de nuevas funcionalidades como las vistas basadas en HTML o el cambio de clabe de los usuarios. |
| 6 | AT006 | La utilización del framework Web Spring Boot facilitará el desarrollo de características comunes de la web como la negociación de contenido, dado que el framework ya contiene herramientas para su implementación. |
| 8 | AT008 | La restricción de acceso mediante *email/password* se considera suficientemente segura para este proceso. Las claves deberían almacenarse encriptadas. |
| 9 | AT009 | El desarrollo de un servicio web REST basado en formatos JSON facilitará la creación de pruebas. El framework Spring Boot contiene varias herramientas para pruebas unitarias y de integración. |
| 12 | AT012 | El uso de un servicio web REST permitirá el acceso automático al sistema a través de software cliente. |
| 13 | AT013 | El API del servicio web es simple y contiene la funcionalidad mínima necesaria. La utilización del framework Spring Boot facilitará el desarrollo por los estudiantes dado que el framework tiene soluciones para toda la funcionalidad requerida. |
| 14 | AT014 | La utilización del framework Spring Boot facilita el despliegue. Hay varios ejemplos que muestran cómo desplegar aplicaciones basadas en Spring Boot en servidores de producción. |

## VotingSyst

La vista del sistema de votación muestra el primer nivel de descripción de los componentes de este módulo.

### Presentación principal



### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Elemento | Propiedades |
| ParametersManager | Se accede a través de un servicio web, *ReadParameters*, que permite a la Junta Electoral introducir los parámetros de configuración en el sistema. |
| DBUpdate | Encapsula todas las operaciones de base de datos usando interfaces para permitir el acceso a la base de datos. |
| VotingManager | Se accede a través de los servicios web *Vote y CheckVoter*, los cuales permiten tanto a votantes como a desarrolladores del sistema electoral registrar los votos en la base de datos y consultar si un votante ha votado respectivamente. |

#### Relaciones

La Junta Electoral invoca *ParametersManager* mediante una llamada a un servicio web que es procesada por *ReadParameters* (enviando los parámetros de configuración del sistema electoral) y éste envía los datos a *DBUpdate* mediante la interface *Insert*. Si todos los parámetros son correctos el sistema queda preparado para procesar los votos mediante *VotingManager* según la configuración establecida.

El usuario (votante) accede a *VotingManager* de forma manual mediante el servicio web *Vote* enviado su id y su opción de voto y éste llama a la interface *RegisterVote* para registrar el voto a través del componente *DBUpdate*.

Cuando el derecho a voto se ejerce de forma presencial, los desarrolladores del sistema electoral acceden a *VotingManager* de forma manual para registrar que un votante ha votado enviando su id o registrar las actas de votación enviando el id del colegio electoral y el id de la mesa que figuran en el acta (o cualesquiera sean los parámetros establecidos en la configuración) y los votos obtenidos por cada partido, incluyendo los nulos y las abstenciones, mediante el servicio web *Vote*, llamando a la interface *RegisterVote* para registrar dichos datos a través del componente *DBUpdate*.

Los desarrolladores del sistema electoral también acceden a *VotingManager* de forma manual para comprobar si un votante ha ejercido su derecho a voto utilizando una llamada a un servicio web que es procesada por *CheckVoter* (enviado *el id de votante*) y éste accede a los datos encapsulados en *DBUpdate* mediante la interface *GetVoter.*

*VotingManager* recupera los parámetros de configuración del sistema electoral accediendo a los datos encapsulados en *DBUpdate* mediante la interface *GetParameters.*

#### Interfaces / Puertos

##### ParametersManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| ReadParameters | Interface | Servicio Web | Permite que la Junta Electoral establezca los parámetros de configuración del sistema de votación. |
| RParameters | Port |  | Hace la validación de los parámetros de configuración del sistema de votación. |
| Insert | Interface | Invocación a Métodos | Llama a un método del componente *DBUpdate* para hacer la inserción en la base de datos. |
| InsertR | Port |  | Verifica los datos y crea el objeto a enviar a *DBUpdate*. |

##### DBUpdate

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| Insert | Interface | Invocación a Métodos | Recibe un objeto con la información para insertar en la base de datos. |
| InsertP | Port |  | Verifica los datos de entrada y si falta algún atributo obligatorio genera el correspondiente error. |
| RegisterVote | Interface | Invocación a Métodos | Recibe un objeto con la información para insertar en la base de datos. |
| RegisterVoteP | Port |  | Verifica los datos de entrada y si falta algún atributo obligatorio genera el correspondiente error. |
| GetParameters | Interface | Invocación a Métodos | Envía un objeto de la base de datos con la información de la configuración del sistema electoral. |
| GetParametersP | Port |  | Crea el objeto a enviar a *VotingManager* con los datos recuperados de la base de datos. |
| GetVoter | Interface | Invocación a Métodos | Permite la solicitud de información de un votante. |
| GetVoterP | Port |  | Crea el objeto a enviar a *VotingManager* con los datos de un votante recuperados de la base de datos. |

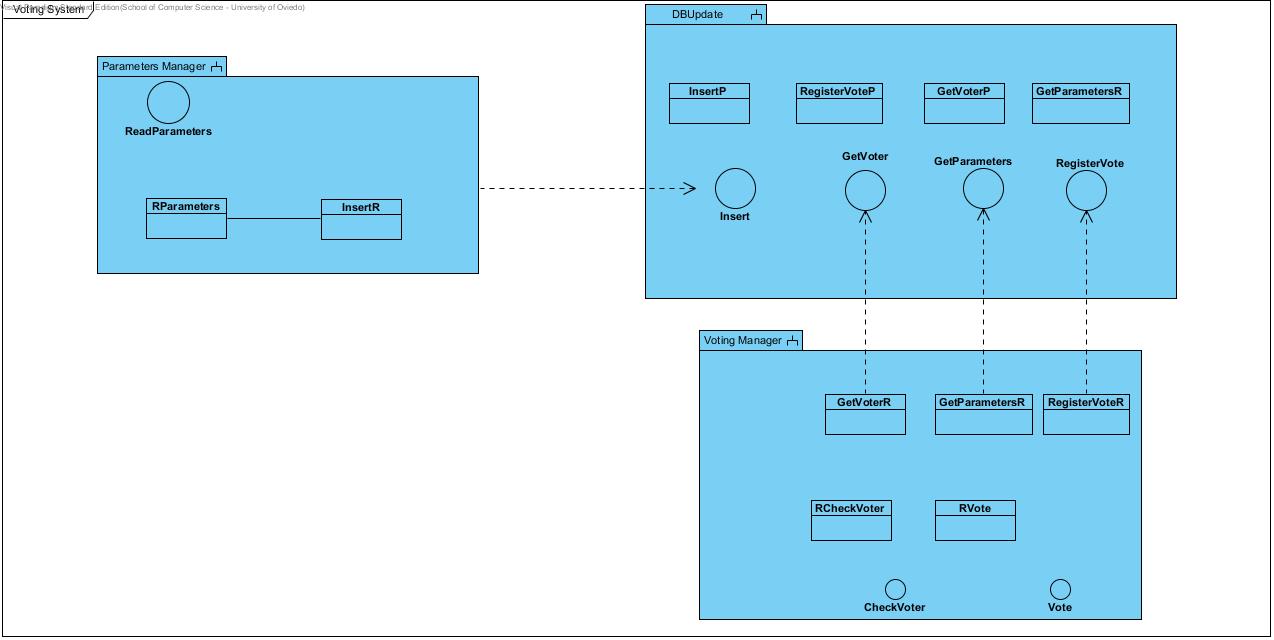
##### VotingManager

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| RegisterVote | Interface | Invocación a Métodos | Llama a un método del componente *DBUpdate* para hacer la inserción del voto en la base de datos. |
| RegisterVoteR | Port |  | Verifica los datos y crea el objeto a enviar a *DBUpdate*. |
| GetParameters | Interface | Invocación a Métodos | Recibe los parámetros de configuración del sistema de votación. |
| GetParametersR | Port |  | Recibe un objeto con los datos de la configuración. |
| Vote | Interface | Servicio Web | Permite que los desarrolladores del sistema electoral y los votantes registren los votos en el sistema. |
| RVote | Port |  | Hace la validación de los datos del voto. |
| CheckVoter | Interface | Servicio Web | Comprueba si un votante ha ejercido su derecho a voto |
| RCheckVoter | Port |  | Hace la validación de los datos de un votante. |
| GetVoter | Interface | Invocación a Métodos | Solicita la información de un votante. |
| GetVoterR | Port |  | Recibe un objeto con los datos de un votante. |

### Diagrama contextual

Ver 9.

### Diagrama de paquetes



### Justificación de las decisiones

| Escenario | Atributos de calidad | Justificación |
| --- | --- | --- |
| 7 | AT007 | El gestor de parámetros se ha separado del gestor de votaciones para facilitar la inclusión de otras informaciones sobre los colegios electorales que ofrezcan datos más reales |
| 8 | AT008 | Para facilitar el uso de la aplicación en otros países u otro tipo de elecciones se han parametrizado los datos de configuración y separado en dos gestores (configuración y emisión del voto) el sistema de votación. |
| 11 | AT011 | El gestor de votaciones no asocia el voto emitido con el usuario que lo emite en ningún caso, por tanto, la privacidad del voto de los usuarios está garantizada |
| 12 | AT012 | La restricción horaria establecida en los parámetros de configuración del sistema garantiza que sólo se permita emitir votos durante el período establecido por la Junta Electoral |
| 16 | AT016 | El sistema no se podrá alterar una vez son configurados sus parámetros de creación. Debería de garantizar la integridad del voto así como la confidencialidad del mismo. |
| 17 | AT017 | El sistema no deberá admitir más votantes que votos ni más votos que votantes, lo que implica que el proceso electoral siempre se hará de forma lícita. |
| 19 | AT019 | El API del servicio web es simple y contiene la funcionalidad mínima necesaria, así se agiliza la configuración de los distintos parámetros. |
| 20 | AT020 | El sistema debe proporcionar un proceso de votación intuitivo para los usuarios, sobre todo para personas mayores con más dificultades en el ámbito de la informática. |
|  |  |  |

## Recount

### Presentación principal

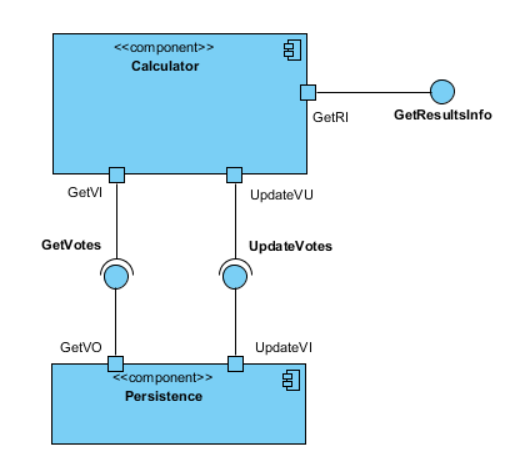


Figura 6. Vista de Recount

### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Elemento | Propiedades |
| Calculator | Se accede a él a través del servicio web GetResultsInfo. Este sistema devuelve al usuario la información de los resultados de las elecciones. |
| Persistence | Se accede a él a través de dos interfaces, GetVotes y UpdateVotes. GetVotes devuelve los votos que aún no hayan sido marcados como calculados. UpdateVotes permite marcar como calculados los votos que ya hayan sido leídos. |

#### Relaciones

El sistema Calculator es accesible a través del servicio web GetResultsInfo. Éste devuelve los datos guardados de los votos que ya se han procesado y por lo tanto, marcados como tal en la base de datos.

El sistema hará sucesivos y periódicos accesos a Persistence para comprobar y calcular, en caso de haber votos nuevos.

#### Interfaces / Puertos

##### Calculator

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetRI | Port | Bean JSF | Acceso a través de Java Beans para obtención de los datos desde la vista. |
| GetVI | Port |  | Recibe los datos de los votos solicitados, una vez lo permite la interfaz correspondiente. |
| UpdateVU | Port |  | Envía los datos necesarios para realizar la actualización del estado de un voto. |
| GetVotes | Interface | Servicio Web | Solicita los datos de los votos necesarios para la representación de los mismos. |
| GetResultsInfo | Acceso web | JSF | Solicita información de los resultados de la votación y los representa. |

##### Persistence

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| UpdateVI | Port |  | Valida los datos de actualización para comprobar que son correctos antes de realizarla. |
| UpdateVotes | Interface | Invocación a Métodos | Autoriza la actualización de los datos referentes a los votos que ya se han recontado. |
| GetVO | Port |  | Envía los datos de los votos solicitados. |
| GetVotes | Interface | Invocación a Métodos | Permite la solicitud de datos referentes a los votos. |

#### Comportamiento

##### Calculator

Implementará un servicio web REST, que siendo accedido mediante una petición HTTP a la URL {servicioweb}/resultados mostrará los resultados asociados a la votación en curso.

{servicioweb} se considera el servidor donde está desplegada la aplicación.

Si el servicio está activo y hay una votación en curso (o finalizada) éste devolverá el código HTTP de respuesta 200.

El sistema no recibirá ni validará ningún dato de entrada.

Además, en segundo plano, y ajeno al acceso web deberá hacer sucesivas lecturas de la base de datos para calcular los votos que no hayan sido marcos como tal ya.

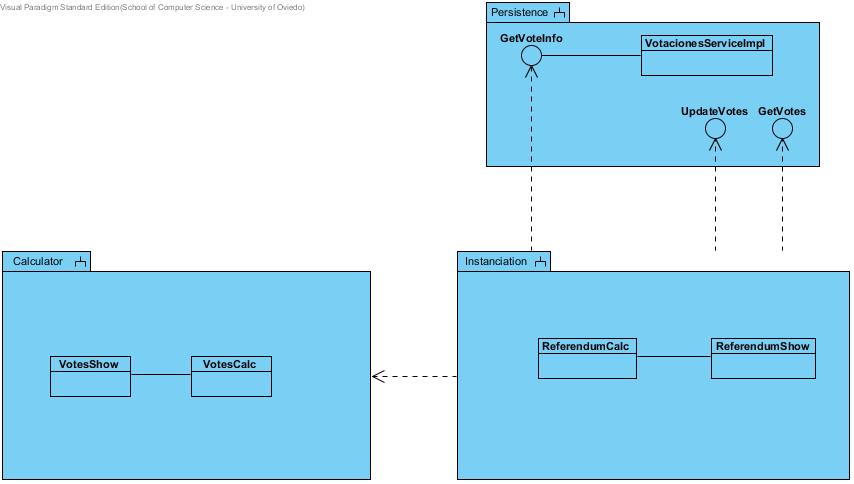
##### Persistence

Encapsula las operaciones permitidas con la base de datos: leer votos y marcar los leídos como tal. A este elemento se accede a través de una fachada.

### Diagrama contextual

Ver 9.1

### Diagrama de paquetes



### Justificación de las decisiones

Se han tomado las siguientes decisiones para llevar a cabo el diseño

| Escenario | Atributos de calidad | Justificación |
| --- | --- | --- |
| 1 | AT001 | Mediante el uso de un servicio web REST, permite la posibilidad de desplegar el sistema de manera constante en un servicio en la nube. De esta forma podrá estar disponible 24/7. |
| 15 | AT015 | Gracias a que es el elemento Calculator (ver 9.4.2.4.1) el que se encarga de, en segundo plano con el servicio web, procesar los votos, el tiempo de respuesta de dicho servicio se ve minimizado al no tener que hacer tareas de procesamiento de datos. |
| 17 | AT017 | El servicio web REST de representación de resultados estará diseñado en cumplimiento de los estándares y especificaciones desarrolladas por W3C. |
| 18 | AT018 | Los datos procesados, es decir, las cifras con los resultados de las votaciones, se almacenan en memoria. Esto garantiza un acceso rápido al servicio web y éste servirá las peticiones en un tiempo reducido. |
| 19 | AT019 | El sistema ha sido dividido en tres módulos para que sea fácilmente ampliable: por un lado, Base de datos, cuyas operaciones están integradas bajo un patrón Facade, el servicio web con un framework MVC (Spring Boot) y por último el Calculador de votos. Dicho módulo de cálculo de votos está diseñado bajo un patrón Abstract Factory con el fin de que no haya que modificar código existente a la hora de introducir una nueva forma de votación con su correspondiente forma de representación de resultados. |

## Instantiation

### Presentación principal

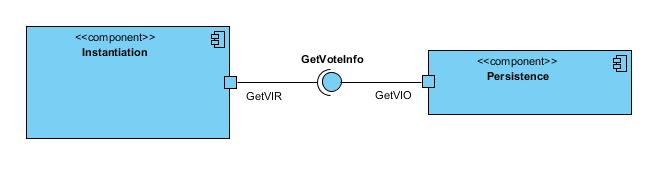


Figura 7. Vista de Instanciación

### Catálogo de elementos

#### Elementos

|  |  |
| --- | --- |
| Elemento | Propiedades |
| Instantiation | Cuando el administrador del sistema pone en marcha este elemento, carga de la base de datos la información de la votación e instancia el sistema de recuento (9.4). |
| Persistence | Se accede a él desde el elemento Instanciador a través de una interfaz: GetVoterInfo. Devolverá la información de la votación, necesaria para instanciar el sistema de recuento. |

#### Relaciones

El elemento Instanciador es iniciado por el administrador del sistema. No hace falta pasarle ningún parámetro o información acerca de la votación, ya que éstos se encuentran ya almacenados en la base de datos.

Automáticamente el elemento instanciará el módulo Calculator (haciendo uso del patrón de diseño Abstract Factory) quién hará las operaciones necesarias para calcular y representar los resultados de la votación.

#### Interfaces / Puertos

##### Instantiation

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetVIR | Port |  | Recibe los datos de la votación una vez se permite el envío de los mismos. |

##### Persistence

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Tipo | Tecnología | Propiedades |
| GetVIO | Port |  | Envía la información solicitada. |
| GetVoteInfo | Interface | Invocación a Métodos | Permite la solicitud de la información sobre la votación. |

#### Comportamiento

##### Instantiation

La ejecución se llevará a cabo a través de un sistema batch en línea de comando por el administrador del sistema el cual ejecutará el módulo de instanciación. Éste no recibirá ningún parámetro de entrada.

Dicho módulo creará el recuento de votación obteniendo los datos necesarios de la base de datos.

Si alguna de estas operaciones no se puede llevar a cabo ya sea porque la base de datos no esté encendida, o no haya la información requerida, el sistema no se iniciará.

##### Persistence

Encapsula el acceso a la base de datos para obtener los parámetros de la votación.

### Diagrama contextual

Ver 9.1

### Diagrama de paquetes

Ver 9.5.4

### Justificación de las decisiones

Se han tomado las siguientes decisiones para llevar a cabo el diseño

| Escenario | Atributos de calidad | Justificación |
| --- | --- | --- |
| 19 | AT019 | Independizar en módulos la instanciación del recuento y publicación, permite que a la hora de introducir el código para calcular y representar resultados de un tipo de votación nueva, no haya que modificar ni compilar el código más que el de Calculator. |
| 21 | AT021 | Instanciar el módulo de Recuento a partir de los datos obtenidos de la base de datos al arrancar la aplicación, permite que dicho módulo funcione para cualquier tipo de votación que tenga implementada, y se le indique por parámetro. |

# Bibliografía

ANSI/IEEE 1471. (2000). *Recommended Practice for Architectural Description of Software-Intensive Systems.* ANSI/IEEE.

Bass, L., Clements, P., & Kazman, R. (2003). *Software Architecture in Practice, Second Edition.* Boston: Addison Wesley.

## Para la toma de decisiones hemos obtenido información de:

http://resultadosgenerales2015.interior.es/#/