



ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

Curso Académico 2009/2010

Proyecto de Fin de Carrera

**UN SISTEMA DE VOTACIÓN ELECTRÓNICA DE APOYO A
PROCESOS DE TOMA DE DECISIONES**

Autor: Hugo Martínez Jara

Tutores: César Alfaro Gimeno y Javier Gómez Miguel

Agradecimientos

Por su implicación en este.

Dedicado a mis familiares, amigos y novia.

Antes de comenzar, me gustaría agradecer a todas aquellas personas que han colaborado en la realización de este proyecto. Ya sea directa o indirectamente, siempre he tenido el apoyo de numerosas personas. A todas aquellas, gracias.

Este proyecto no hubiera sido posible sin la ayuda de mis familiares. Gracias por haberme brindado la oportunidad de estudiar de manera libre y gracias por el apoyo recibido durante estos largos años de carrera.

Durante mi etapa en la universidad, he tenido la oportunidad de conocer a grandes amigos, grandes personas que han hecho que mi paso por la universidad haya sido inolvidable.

A mis amigos de siempre, gracias.

A mi novia, por apoyarme en los buenos y en los malos momentos. Gracias.

Por último, a los tutores que han hecho posible que pueda realizar el proyecto fin de carrera y por su implicación en este. Gracias por confiar ciegamente en un estudiante sin experiencia.

En definitiva, muchas gracias a todos los que confiaron en mí.

Resumen

En un primer momento pudiera pensarse que en el mundo electrónico la toma de decisiones es de lo más sencillo: no es sino una lista de votantes, una lista de opciones y un contador... nada más fácil. Pero, puntualicemos, en un programa de este estilo la sencillez es sólo aparente. Si buscamos en Internet programas de voto electrónico, nos encontraremos con una sorpresa: sólo existen programas de encuestas, no existiendo casi ninguno sobre votaciones

Este proyecto presenta como objetivo principal dar soporte informático a dicha funcionalidad a aquellos usuarios que estén interesados en realizar este tipo de actividad. El objetivo es poder realizar votaciones para diferentes acontecimientos mediante una aplicación web, donde un usuario puede dar su voto, entre una lista de candidatos y ver los resultados de todos los acontecimientos disponibles.

En este proyecto se tratará cómo se ha desarrollado el sistema, así como sus funcionalidades vía Internet. También se llevará a cabo la definición y explicación de los diferentes sistemas de votación. Además, la base teórica en la que se sustenta el proyecto incluye la presentación de algunos sistemas comerciales de voto electrónico.

Cabe destacar que la aplicación web forma parte de una arquitectura que consta de varios módulos. El sistema de votación es sólo uno de los módulos. Lo que se pretende es crear una plataforma electrónica de participación ciudadana.

Para implementar el proyecto se ha hecho uso de una plataforma Windows, utilizando herramientas de programación en Java, un web Tomcat, tecnología Hibernate y JPA para el manejo de la base de datos y la base de datos MySQL.

Índice

<i>Portada</i>	<i>I</i>
<i>Agradecimientos</i>	<i>III</i>
<i>Resumen</i>	<i>IV</i>

1.Introducción

1.1.Voto electrónico	<i>9</i>
1.2.Sistemas de votación	<i>11</i>
1.3.Sistemas comerciales de voto electrónico	<i>15</i>

2.Objetivos

2.1.Metodología	<i>20</i>
2.2.Plan de trabajo	<i>21</i>

3.Descripción informática

3.1.Especificación de requisitos.....	<i>22</i>
3.1.1. Requisitos funcionales	<i>22</i>
3.1.2. Requisitos no funcionales	<i>36</i>
3.2.Análisis	<i>38</i>
3.2.1. Requisitos de administrador	<i>39</i>
3.2.2. Requisitos de usuario	<i>63</i>
3.3.Diseño	<i>67</i>
3.3.1. Arquitectura	<i>67</i>
3.3.2. Diagramas de clases y paquetes	<i>69</i>
3.4.Implementación	<i>83</i>
3.4.1. Funcionalidad	<i>83</i>
3.4.2. Tecnologías	<i>84</i>
3.4.3. Interfaz gráfica	<i>86</i>
3.5.Pruebas	<i>92</i>

4. Conclusiones y mejoras

4.1. Conclusión **98**

4.2. Mejoras **98**

Referencias

101

Índice de figuras

<i>1. Resultados votación mayoría simple</i>	<i>12</i>
<i>2. Resultados votación mayoría cualificada</i>	<i>13</i>
<i>3. Resultados votación acumulativa</i>	<i>13</i>
<i>4. Resultados votación sistema ponderado</i>	<i>13</i>
<i>5. Resultados votación sistema preferencial</i>	<i>14</i>
<i>6. Resultados votación sistema votación aprobación</i>	<i>14</i>
<i>7. Diagrama de Gantt</i>	<i>21</i>
<i>8. Diagrama casos de uso de usuario</i>	<i>38</i>
<i>9. Diagrama casos de uso de administrador</i>	<i>38</i>
<i>10. Modelo - Vista - Controlador</i>	<i>69</i>
<i>11. Diagrama de paquetes</i>	<i>71</i>
<i>12. Diagrama de clases de eventos</i>	<i>73</i>
<i>13. Diagrama de clases de usuarios</i>	<i>75</i>
<i>14. Diagrama de clases de roles</i>	<i>76</i>
<i>15. Diagrama de clases de votación</i>	<i>77</i>
<i>16. Diagrama de clases de resultados</i>	<i>77</i>
<i>17. Diagrama de clases de login</i>	<i>78</i>
<i>18. Diagrama de base de datos</i>	<i>79</i>
<i>19. Diagrama de interfaz</i>	<i>81</i>
<i>20. Interfaz gráfica de login</i>	<i>86</i>
<i>21. Interfaz gráfica de index</i>	<i>86</i>
<i>22. Interfaz gráfica de admin</i>	<i>87</i>
<i>23. Interfaz gráfica de adminEventos</i>	<i>87</i>
<i>24. Interfaz gráfica de edicionEvento</i>	<i>87</i>
<i>25. Interfaz gráfica de asignarUsuarios</i>	<i>88</i>
<i>26. Interfaz gráfica de adminUsuarios</i>	<i>88</i>
<i>27. Interfaz gráfica de edicionUsuario</i>	<i>88</i>

28. Interfaz gráfica de adminRoles	89
29. Interfaz gráfica de edicionRol	89
30. Interfaz gráfica de listaResultados	90
31. Interfaz gráfica de resultado	90
32. Interfaz gráfica de listaEventos	91
33. Interfaz gráfica de votar	91
34. Interfaz gráfica de noAdmin	91
35. Prueba no válida: login y/o password	92
36. Prueba no válida: editando roles 1	92
37. Prueba no válida: editando roles 2	93
38. Prueba no válida: editando roles 3	93
39. Prueba no válida: editando usuarios 1	94
40. Prueba no válida: editando usuarios 2	94
41. Prueba no válida: editando usuarios 3	95
42. Prueba no válida: editando eventos 1	95
43. Prueba no válida: editando eventos 2	96
44. Prueba no válida: viendo resultado	96
45. Prueba no válida: votando 1	97
46. Prueba no válida: votando 2	97
47. Prueba no válida: votando 3	97

1. Introducción

La votación electrónica por medio de Internet está cogiendo, cada vez, un mayor protagonismo. Internet genera la posibilidad de expandir los mundos sociales, eliminando toda distancia geográfica, implicando al usuario. Hoy en día, no importa en qué lugar te encuentres, o en qué lugar tengas tu casa, si formas parte de una comunidad, empresa o grupo y hay que tomar decisiones todas ellas estarán al alcance de la mano, a través de un ordenador conectado a Internet ya que supone una vía de contacto que saca al hombre del aislamiento y lo hace sentir parte de un universo, cada vez más globalizado y ahora que ha derribado las fronteras nacionales y ha hecho del mundo una aldea global, es el momento de sacarle el mayor rendimiento a la votación electrónica.

En la actualidad, los acontecimientos relevantes se encuentran al alcance de la mano y no es necesario, tan siquiera, encontrarse en presencia física para poder dar una opinión. Para ello, el usuario interactúa con el equipo, frente a la pantalla de su PC, tratando de buscar información necesaria a su requerimiento.

Cabe destacar que hoy en día, realizar una votación a través de Internet para elegir un gobierno representa un fuerte respaldo a la tecnología. Sin ir más lejos, este tipo de acontecimiento se ha empleado de manera oficial en Estonia. Este país se ha convertido en la pionera mundial del sufragio por Internet. Los ciudadanos sólo necesitan un ordenador dotado de un lector electrónico de datos que pueda leer el documento de identidad, una firma electrónica y una contraseña. En un futuro cercano será una práctica habitual en todos los países desarrollados.

Llegados a este punto, comenzaremos a tratar los aspectos fundamentales del voto electrónico y sus aplicaciones, así como algunos de los sistemas o reglas de votación más extendidos en la actualidad. También describiremos algunos sistemas comerciales de voto electrónico que circulan por Internet.

1.1.- Voto electrónico

Hoy en día, la mayoría de procesos electorales en el sector público, involucran tecnología informática en la fase de recuento de los votos, para el escrutinio provisional y la difusión de datos. Esta informatización es la que permite que podamos disfrutar de los resultados finales de las elecciones pocas horas después del cierre de los colegios electorales. Pero en tiempos recientes, se ha ido más allá y se ha intentado automatizar, el acto de votación: los avances tecnológicos están favoreciendo que se introduzca la posibilidad de un cambio de paradigma, el uso de tecnología en la propia captación de los votos para los acontecimientos.

Cuando hablamos de voto electrónico [1] debemos saber que existen dos grandes grupos: presenciales y remotos. En los primeros, el votante debe desplazarse a un recinto electoral tradicional. Son los sistemas también denominados sistemas electrónicos de registro digital, basados en pantallas táctiles, teclados... En los sistemas remotos, se utiliza una red de comunicaciones, típicamente Internet, para transportar el voto desde el lugar donde se encuentra el votante, hasta un servidor que lo almacena.

Por tanto, el voto electrónico remoto, es el que se emite a través de un dispositivo que no está controlado por el organizador del evento. En este caso, hablamos también de voto online o a través de Internet. Claramente, una diferencia importante entre el voto tradicional y el voto electrónico está en que este último, gracias al uso de una infraestructura de telecomunicaciones e informática, permitiría informar y trasladar conocimiento al ciudadano acerca de todos los asuntos relacionados con la votación a realizar. Así, se verían potencialmente mejoradas las capacidades de los ciudadanos, que podrían reflejar sus preferencias a través del voto, mucho mejor informados.

Para considerarse realmente una herramienta de modernización del servicio público, el voto electrónico debe, como mínimo, respetar las exigencias básicas que se hacen a un sistema de voto, que deberían constituir el punto de partida para el diseño del sistema. Las agrupamos en cuatro categorías principales:

1. Registro de votantes:

Un sistema de voto necesita tener asociado una base de datos, o registro de electores, que garantice que las personas que tengan legalmente derecho a votar puedan hacerlo y que lo hagan una sola vez.

2. Autenticación de votantes:

En un sistema de voto, el votante tiene que poder identificarse como tal. En el caso del voto electrónico, la autenticación podría realizarse con la ayuda de uno o más mecanismos como una contraseña, un código PIN (Personal Identification Number), una firma electrónica, una tarjeta electrónica que contenga datos personales, datos biométricos...

3. Confidencialidad del voto:

La transmisión del voto desde la ubicación del votante hasta la urna debe estar protegida, no teniendo que ser posible ninguna modificación de los datos enviados.

4. Comprobación de los resultados:

Todos los votos tienen que ser contabilizados y asignados correctamente, asegurándose el recuento correcto de los votos.

También se han de tener en cuenta algunos requisitos indispensables para que los procesos de votación se desarrollen sin incidencias, por ejemplo:

1. La disponibilidad de los sistemas de captación de votos, cuando sean necesarios, en lo que se refiere a protección frente a los denominados ataques de denegación de servicio.
2. La garantía de la privacidad de los votantes, aún cuando éstos deben identificarse adecuadamente.
3. La protección de los votos digitales en cuanto a su manipulación, eliminación o la adición de votos falsos.
4. La verificabilidad respecto al tratamiento de los votos, una vez los resultados son publicados.

Por último, vamos a explicar las principales ventajas, y algunos inconvenientes, del voto electrónico respecto al tradicional:

1. *Mejora de la accesibilidad física.* El voto electrónico permite a los ciudadanos no tener que aguardar colas en sus colegios electorales, o en cualquier acontecimiento, pudiendo votar desde cualquier acceso a Internet en su casa, su trabajo... De igual

forma, facilitaría el acceso a las urnas a personas con dificultades físicas, pudiendo también de esta forma permitir contar los votos de ausentes en tiempo real. Ahora bien, la desigualdad en el acceso a los medios tecnológicos podría provocar que se favoreciese a aquéllos con medios para adquirir un ordenador y acceso a Internet. En el voto electrónico presencial, este hecho también se podría dar, pero más mitigado, ya que el uso de un sistema electrónico de registro digital podría hacer que los ciudadanos reacios a la tecnología se mostrasen menos predispuestos a participar, pero en caso alguno daría lugar a que los favorecidos tecnológicamente ganasen en influencia.

2. *Transformación de la interacción.* El uso del voto electrónico transformaría de forma sustancial las interacciones de los ciudadanos con sus representantes electos, con el consiguiente impacto psicológico.
3. *Nuevos medios de participación.* El uso de Internet ofrece nuevas alternativas para la distribución de ideas, la movilización de ciudadanos y la comunicación de opiniones sobre los temas de interés. De esta forma, los ciudadanos podrían obtener más capacidad de influencia, mediante la creación de redes y organizaciones de interés. Este hecho resulta de gran importancia, ya que haría que los ciudadanos pudieran tomar el pulso a los candidatos de una forma mucho más directa.
4. *Acceso universal y equitativo.* El uso de Internet será capaz de garantizar un acceso universal y equitativo a las urnas.
5. *Escalabilidad.* Mientras que en el voto ordinario, este aspecto viene dado por las esperas en los colegios electorales, en la votación electrónica hay que tener en cuenta la capacidad que tienen que tener los equipos para comportarse de manera adecuada frente a un gran número de usuarios.
6. *Fiabilidad.* Los equipos de los usuarios finales pueden no estar suficientemente preparados desde el punto de vista de la seguridad y la implantación, configuración y utilización de las herramientas del sistema; análogamente, la calidad de la conexión puede ser determinante a la hora de diseñar el interfaz de acceso.

1.2.- Sistemas de votación

Un sistema de votación [2] es una forma de elegir entre varias opciones basado en la decisión de los votantes. El acto de votar [3] tiene su uso más conocido en las elecciones, en las que se elige de entre los candidatos a cargos públicos. Las votaciones pueden emplearse también para conceder premios, elegir entre diferentes planes de

acción... En éstas, un individuo expresa apoyo o preferencia por cierto candidato, o selección de candidatos, durante una votación, de forma secreta o pública. Es, por tanto, "un método de toma de decisiones en el que un grupo, tal como una junta o un electorado, trata de medir su opinión conjunta".

Cabe destacar que los sistemas de votación contienen reglas para que la votación sea válida y especifican la manera de añadir los votos de los usuarios a los candidatos, dependiendo del tipo de sistema. El sistema de votación más popular quizás sea el de mayoría. Si cada elección tuviera solamente dos opciones, determinarían al ganador usando, únicamente, la regla de la mayoría simple. Sin embargo, cuando hay tres o más opciones, es posible que pueda no haber una sola opción que es preferida por una mayoría. Por ello, existen diversos sistemas de votación que pueden dar resultados muy diferentes, como podemos comprobar a continuación.

1.- Mayoría simple: en este sistema de votación, cada ciudadano da un voto, y sólo uno, a una alternativa. La alternativa ganadora es aquella que obtiene el mayor número de votos. En teoría, una opción puede ser elegida con solo dos votos, si las otras alternativas solo consiguieron uno.

Ejemplo: Seis votantes ($D_1...D_6$) y tres alternativas (A_1, A_2, A_3) donde los votantes darán un voto como máximo:

	D_1	D_2	D_3	D_4	D_5	D_6	VOTOS
A_1	X	X				X	3(Ganadora)
A_2			X				1
A_3				X	X		2

Figura 1: Resultados votación mayoría simple

2.- Mayoría cualificada: en este sistema, al igual que ocurre en el sistema de mayoría simple, cada ciudadano da un voto, y sólo uno, a una alternativa. La alternativa ganadora es aquella que obtiene un número de votos superior a un determinado porcentaje: dos tercios, cuatro quintos, cinco séptimos... La **mayoría absoluta** es un caso particular de sistemas de mayoría cualificada, donde una alternativa necesita más de la mitad de los votos para resultar ganadora.

Ejemplo: Seis votantes ($D_1...D_6$) y tres alternativas (A_1, A_2, A_3) donde los votantes darán un voto como máximo. El porcentaje mínimo es de tres quintos:

	D_1	D_2	D_3	D_4	D_5	D_6	VOTOS
A_1	X	X			X	X	4(Ganadora)
A_2			X				1
A_3				X	X		2

Figura 2: Resultados votación mayoría cualificada

3.- Voto acumulativo: en este sistema, a cada votante se le asigna un número de votos a distribuir entre las alternativas. Los votantes pueden otorgar más de un voto a una misma alternativa. La alternativa ganadora, será aquella que obtenga mayor número de votos, sin necesidad de obtener mayoría absoluta.

Ejemplo: Seis votantes ($D_1...D_6$) y tres alternativas (A_1, A_2, A_3) donde los votantes disponen de diez votos en total:

	D_1	D_2	D_3	D_4	D_5	D_6	VOTOS
A_1	2	5	4	8	3	5	27(Ganadora)
A_2	5	1	2	1	4	4	17
A_3	3	4	4	1	3	1	16

Figura 3: Resultados votación acumulativa

4.- Sistemas ponderados: en este sistema, cada participante en la votación, tiene un número de votos a repartir entre las alternativas, pero a diferencia de lo que ocurre en el voto acumulativo, con este sistema cada uno de los participantes puede tener un número distinto de votos a repartir. La alternativa ganadora será la que obtenga el mayor número de votos. Estos sistemas son muy útiles para empresas, donde dicho número de votos se mide en función de la participación, en forma de acciones, del votante en la empresa.

Ejemplo: Tres votantes ($D_1...D_3$) donde D_1 tiene rol de "Gerente", con diez votos disponibles. D_2 posee el rol de "Jefe de departamento", con cinco votos disponibles y, por último, D_3 que tiene el rol de "Becario", con 2 votos. En el evento hay dos alternativas (A_1, A_2) disponibles:

	D_1	D_2	D_3	VOTOS
A_1	3	3	1	7
A_2	7	2	1	10(Ganadora)

Figura 4: Resultados votación sistema ponderado

5.- Voto preferencial: en este sistema, los votantes, en lugar de indicar simplemente su opción favorita, colocan sus opciones por orden de preferencia, considerando primero su opción preferida, luego la opción siguiente, luego la tercera... y así sucesivamente. De esta manera, el sistema permite que los votantes expresen sus preferencias, entre las opciones dadas, en lugar de simplemente expresar su primera elección. A la hora de contar los votos se sigue el siguiente esquema, siendo n el número de alternativas:

- n-1 votos a la alternativa más preferida.
-
- 0 a la alternativa menos preferida.

Ejemplo: Tres votantes ($D_1...D_3$) y tres alternativas (A_1, A_2, A_3) disponibles. En la tabla se mostrará del uno al tres el orden de preferencia, siendo el 1 la alternativa que más preferencia tiene y 3 la que menos. Por tanto, al tener tres alternativas, la preferencia 1 valdrá 2 votos, la preferencia 2 valdrá un voto y la preferencia 3 obtendrá 0 votos.

	D_1	D_2	D_3	VOTOS
A_1	1	2	3	$1*3 + 2*1 + 3*0 = 5$
A_2	3	1	1	$3*0 + 1*3 + 1*3 = 6(\text{Ganadora})$
A_3	2	3	2	$2*1 + 3*0 + 2*1 = 4$

Figura 5: Resultados votación sistema preferencial

6.- Voto por aprobación: el último tipo de sistema consiste en que el votante puede votar todas las alternativas que considere aceptables. Al final de la votación, la alternativa ganadora será aquella que más votos haya obtenido.

Ejemplo: Seis votantes ($D_1...D_6$) y tres alternativas (A_1, A_2, A_3) donde los votantes darán un voto (y solo uno) a cada alternativa que estime oportuna:

	D_1	D_2	D_3	D_4	D_5	D_6	VOTOS
A_1	X	X	X		X	X	5(Ganadora)
A_2	X		X	X		X	1
A_3		X		X	X	X	2

Figura 6: Resultados sistema votación aprobación

1.3.- Sistemas comerciales de voto electrónico

El uso de software libre proporciona el medio ideal para poder realizar los procesos de verificabilidad: si el votante tiene acceso al código fuente y tiene, a su vez, la certificación de que dicho software es efectivamente el que se está ejecutando en el servidor, los demás problemas son obviados, pues es directamente verificable en el código que éste hace lo que dice, y no otra cosa.

Explicaremos diversos sistemas de software que existen en la actualidad para el voto electrónico:

1. Scytl:

El nombre viene de Scytale,[4] "el primer instrumento criptográfico de la historia, un sistema para codificar mensajes que se utilizaba en la Antigua Grecia hace 2.500 años.". Scytl es una empresa de software especializada en el desarrollo y comercialización de soluciones de modernización electoral. Estas soluciones incorporan protocolos criptográficos avanzados que permiten llevar a cabo, por medios electrónicos, cualquier tipo de elecciones o consultas de forma totalmente segura y auditable.

Scytl se formó a partir de un *spin-off* de un prestigioso grupo de investigación de la Universidad Autónoma de Barcelona. Este grupo, financiado por el Ministerio de Ciencia y Tecnología de España, fue pionero en la investigación sobre seguridad para el voto electrónico en Europa desde 1994. Como curiosidad, en 2007, el condado de Okaloosa (Florida) implementó la tecnología de voto electrónico seguro de Scytl para mejorar, notablemente, los procesos de votación para los votantes en el exterior. Esta empresa instaló varios kioscos de votación en Inglaterra, Alemania y Japón para permitir que los ciudadanos que vivían fuera de EEUU pudieran votar electrónicamente durante varios días. El proceso fue un éxito.

2. M-Vote Tempus [5]:

Es un ejemplo de una aplicación que permite instalar todo un sistema de votación a través de mensajes de texto, en los teléfonos. Estas son sus algunas de sus características:

1. Creación de servicios de eventos (concursos) sobre SMS Premium.
2. Selección de tarifa de los mensajes SMS para el concursante, de entre los contratados por el cliente.
3. Una vez dado de alta, los usuarios autorizados del cliente pueden configurar sus propios concursos.

4. Control de resultados en tiempo real.
5. Implementación inmediata.
6. Fácil de usar.

Además tiene diversas aplicaciones, por ejemplo:

1. Concurso Simple. Comunicación de un reto que el usuario responde identificando la posible respuesta A, B o C.
2. Porras.
3. Envío de códigos de producto, pruebas de compra.
4. Adivinar slogan de un producto, palabra oculta en una frase...
5. Puzzle. Cada unidad del producto lleva un código que, al enviarlo por SMS, le devuelve al usuario una letra para completar la palabra.
6. Publicidad en mensajes de consulta.

Para más información de la funcionalidad del sistema, se encuentra disponible a través de Internet un extenso manual:

http://www.tempos21.com/web/files/soporte/Manual_Usuario_M-Vote.pdf

3. *E-Vote* [6]:

Es una empresa alemana que ofrece diversos servicios de toma de decisiones distribuida. Para ello, alquila una aplicación que actúa como cliente de sistemas Windows y que permite conectarse a un servidor de toma de decisiones. Se utilizan mecanismos especiales para identificar unívocamente a los usuarios y garantizar la unicidad y secreto del voto, así como para cumplir la reglamentación europea sobre tratamiento de datos personales.

La aplicación tiene todas las características que se esperan de un sistema de voto:

1. Sistema de registro de usuarios.
2. Elaboración de consultas y votaciones.
3. Foro de discusiones.
4. Mecanismos de chat y videoconferencia en línea.
5. Planificadores de actividades.

4. MyPools [7]:

Es un sistema más simple que los anteriores, sin posibilidad de reconfiguración. Fue desarrollado por Josh Levine, basado en PHP y MySQL. En este programa, el administrador edita un fichero de configuración, donde se define la consulta y las opciones. Mediante cookies, se guardan las votaciones y sus resultados. Está orientado a una única votación sobre un tema concreto. El administrador es quien define los datos de la encuesta, teniendo que proceder manualmente a la mayor parte de las operaciones.

5. BVE [8]:

La BVE o Boleta de Voto Electrónico desarrollada por MSA es el elemento que fusiona en una única pieza de estructura física y lógica, las virtudes y fortalezas de la tecnología. Esta solución se implementa con un chip de radio frecuencia como los usados en las tarjetas de controles de acceso. Más concretamente, se trata de un documento que llevará impresa la opción electoral del votante una vez que haya votado, tanto en texto claro como en formato digital en la memoria del chip incorporado a la misma. A la hora de emitir el voto, cada votante deberá corroborar la total coincidencia entre lo que se ha impreso en el papel de la BVE y lo que electrónicamente se ha registrado en el chip de RF (radio frecuencia). El sistema asegura que la lectura y suma de cada chip coincida con el conjunto de BVE® que hay en una urna, o en un conjunto de urnas, posibilitando la obtención de un "escrutinio automático y electrónico".

2. Objetivos

El PFC, Proyecto de Fin de Carrera, constituye el colofón a los estudios de Ingeniería Informática y como tal, presenta como objetivo principal enfrentar al alumno al desarrollo de un proyecto informático, cuya finalidad sea el aprendizaje, que resuelva un problema concreto, dando lugar al análisis, diseño, evaluación o desarrollo total o parcial de sistemas software y/o hardware en relación al problema informático de partida.

La finalidad del proyecto va más allá de la resolución del problema en cuestión, siendo primordial el aprendizaje, por parte del alumno, de las técnicas y procedimientos empleados en la elaboración del proyecto, dando lugar, en última instancia, a sistemas informáticos que resuelvan la dificultad descrita. Es de rigor, no obstante, enfocar los objetivos del proyecto en el problema a resolver.

Después de haber realizado una introducción, en líneas generales, sobre el contexto en el que se engloba el proyecto, vamos a detallar algunos aspectos básicos acerca los objetivos de esta aplicación. Se plantea la elaboración de un proyecto que trate del desarrollo de una aplicación software sobre un sistema de votación electrónica, en el cual los usuarios, una vez registrados, podrán elegir sobre determinados acontecimientos y dar su voto o votos, a los distintos candidatos del evento escogido. Además, el usuario podrá ver el resultado de la votación cuando ésta haya finalizado. Por último, también podrá ver aquellos acontecimientos en los que haya votado, o no, siempre y cuando hayan finalizado.

De manera paralela, el software viene preparado con la funcionalidad de administrador. Cuando un usuario tenga este rol, tendrá la capacidad de editar todas las opciones configurables de la aplicación. Esta persona tendrá privilegios para añadir o editar eventos, registrar usuarios para que puedan utilizar la aplicación y encargarse de la edición de roles, personalizados, a los usuarios que lo requieran. Es decir, un control absoluto de la aplicación. El rol de administrador no estará autorizado para modificar el proceso de votación cuando este haya comenzado, ni para ejercer el derecho a voto.

2.1.- Metodología

El objetivo de la metodología es mejorar el proceso de desarrollo, definiendo los requisitos de forma acertada, estableciendo un método sistemático de desarrollo y construir un sistema fácil de mantener.

La Ingeniería del Software es una disciplina de la ingeniería que comprende todos los aspectos relacionados con la producción de software, desde las etapas iniciales, correspondientes a la especificación del sistema, hasta el mantenimiento de éste tras su utilización, es decir, el ciclo de vida software. Todo software posee un ciclo de vida, es decir, tiene un conjunto de actividades y resultados asociados que dan lugar a un producto software. Además este proceso está dirigido por un modelo de proceso, el cual determina el orden en el que se realizarán las tareas en el proyecto y provee de requisitos de entrada y salida a cada una de las actividades que lo sustentan. En definitiva, cada metodología se elige mediante la evaluación de las características que tiene el proyecto y se comparan con las que tiene la metodología. Una vez realizada esta comparación, para este proyecto, se ha decidido utilizar *Extreme Programming*.

La Programación Extrema [9] es la metodología más destacada de los procesos ágiles de desarrollo de software. Fue propuesta por Kent Beck en 1999 [10]. Es un enfoque basado en el desarrollo y la entrega de incrementos de funcionalidad muy limitada. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de esta metodología consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos, siendo capaces de adaptarse a los cambios de requisitos en cualquier punto de vida del proyecto para una mejor aproximación, más realista que definiendo los requisitos al comienzo del proyecto e invertir esfuerzos después de controlar los cambios de los requisitos. La programación extrema es considerada la adopción de las mejores metodologías de desarrollo de acuerdo a lo que pretende llevar a cabo con el proyecto y aplicarse, de forma dinámica, durante el ciclo de vida del software. El objetivo son grupos pequeños y medianos de construcción de software en donde los requisitos aún son muy ambiguos, cambian rápidamente o son de alto riesgo. Se busca la satisfacción del cliente tratando de mantener durante todo el tiempo su confianza en el producto.

A modo de síntesis, puesto que es un proyecto de complejidad media, para una única persona y donde, además, los requisitos pueden verse modificados, por los tutores del proyecto, esta metodología encaja perfectamente en la situación en la que se encuentra el proyecto. Además, puesto que solo hay un miembro, no hay divisiones en espacio y se tiene un control absoluto del proyecto, comprendiendo las decisiones y debatiendo, con los tutores, las posibles implementaciones, siendo así la comunicación con el cliente (tutores) fluida, lo que le hará conocedor de la aplicación como otro programador más.

2.2.- Plan de trabajo

A continuación se recoge el plan de trabajo que hemos seguido durante la elaboración del proyecto:

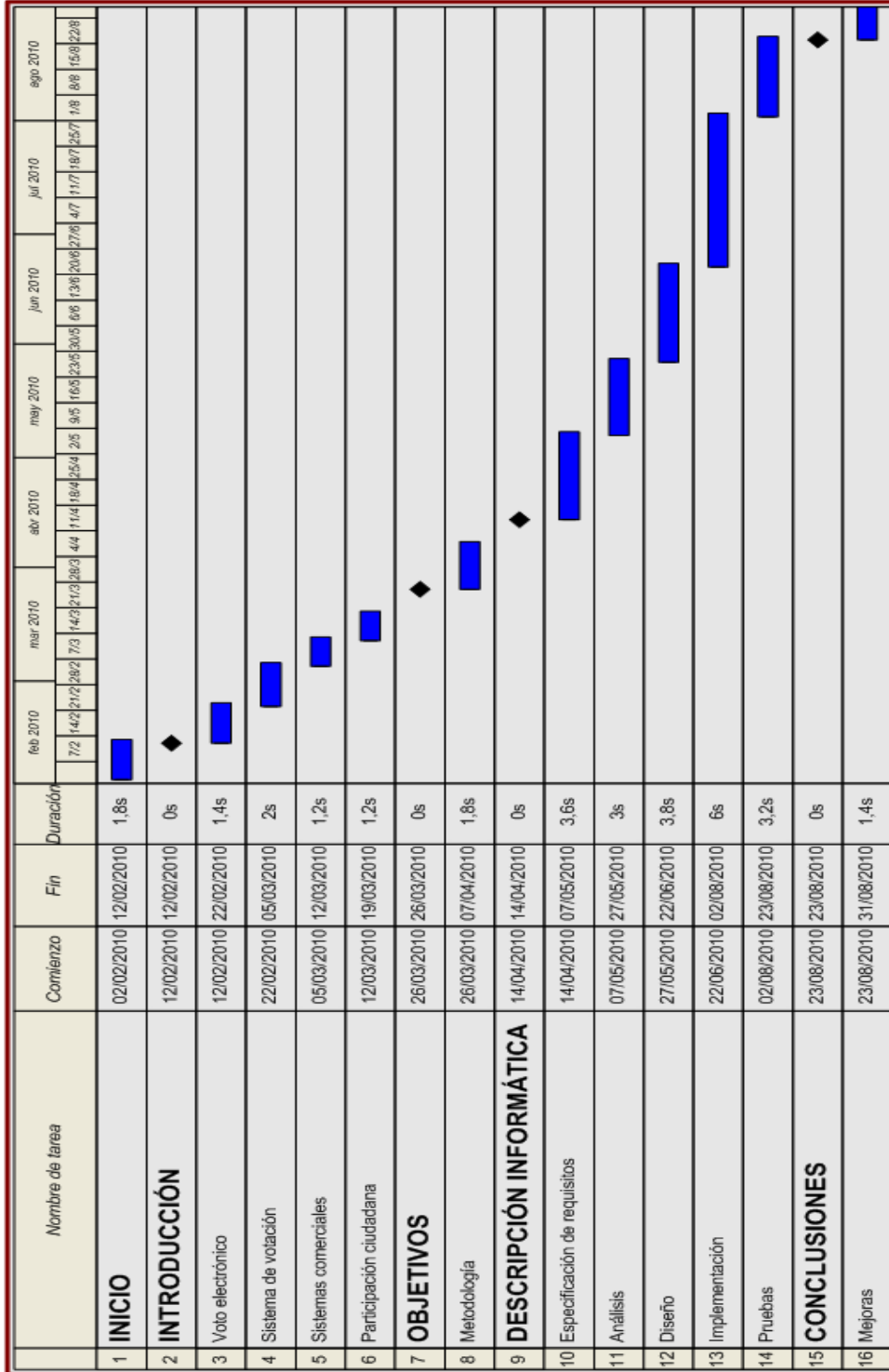


Figura 7: Diagrama de Gantt.

3. *Descripción informática*

Este apartado es esencial para comprender el software desarrollado siendo, por tanto, el pilar de este documento. Llegados a este punto, tenemos una visión general del medio en el que se desenvolverá el sistema, su estructura, configuración, etcétera, así como los objetivos que se pretenden alcanzar. Esta sección, servirá de nexo entre los apartados descritos anteriormente y el producto final.

3.1.- *Especificación de requisitos*

La identificación y especificación de requisitos es una de las tareas más importantes a la hora de desarrollar un sistema informático. Debe contener una lista detallada y completa de los requisitos que ha de cumplir el sistema a desarrollar. El nivel de detalle de éstos, debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen. Una buena especificación de requisitos nos evita el fracaso de proyectos y los elevados costes que puede conllevar.

Comenzaremos presentando los requisitos *funcionales* y *no funcionales*. Los requisitos funcionales describen lo que el sistema debe hacer, es decir, acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados. Los requisitos no funcionales no se refieren directamente a las funcionalidades que el sistema debe cumplir, sino a las propiedades emergentes de éste como: fiabilidad, rendimiento, usabilidad,...

Para mayor comprensión de esta sección, se ha optado por utilizar listas numeradas para la identificación, seguimiento, trazabilidad y validación de los requisitos.

Debido a que la aplicación a desarrollar tiene distintas categorías de usuarios (roles), los requisitos del sistema se dividirán en *requisitos de administrador* y *requisitos de usuario*.

3.1.1.- *Requisitos funcionales.*

3.1.1.1 *Requisitos de Administrador.*

RF1: Entrar en la aplicación.

Para iniciar la aplicación, el administrador deberá identificarse proporcionando su *usuario* y *contraseña*. En caso de que sus credenciales no sean correctas, se le

informará de manera detallada acerca del error y se le mostrará la pantalla de inicio para que pueda volverlo a intentar.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Entrada a la aplicación.

Proceso

1.- Login.

RF2: Salir de la aplicación.

El administrador podrá salir de la aplicación en cualquier momento.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Salida de la aplicación.

Proceso

1.- Login.

2.- Logout.

RF3: Alta de evento.

El administrador podrá crear un evento con las características que desee. Este evento será el que posteriormente pueda elegir el usuario para participar. El administrador deberá especificar el nombre del evento por el cual se reconozca el acontecimiento. Además irá acompañado de una pequeña descripción sobre el evento. También se deberá especificar su fecha de inicio y de finalización. Además, hay que asignar al evento un sistema de votación, a elegir entre las siguientes opciones: mayoría simple, mayoría cualificada, acumulativa, ponderada, preferencial y voto por aprobación. El evento también está compuesto de los candidatos y, por tanto, habrá que añadirlos (RF3.1). Por último, en el evento hay que especificar los usuarios que están autorizados a participar en éste (RF3.4).

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos del evento.

Salidas

- Nuevo evento.

Proceso

- 1.- Login.
- 2.- "Lista de eventos".
- 3.- "Añadir evento".
- 4.- Rellenar Datos.
- 5.- "Guardar".

RF3.1: Alta de candidato

Los candidatos son aquellas alternativas de las que disponen los usuarios a la hora de emitir su voto. El administrador podrá agregar tantos candidatos como desee, una vez que haya accedido a la creación de un nuevo evento. Para la creación de un nuevo candidato se tendrá que especificar su nombre, así como una pequeña descripción de éste.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos del candidato.

Salidas

- Nuevo candidato.

Proceso

- 1.- Login.
- 2.- "Lista de eventos".
- 3.- "Añadir evento".
- 4.- "Añadir candidato".
- 5.- "Guardar candidato".

RF3.2: Baja de candidato

Para que el administrador pueda dar de baja un candidato, primero tendrá que existir una lista ya creada de éstos. El administrador podrá eliminar tantos candidatos como desee.

Entradas

- Datos de Usuario (usuario y contraseña)

Salidas

- Eliminación de candidato.

Proceso

- 1.- Login
- 2.- "Lista de eventos"
- 3.- "Añadir evento"
- 4.- "Eliminar candidato"

RF3.3: Ver detalle

Este requisito permitirá al administrador conocer la descripción del candidato seleccionado.

Entradas

- Datos de Usuario (usuario y contraseña)

Salidas

- Ver descripción del candidato

Proceso

- 1.- Login
- 2.- "Lista de eventos"
- 3.- "Añadir evento"
- 4.- "Ver detalle"

RF3.4: Asignar usuario

Los usuarios autorizados a participar en el evento serán los que el administrador asigne en la ventana de asignación de usuarios. En ésta, se mostrará una lista de usuarios para que el administrador seleccione aquellos que considere oportuno. Puede seleccionarlos de uno en uno o bien seleccionar todos los que haya disponibles.

Entradas

- Datos de Usuario (usuario y contraseña)

Salidas

- Usuario/s asignado/s.

Proceso

- 1.- Login
- 2.- "Lista de eventos"
- 3.- "Añadir evento"
- 4.- "Asignar usuario"
- 5.- "Elección de usuarios"
- 6.- Guardar elección"

RF4: Modificación de evento.

El administrador podrá modificar un evento siempre y cuando no haya comenzado la votación. Además, tendrá la opción de editar los campos que necesite, así como añadir o eliminar participantes.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos del evento.

Salidas

- Modificación del evento.

Proceso

- 1.- Login.
- 2.- "Lista de eventos".
- 3.- "Seleccionar evento".
- 4.- Reescribir Datos.
- 5.- "Guardar".

RF4.1: Alta de candidato

Los candidatos son aquellas alternativas de las que disponen los usuarios a la hora de emitir su voto. El administrador podrá agregar tantos candidatos como desee, una vez que haya accedido a la edición de un evento ya existente. Para la creación de un nuevo candidato, se tendrá que especificar su nombre y una descripción de éste.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos del candidato.

Salidas

- Nuevo candidato.

Proceso

- 1.- Login.
- 2.- "Lista de eventos".
- 3.- "Seleccionar evento".
- 4.- "Añadir candidato".
- 5.- "Guardar candidato".

RF4.2: Baja de candidato

Para que el administrador pueda dar de baja un candidato, éste debe existir. El administrador elegirá al candidato a eliminar de una lista. Del mismo modo, podrá eliminar tantos candidatos como desee.

Entradas

- Datos de Usuario (usuario y contraseña)

Salidas

- Eliminación de candidato.

Proceso

- 1.- Login
- 2.- "Lista de eventos"
- 3.- "Seleccionar evento"
- 4.- "Eliminar candidato"

RF4.3: Ver detalle

Permitirá al administrador conocer la descripción del candidato seleccionado.

Entradas

- Datos de Usuario (usuario y contraseña)

Salidas

- Ver descripción del candidato

Proceso

- 1.- Login
- 2.- "Lista de eventos"
- 3.- "Seleccionar evento"
- 4.- "Ver detalle"

RF4.4: Asignar usuario

Los usuarios autorizados a participar en el evento serán los que el administrador asigne en la ventana de asignación de usuarios. En ésta, se mostrará una lista de usuarios para que el administrador seleccione aquellos que estime oportuno. Puede seleccionar de uno en uno, o bien seleccionar todos los que haya disponibles.

Entradas

- Datos de Usuario (usuario y contraseña)

Salidas

- Usuario/s asignado/s.

Proceso

- 1.- Login
- 2.- "Lista de eventos"
- 3.- "Seleccionar evento"
- 4.- "Asignar usuario"
- 5.- "Elección de usuarios"
- 6.- Guardar elección"

RF5: Baja de evento.

El administrador podrá eliminar un evento siempre y cuando éste no haya comenzado.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Eliminación evento.

Proceso

- 1.- Login.
- 2.- "Lista de eventos".
- 3.- "Eliminar evento".

RF6: Alta de usuario.

El administrador podrá crear usuarios para que puedan usar la aplicación. Este usuario podrá acceder a la aplicación y estará autorizado, de manera automática, a ver cualquier resultado de un acontecimiento que haya finalizado. Cabe destacar que al crear el usuario no se le autoriza a votar en los eventos. Para que esto sea posible, en los eventos habrá una opción para añadir a los usuarios que el administrador estime oportuno.

Para dar de alta un usuario será necesario especificar su nombre, la contraseña y el rol (RF6.1). El rol ha de estar creado previamente para que sea posible asignarlo al usuario.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos del usuario..

Salidas

- Nuevo Usuario.

Proceso

- 1.- Login.

- 2.- "Lista de usuarios".
- 3.- "Añadir usuario".
- 4.- Rellenar Datos.
- 5.- "Guardar usuario".

RF6.1: Asignar rol

Todo usuario que sea creado debe tener un rol asignado. No será posible, por tanto, un usuario sin rol específico.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos de usuario.

Salidas

- Rol asignado a un usuario.

Proceso

- 1.- Login
- 2.- " Lista de usuarios ".
- 3.- " Añadir usuario ".
- 4.- "Asignar rol".
- 5.- " Guardar usuario".

RF7: Modificación de usuario.

El administrador podrá modificar los campos del usuario que estime oportuno.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos del usuario..

Salidas

- Modificación del usuario.

Proceso

- 1.- Login.
- 2.- "Lista de usuarios".

- 3.- "Seleccionar usuario".
- 4.- Rellenar Datos.
- 5.- "Guardar usuario".

RF7.1: Asignar rol

Todo usuario que sea modificado deberá tener un rol asignado. Por defecto, el rol seleccionado será el que ya tenía antes de modificarlo.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos de usuario.

Salidas

- Rol asignado a un usuario.

Proceso

- 1.- Login
- 2.- " Lista de usuarios ".
- 3.- " Seleccionar usuario ".
- 4.- "Asignar rol".
- 5.- " Guardar usuario".

RF8: Baja de usuario.

El administrador podrá eliminar un usuario siempre y cuando no haya votado en ningún evento.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Eliminación del usuario.

Proceso

- 1.- Login.
- 2.- "Lista de usuarios".
- 3.- "Eliminar usuario".

RF9: Alta de rol.

El administrador podrá crear roles para que puedan ser asignados a los usuarios de la aplicación. Para dar de alta un rol será necesario especificar el nombre del rol y el número de votos que tendrá disponible ya que será necesario para el sistema de votación ponderado.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos de rol.

Salidas

- Creación de rol.

Proceso

- 1.- Login.
- 2.- "Lista de roles".
- 3.- "Añadir rol".
- 4.- Rellenar datos.
- 5.- "Guardar rol".

RF10: Modificación de rol.

El administrador podrá modificar el número de votos de un rol seleccionado. Así evitamos tener que eliminar un rol y crearlo de nuevo con el mismo nombre. No se podrá cambiar el nombre del rol puesto que en ese caso sería un rol distinto.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos de rol.

Salidas

- Modificación de rol.

Proceso

- 1.- Login.
- 2.- "Lista de roles".

- 3.- "Seleccionar rol".
- 4.- Rellenar datos.
- 5.- "Guardar rol".

RF11: Baja de rol.

El administrador podrá eliminar un rol si fuera necesario. No se podrá eliminar un rol que este asignado a algún usuario de la aplicación. Para realizar esto, habría que eliminar aquellos usuarios que tienen asignado el rol a eliminar.

Entradas

- Datos de Usuario (usuario y contraseña).
- Datos de rol.

Salidas

- Eliminación del rol.

Proceso

- 1.- Login.
- 2.- "Lista de roles".
- 3.- "Eliminar rol".

RF12: Ver resultados.

El administrador podrá ver el resultado de las votaciones una vez hayan acabado.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Resultado de la votación.

Proceso

- 1.- Login.
- 2.- "Ver resultados".
- 3.- "Selección de evento".

3.1.1.2.- Requisitos de Usuario.

RF13: Entrar en la aplicación.

Para iniciar la aplicación, los usuarios deberán identificarse proporcionando su *usuario* y *contraseña*. En caso de que sus credenciales no sean correctas, se les informará de manera detallada acerca del error y se le mostrará la pantalla de inicio para que puedan volverlo a intentar.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Entrada a la aplicación.

Proceso

1.- Login.

RF14: Salir de la aplicación.

Todos los usuarios tendrán acceso a salir de la sesión en cualquier momento.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Salida de la aplicación.

Proceso

1.- Login.

2.- Logout.

RF15: Votar.

Todos los usuarios tendrán acceso a votar en aquellos eventos en los que estén autorizados. Además de esta condición, no podrán votar en un evento que haya finalizado.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Votación.

Proceso

- 1.- Login.
- 2.- "Votaciones".
- 3.- "Selección de evento".
- 4.- "Votar".

RF15.1: Ver detalle.

Cuando un usuario tiene que elegir una alternativa, para emitir su voto, tiene la posibilidad de ver una pequeña descripción sobre cada uno de los candidatos.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Ver detalle.

Proceso

- 1.- Login.
- 2.- "Votaciones".
- 3.- "Selección de evento".
- 4.- "Ver detalle del candidato".

RF16: Ver resultados.

Todos los usuarios tendrán acceso a ver los resultados de aquellos eventos que hayan finalizado. Podrán tener acceso hayan votado, o no, en ese evento.

Entradas

- Datos de Usuario (usuario y contraseña).

Salidas

- Ver resultado.

Proceso

- 1.- Login.
- 2.- "Ver resultados".
- 3.- "Selección de evento".

3.1.2.- Requisitos no funcionales.

3.1.2.1.- Requisitos de rendimiento.

La carga que se espera que tenga que soportar el sistema será importante. Mediante un servidor potente, se conseguirá que el margen de operaciones no pierdan calidad y, además, que el crecimiento continuo de trabajo sea fluido. Además, la aplicación será eficiente en el uso de recursos, tanto computacionales, como de almacenamiento, lo que contribuirá al buen rendimiento de la aplicación.

3.1.2.2.- Requisitos de seguridad.

La aplicación garantizará la protección de acceso a información de otros usuarios. Además, los servicios proporcionados por el sistema y la información almacenada serán únicamente accesibles a los usuarios autorizados. Estará protegida de sabotajes, así como de destrucciones maliciosas o accidentales. Además se emplearán técnicas criptográficas para la creación de contraseñas.

3.1.2.3.- Requisitos de fiabilidad.

La aplicación deberá realizar la operación que se le ordene. En el sistema de votación es muy importante este aspecto ya que cuando un usuario emita su voto, éste debe ser registrado de manera correcta.

3.1.2.4.- Requisitos de mantenibilidad

La tarea de mantenimiento principal será hacer un seguimiento diario para evitar caídas de servidor. En principio, será una aplicación fácil de mantener y fácil de modificar, para futuras mejoras. El administrador será el encargado de realizar dicho mantenimiento.

3.1.2.5.- Requisitos de disponibilidad.

La aplicación web será accesible el 100% del tiempo, salvo durante las tareas de mantenimiento, para que cualquier usuario pueda usar la aplicación en todo momento.

3.1.2.6.- Requisitos de usabilidad.

La aplicación será amigable para el usuario y muy intuitiva a la hora de utilizarla. Será utilizada por todo tipo de usuarios y será fácil de usar para personas sin experiencia con las computadoras.

3.1.2.7.- Requisitos de portabilidad

La aplicación deberá funcionar en todos los navegadores independientemente del sistema operativo y de la calidad de conexión.

3.1.2.8.- Requisitos de estabilidad

La aplicación deberá mantenerse frente a comportamientos inesperados, logrando que se recupere ante cualquier excepción.

3.2.- Análisis

A continuación, presentaremos el diagrama de casos de uso de la aplicación:

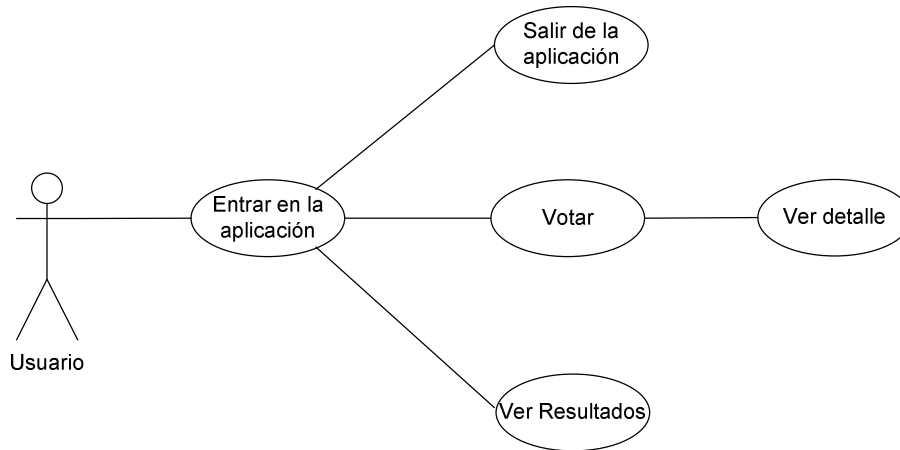


Figura 8: Diagrama de casos de uso de usuario

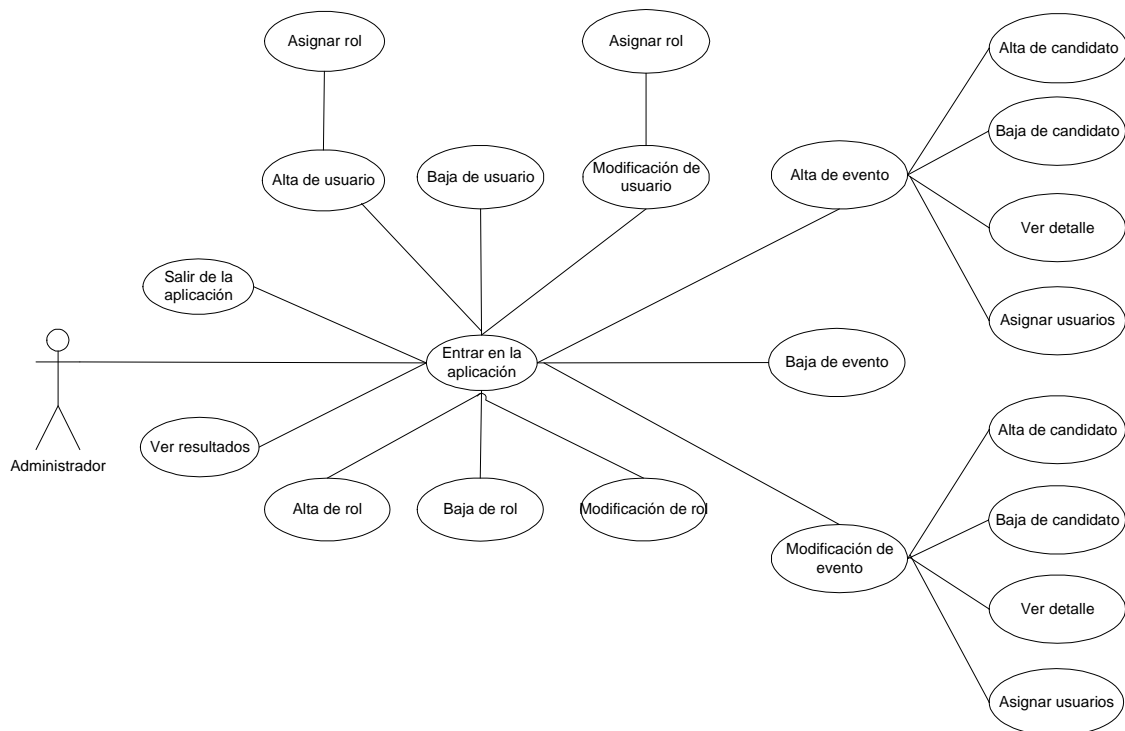


Figura 9: Diagrama de casos de uso de administrador

Por tanto, podemos afirmar que nuestro sistema tendrá dos actores: el usuario y el administrador. A continuación se especificarán los casos de uso de cada requisito:

3.2.1.- Requisitos de Administrador:

RF1: Entrar en la aplicación.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú.

- *Caminos alternativos:*

- Datos erróneos Login

<i>USUARIO</i>	<i>SISTEMA</i>
2. Introducir usuario y contraseña.	1. Solicitar datos login. 3. Comprobar datos login. 4. Mostrar mensaje de error. 5.- Volver a RF1.

RF2: Salir de la aplicación.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Logout.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Solicitar datos login.

RF3: Alta de evento.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Añadir nuevo evento. 11.- Rellenar datos. 12.- Guardar evento.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Nuevo evento. 13.- Almacenar datos. 14.- Lista de eventos.

- *Camino alternativo:*

- Datos erróneos al rellenar datos del evento.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Añadir nuevo evento.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de eventos.

11.- Rellenar datos erróneos. 12.- Guardar evento.	10.- Nuevo evento. 13.- Informar del tipo de error. 14.- Volver a RF3.
---	--

RF3.1: Alta de candidato.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Añadir nuevo evento. 11.- Añadir un candidato. 13.- Rellenar datos. 14.- Guardar candidato.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Nuevo evento. 12.- Nuevo candidato. 15.- Almacenar datos. 16.- Datos del evento.

- *Camino alternativo:*
 - Datos erróneos al rellenar datos del candidato.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de eventos.	4.- Mostrar Menú.
9.- Añadir nuevo evento.	6.- Mostrar menú administrador.
11.- Añadir un candidato.	8.- Lista de eventos.
12.- Rellenar datos erróneos.	10.- Nuevo evento.
13.- Guardar candidato.	12.- Nuevo candidato.
	14.- Informar del tipo de error.
	15.- Volver a RF3.

RF3.2: Baja de candidato.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
	4.- Mostrar menú.
	6.- Mostrar menú administrador.

7.- Edición de eventos.	8.- Lista de eventos.
9.- Añadir nuevo evento.	10.- Nuevo evento.
11.- Añadir un candidato.	12.- Nuevo candidato.
13.- Rellenar datos.	15.- Almacenar datos.
14.- Guardar candidato.	16.- Datos del evento.
17.- Eliminar candidato.	18.- Almacenar datos.

RF3.3: Ver detalle.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de eventos.	4.- Mostrar menú.
9.- Añadir nuevo evento.	6.- Mostrar menú administrador.
11.- Añadir un candidato.	8.- Lista de eventos.
13.- Rellenar datos.	10.- Nuevo evento.
	12.- Nuevo candidato

14.- Guardar candidato.	15.- Almacenar datos.
17.- Ver detalle.	16.- Datos del evento.
	18.- Mostrar detalle.

RF3.4: Asignar usuarios.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de eventos.	4.- Mostrar menú.
9.- Añadir nuevo evento.	6.- Mostrar menú administrador.
11.- Asignar usuarios.	8.- Lista de eventos.
13.- Seleccionar usuarios.	10.- Nuevo evento.
14.- Guardar selección.	12.- Lista de usuarios.
	15.- Almacenar datos.
	16.- Datos del evento.

- *Camino alternativo:*
 - Cancelar selección de usuarios.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Añadir nuevo evento. 11.- Asignar usuarios. 13.- Seleccionar usuarios. 14.- Cancelar selección.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Nuevo evento. 12.- Lista de usuarios. 16.- Datos del evento.

RF4: Modificación de evento.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador.

<p>9.- Seleccionar evento.</p> <p>11.- Sobrescribir datos.</p> <p>12.- Guardar evento.</p>	<p>8.- Lista de eventos.</p> <p>10.- Evento seleccionado.</p> <p>13.- Almacenar datos.</p> <p>14.- Lista de eventos.</p>
--	--

- *Camino alternativo:*

- Datos erróneos al sobrescribir datos del evento.

<i>USUARIO</i>	<i>SISTEMA</i>
<p>2.- Introducir usuario y contraseña.</p> <p>5.- Menú administrador.</p> <p>7.- Edición de eventos.</p> <p>9.- Seleccionar evento.</p> <p>11.- Sobrescribir datos erróneos.</p> <p>12.- Guardar evento.</p>	<p>1.- Solicitar datos login.</p> <p>3.- Comprobar datos login.</p> <p>4.- Mostrar menú.</p> <p>6.- Mostrar menú administrador.</p> <p>8.- Lista de eventos.</p> <p>10.- Evento seleccionado.</p> <p>13.- Informar del tipo de error.</p> <p>14.- Volver a RF4.</p>

RF3.1: Alta de candidato.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de eventos.	4.- Mostrar menú.
9.- Seleccionar Evento.	6.- Mostrar menú administrador.
11.- Añadir un candidato.	8.- Lista de eventos.
13.- Rellenar datos.	10.- Evento seleccionado.
14.- Guardar candidato.	12.- Nuevo candidato.
	15.- Almacenar datos.
	16.- Datos del evento.

- *Camino alternativo:*

- Datos erróneos al rellenar datos del candidato.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de eventos.	4.- Mostrar Menú.
	6.- Mostrar menú administrador.

<p>9.- Seleccionar evento.</p> <p>11.- Añadir un candidato.</p> <p>12.- Rellenar datos erróneos.</p> <p>13.- Guardar candidato.</p>	<p>8.- Lista de eventos.</p> <p>10.- Evento seleccionado.</p> <p>12.- Nuevo candidato.</p> <p>14.- Informar del tipo de error.</p> <p>15.- Volver a RF4.</p>
---	--

RF3.2: Baja de candidato.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
<p>2.- Introducir usuario y contraseña.</p> <p>5.- Menú administrador.</p> <p>7.- Edición de eventos.</p> <p>9.- Seleccionar evento.</p> <p>11.- Eliminar candidato.</p>	<p>1.- Solicitar datos login.</p> <p>3.- Comprobar datos login.</p> <p>4.- Mostrar menú.</p> <p>6.- Mostrar menú administrador.</p> <p>8.- Lista de eventos.</p> <p>10.- Evento seleccionado.</p> <p>12.- Almacenar datos.</p> <p>13.- Datos del evento.</p>

RF3.3: Ver detalle.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Seleccionar evento. 11.- Ver detalle del candidato.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Evento seleccionado.

RF3.4: Asignar usuarios.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Seleccionar evento. 11.- Asignar usuarios.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Evento seleccionado.

13.- Seleccionar usuarios. 14.- Guardar selección.	12.- Lista de usuarios. 15.- Almacenar datos. 16.- Datos del evento.
---	--

- *Camino alternativo:*

- Cancelar selección de usuarios.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Seleccionar evento. 11.- Asignar usuarios. 13.- Seleccionar usuarios. 14.- Cancelar selección.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Evento seleccionado. 12.- Lista de usuarios. 16.- Datos del evento.

RF5: Baja de evento.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Eliminar evento. 11.- Aceptar confirmación.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Confirmar eliminación. 12.- Almacenar datos. 13.- Lista de eventos.

- *Camino alternativo:*

- Cancelar baja de evento.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de eventos. 9.- Eliminar evento.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de eventos. 10.- Confirmar eliminación.

11.- Cancelar confirmación.	12.- Lista de eventos.
-----------------------------	------------------------

- o El evento ha comenzado.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de eventos.	4.- Mostrar Menú.
9.- Eliminar evento.	6.- Mostrar menú administrador.
11.- Aceptar confirmación.	8.- Lista de eventos.
	10.- Confirmar eliminación.
	12.- Mensaje de error.
	13.- Lista de eventos.

RF6: Alta de usuario.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de usuarios.	4.- Mostrar Menú.
	6.- Mostrar menú administrador.
	8.- Lista de usuarios.

9.- Añadir nuevo usuario.	
11.- Rellenar datos.	10.- Nuevo usuario.
12.- Guardar usuario.	13.- Almacenar datos.
	14.- Lista de usuarios.

- *Camino alternativo:*

- Datos erróneos al rellenar datos del usuario.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de usuarios.	4.- Mostrar menú.
9.- Añadir nuevo usuario.	6.- Mostrar menú administrador.
11.- Rellenar datos erróneos.	8.- Lista de usuarios.
12.- Guardar usuario.	10.- Nuevo usuario.
	13.- Informar del tipo de error.
	14.- Volver a RF6.

RF6.1: Asignar rol.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de usuarios. 9.- Añadir nuevo usuario. 11.- Asignar rol. 12.- Guardar usuario.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de usuarios. 10.- Nuevo usuario. 13.- Almacenar datos. 14.- Lista de usuarios.

RF7: Modificación de usuario.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de usuarios.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de usuarios.

9.- Seleccionar usuario.	10.- Usuario seleccionado.
11.- Sobrescribir datos.	
12.- Guardar usuario.	13.- Almacenar datos.
	14.- Lista de usuarios.

- *Camino alternativo:*

- Datos erróneos al sobrescribir datos del usuario.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
	3.- Comprobar datos login.
	4.- Mostrar menú.
5.- Menú administrador.	6.- Mostrar menú administrador.
7.- Edición de usuarios.	8.- Lista de usuarios.
9.- Seleccionar usuario.	10.- Usuario seleccionado.
11.- Sobrescribir datos erróneos.	
12.- Guardar usuario.	13.- Informar del tipo de error.
	14.- Volver a RF7.

RF7.1: Asignar rol.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de usuarios. 9.- Seleccionar usuario. 11.- Asignar rol. 12.- Guardar usuario.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de usuarios. 10.- Usuario seleccionado. 13.- Almacenar datos. 14.- Lista de usuarios.

RF8: Baja de usuario.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de usuarios.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de usuarios.

9.- Eliminar usuario. 11.- Aceptar eliminación.	10.- Confirmación de eliminación. 12.- Almacenar datos. 13.- Lista de usuarios.
--	---

- *Camino alternativo:*

- Cancelar baja de usuario.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de usuarios. 9.- Eliminar usuario. 11.- Cancelar eliminación.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de usuarios. 10.- Confirmar eliminación. 12.- Lista de usuarios.

- El usuario ya ha votado.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador.

7.- Edición de usuarios.	8.- Lista de usuarios.
9.- Eliminar usuario.	10.- Confirmación de eliminación.
11.- Aceptar eliminación.	12.- Mensaje de error.
	13.- Lista de usuarios.

RF9: Alta de rol.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de roles.	4.- Mostrar Menú.
9.- Añadir rol.	6.- Mostrar menú administrador.
11.- Rellenar datos.	8.- Lista de roles.
12.- Guardar rol.	10.- Nuevo rol.
	13.- Almacenar datos.
	14.- Lista de roles.

- *Camino alternativo:*
 - Datos erróneos al rellenar los datos del rol.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de roles. 9.- Añadir rol. 11.- Rellenar datos erróneos. 12.- Guardar rol.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Mostrar menú administrador. 8.- Lista de roles. 10.- Nuevo rol. 13.- Mensaje de error. 14.- Volver a RF9.

RF10: Modificación de rol.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de roles.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de roles.

9.- Seleccionar rol.	
11.- Sobrescribir datos.	10.- Rol seleccionado.
12.- Guardar rol.	13.- Almacenar datos.
	14.- Lista de roles.

- *Camino alternativo:*

- Datos erróneos al sobrescribir los datos del rol.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Menú administrador.	3.- Comprobar datos login.
7.- Edición de roles.	4.- Mostrar menú.
9.- Seleccionar rol.	6.- Mostrar menú administrador.
11.- Sobrescribir datos erróneos.	8.- Lista de roles.
12.,. Guardar rol.	10.- Rol seleccionado.
	13.- Mensaje de error.
	14.- Volver a RF9.

RF11: Baja de rol.

• *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de roles. 9.- Eliminar rol. 11.- Aceptar confirmación.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de roles. 10.- Confirmación de eliminación. 12.- Almacenar datos. 13.- Lista de roles.

• *Camino alternativo:*

- Cancelar confirmación de eliminación de rol.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de roles. 9.- Eliminar rol.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de roles. 10.- Confirmación de eliminación.

11.- Cancelar confirmación.	12.- Lista de roles.
-----------------------------	----------------------

- o Rol asignado a un usuario.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Menú administrador. 7.- Edición de roles. 9.- Eliminar rol. 11.- Aceptar confirmación.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Mostrar menú administrador. 8.- Lista de roles. 10.- Confirmación de eliminación. 12.- Mensaje de error. 13.- Lista de roles.

RF12: Ver resultados.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Ver resultados. 7.- Seleccionar evento.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de resultados. 8.- Mostrar resultado seleccionado.

- *Camino alternativo:*

- El evento aún no ha finalizado.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Ver resultados.	3.- Comprobar datos login.
7.- Seleccionar evento.	4.- Mostrar Menú.
	6.- Lista de resultados.
	8.- Mensaje de error.
	9.- Lista de resultados.

3.2.2.- *Requisitos de usuario:*

RF13: Entrar en la aplicación.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
	3.- Comprobar datos login.
	4.- Mostrar menú.

- *Caminos alternativos:*

- Datos erróneos Login

<i>USUARIO</i>	<i>SISTEMA</i>
2. Introducir usuario y contraseña.	1. Solicitar datos login.
	3. Comprobar datos login.
	4. Mostrar mensaje de error.
	5.- Volver a RF13.

RF14: Salir de la aplicación.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Logout.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar menú. 6.- Solicitar datos login.

RF15: Votar.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Votación. 7.- Seleccionar evento. 9.- Votar.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de eventos. 8.- Mostrar evento seleccionado. 10.- Almacenar datos. 12.- Mensaje de éxito. 12.- Lista de eventos.

- *Camino alternativo:*
 - El usuario ya ha votado anteriormente.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Votación. 7.- Seleccionar evento. 9.- Votar. 11.- Aceptar.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de eventos. 8.- Mostrar evento seleccionado. 10.- Confirmación para modificar voto. 12.- Eliminar votos anteriores. 13.- Almacenar nuevos datos. 14.- Mensaje de éxito. 15.- Lista de eventos.

- *Camino alternativo:*

- El evento ya ha finalizado.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Votación. 7.- Seleccionar evento.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de eventos. 8.- Mostrar mensaje error. 9.- Lista de eventos.

- Votación no válida.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Votación. 7.- Seleccionar evento. 9.- Votar de forma errónea.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de eventos. 8.- Mostrar evento seleccionado. 10.- Mensaje tipo error. 11.- Volver RF15.

RF15.1: Ver detalle.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Votación. 7.- Seleccionar evento. 11.- Ver detalle de un candidato.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de eventos. 9.- Mostrar evento seleccionado. 12.- Mostrar detalle de candidato.

RF16: Ver resultados.

- *Camino estándar:*

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña. 5.- Ver resultados. 7.- Seleccionar evento.	1.- Solicitar datos login. 3.- Comprobar datos login. 4.- Mostrar Menú. 6.- Lista de eventos. 8.- Mostrar evento seleccionado.

- *Camino alternativo:*
 - Evento aún no ha finalizado.

<i>USUARIO</i>	<i>SISTEMA</i>
2.- Introducir usuario y contraseña.	1.- Solicitar datos login.
5.- Ver resultados.	3.- Comprobar datos login.
7.- Seleccionar evento.	4.- Mostrar Menú.
	6.- Lista de eventos.
	8.- Mensaje de error.
	9.- Lista de eventos.

3.3.- Diseño

Una vez finalizado el análisis del software sigue la fase de diseño. Para poder desarrollar esta fase es necesario apoyarse en el análisis. Durante el diseño se podrán encontrar posibles errores que no se hayan identificado en la etapa anterior. En resumen, esta fase será como el manual de apoyo para poder realizar el sistema de la manera más adecuada.

Con este apartado conseguiremos crear una abstracción del proceso de implementación y realizaremos una entrada apropiada para esa fase. A continuación, vamos a determinar la arquitectura software de la aplicación, así como, diagramas de clases para visualizar globalmente y, desde un alto nivel de abstracción, el funcionamiento interno de la aplicación.

3.3.1.- Arquitectura

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema [11]. Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

La aplicación ha sido desarrollada utilizando Java Server Faces que es un Framework incluido dentro de Java EE que tiene como misión facilitar la construcción y mantenimiento de aplicaciones web en Java, siguiendo una arquitectura MVC (Modelo-Vista-Controlador) [12].

De manera gráfica, las capas serán distribuidas de la siguiente manera:

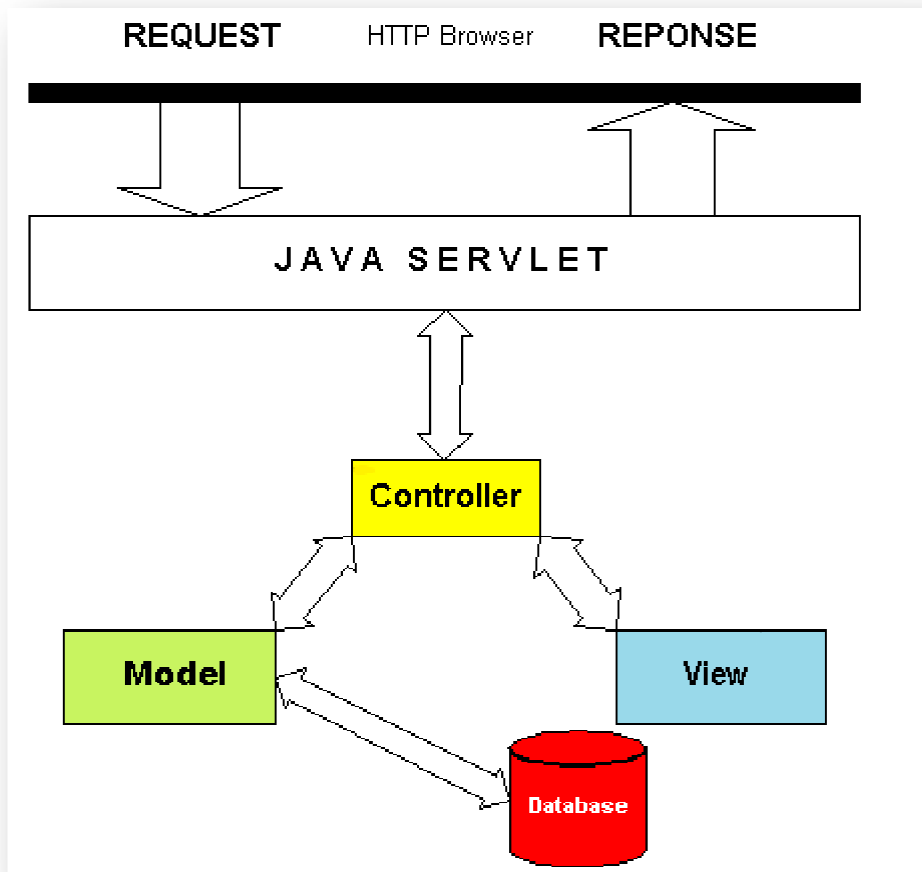


Figura 10: Modelo - Vista - Controlador.

- **Modelo:**

En esta capa disponemos las clases de acceso a los datos y aquellas que modelan la parte de la realidad sobre la que se desea actuar. El modelo es el objeto que representa y trabaja directamente con los datos del programa: los gestiona y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los diferentes controladores y/o vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas y, notificar a las vistas, cuándo deben reflejar un cambio en el modelo.

- *Vista:*

La vista es el objeto que maneja la presentación visual de los datos gestionados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interacciona con el modelo a través de una referencia al propio modelo. En esta aplicación, la vista está manipulada a través de las páginas JSF que conectan la vista y el modelo.

- *Controlador:*

El controlador es el objeto que proporciona significado a las órdenes del usuario actuando sobre los datos representados por el modelo. Entrará en acción cuando se realice alguna operación, ya sea un cambio en la información del modelo o una interacción sobre la vista. Se comunica con el modelo y la vista a través de una referencia al propio modelo. Además, JSF opera como un gestor que reacciona ante los eventos provocados por el usuario, procesa sus acciones y los valores de estos eventos, y ejecuta código para actualizar el modelo o la vista. El controlador que utiliza JSF es el servlet FacesServlet.

3.3.2.- Diagramas de clases y paquetes

Los diagramas de clases representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos [13].

Los diagramas de paquetes muestran cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Todos estos diagramas han sido incluidos puesto que favorecen la comprensión del proceso a través de mostrarlo de forma gráfica. Dado que un buen diagrama reemplaza varias páginas de texto. Además, permiten identificar los problemas y las oportunidades de mejora del proceso, así como la estructura interna de la aplicación.

3.3.2.1- Diagrama de paquetes:

En esta aplicación, los paquetes estarán organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes.

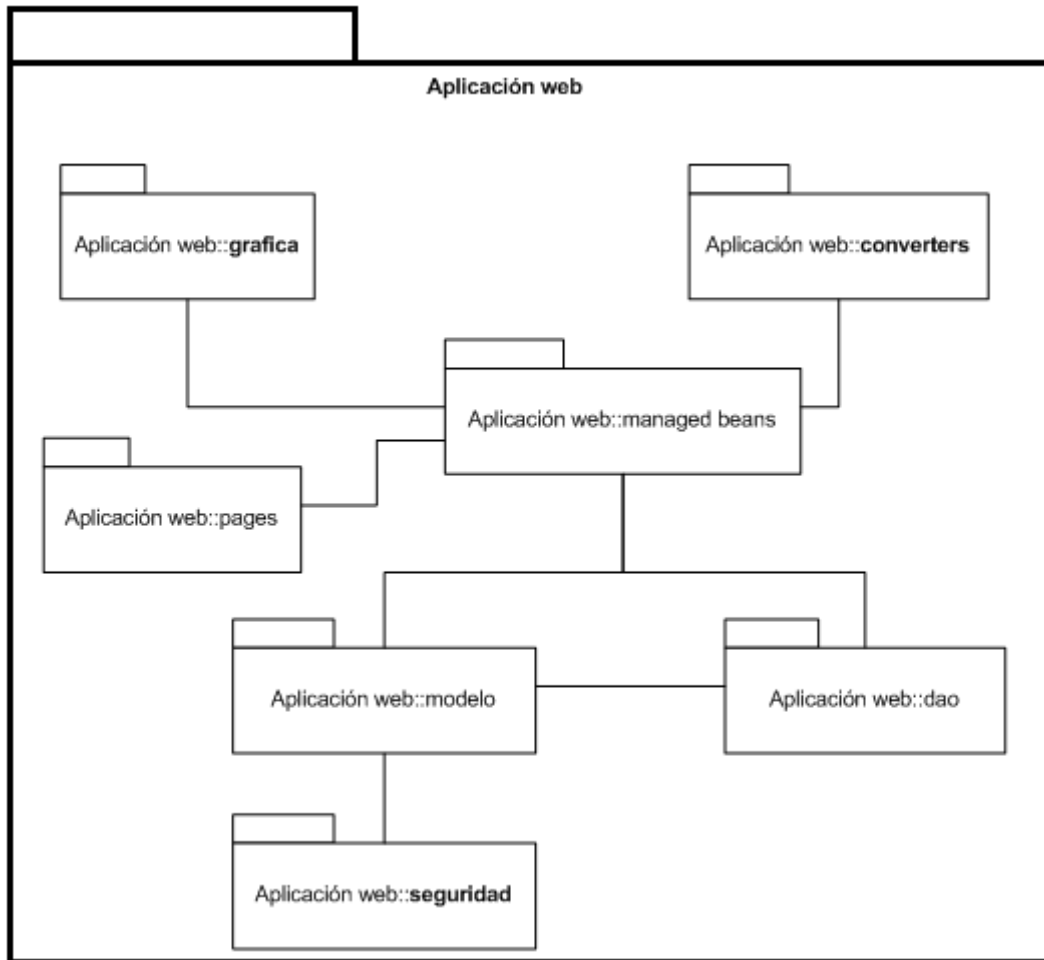


Figura 11: Diagrama de paquetes.

- *Aplicación web:*
Este es el paquete que engloba toda la aplicación.
- *Managed beans:*
Este paquete contiene a todos los objetos que realizan la función del controlador (controlar la interacción entre la interfaz de usuario y el modelo).
- *Gráfica:*
Este paquete contiene las clases necesarias para construir la gráfica de los resultados de un determinado evento.
- *Converters:*
Este paquete contendrá los convertidores que traducen y formatean el valor de los componentes para su visualización. También generarán mensajes de error.
- *Modelo:*
Este paquete contiene las clases que modelan la parte de la realidad sobre la que se desea actuar. Todo lo relativo a eventos, candidatos, usuarios y roles.

- *Seguridad:*

Este paquete contiene los métodos criptográficos que utilizaremos para aportar el aspecto de seguridad a la aplicación.

- *DAO:*

Este paquete contiene todo lo necesario a la gestión de base de datos. Aquí se incluye todo lo necesario para que el sistema interactúe con la base de datos mediante Hibernate y JPA con MySQL.

- *Pages:*

Este paquete incluye todo lo que es la interfaz de usuario, es decir, la parte de la vista de la aplicación. Está compuesto por páginas "jsp".

3.3.2.2.- *Diagrama de clases:*

Para visualizar las relaciones entre las clases de la aplicación, hemos construido varios diagramas de clases del sistema. En estos diagramas, se podrá observar todas las asociaciones entre clases que existen. Sin embargo, los diagramas que se muestran a continuación serán resumidos, es decir, no se especificarán los atributos propios de la clase ni los métodos. Se ha tomado esta decisión al considerar que es más legible y más claro un esquema limpio, sin mucha información. Además, explicaremos todas las clases implicadas en cada diagrama. Muchas clases de los diagramas estarán repetidas, por tanto, solo serán explicadas la primera vez.

Antes de comenzar a describir los diagramas, es importante conocer que un managed bean (MB) es un objeto java manejado por el controlador de JSF (FacesServlet) y asocia a cada vista con formularios que facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.

1. Eventos:

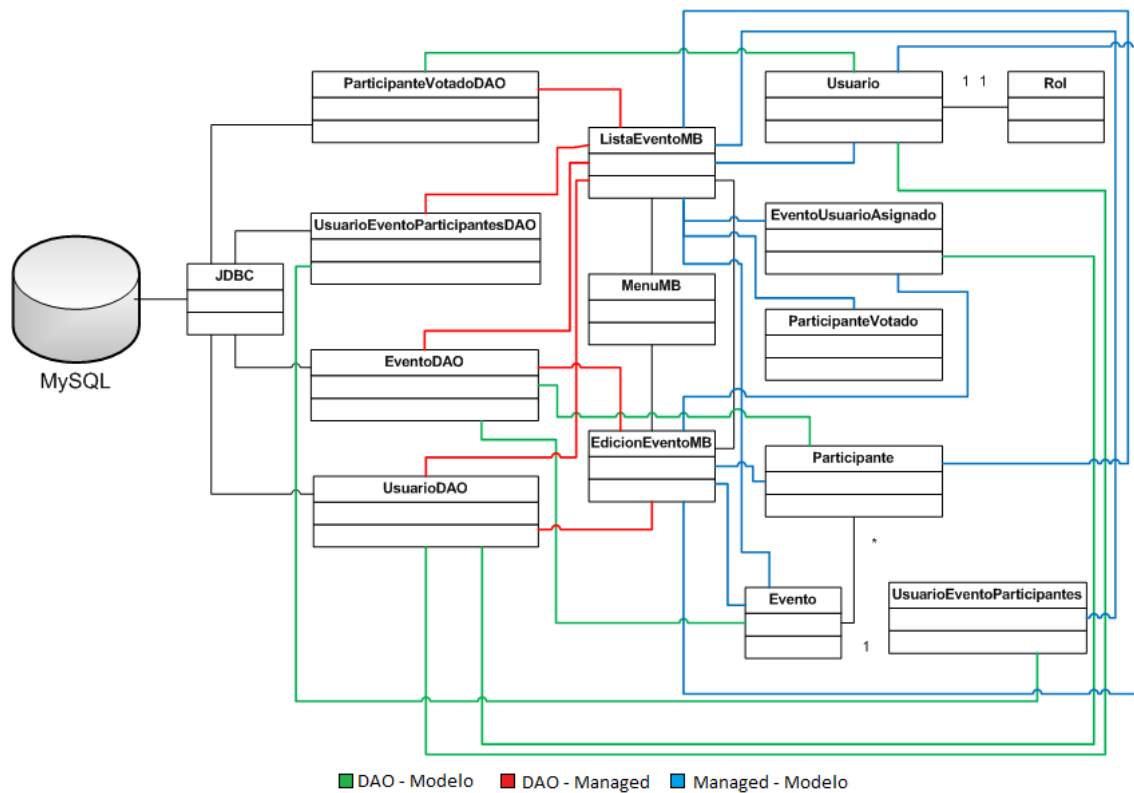


Figura 12: Diagrama de clases: eventos.

- **MenuMB:**

Se encarga de la gestión de memoria de los managed beans que se encuentran en la memoria del servidor. Al solicitar una página, el bean correspondiente entra en memoria, pero todos los managed anteriores se eliminan haciendo un uso eficiente de la memoria del servidor.

- **ListaEventoMB:**

Es el encargado de manejar la lista de los eventos disponibles registrados en el sistema. Gestionará la selección de un evento determinado o bien dar paso al siguiente managed bean para la creación de uno nuevo.

- **EdicionEventoMB:**

Se encarga de la creación de un nuevo evento y almacenarlo en la base de datos una vez rellenado los datos necesarios. También se encarga de la edición de un evento existente.

- **Usuario:**

Esta clase modela todo lo referente a un usuario de la aplicación. Aquí se especifica su nombre, contraseña y el rol que desempeña (ha de estar previamente creado).

- *Rol:*

Esta clase modela el concepto de rol al que pueden pertenecer los usuarios.

- *Evento:*

Esta clase modela todo lo referente al evento. En esta clase se especificarán todos los datos acerca de un evento, como por ejemplo, las fechas de inicio y finalización. Además, contendrá una lista de participantes que podrán participar en el evento.

- *Participante:*

Esta clase del modelo pertenece a la abstracción de un participante que pertenece a un evento. En ésta, se especificará el nombre, el número de votos y una pequeña descripción que reúna las condiciones del participante.

- *UsuarioEventoParticipantes:*

La implementación de esta clase ha sido necesaria para tener el control de las votaciones de los usuarios. Se utiliza para almacenar en la base de datos un usuario que vota a un determinado evento, para poder conocer si el usuario ha votado, o no. En esta clase se especifica el nombre del usuario y el identificador del evento en el que votó.

- *ParticipanteVotado:*

Para complementar la clase anterior, se necesita almacenar a todos los candidatos que ha votado un usuario. Es imprescindible para poder modificar el voto de un usuario cuando sea necesario. Aquí se almacena el participante al que votó y el número de votos que se le asignó.

- *EventoUsuarioAsignado:*

Esta clase es utilizada para llevar el control de la autorización de los usuarios en un determinado evento. Por tanto, se especificará el nombre de usuario y el evento en donde le esté permitido votar.

- *UsuarioDAO:*

Se encarga del acceso a la base de datos cuando se necesite almacenar todo lo relevante a los usuarios de la aplicación. Dará opción de guardar, modificar, eliminar y buscar usuarios. Además, obtendrá el manejador que hace posible la manipulación de base de datos con objetos en java (EntityManager de JPA).

- *EventoDAO:*

Se encarga del acceso a la base de datos cuando se necesite almacenar todo lo relevante a los eventos de la aplicación. Dará opción de guardar, modificar, eliminar y buscar eventos. Además, contendrá los aspectos necesarios para almacenar los candidatos que participan en el evento. Como en los demás objetos DAO, también obtendrá el manejador que hace posible la manipulación de base de datos con objetos en java (EntityManager de JPA).

- *ParticipanteVotadoDAO:*

Se encarga del acceso a la base de datos cuando se necesite almacenar todo lo relevante a los candidatos que el usuario votó (en el caso de mayoría simple tan solo se otorgan los votos a un candidato, pero en otros sistemas pueden darse votos a distintos candidatos). Dará opción de guardarlos y eliminarlos, además de buscarlos. Como en los demás objetos DAO, también, obtendrá el manejador que hace posible la manipulación de base de datos con objetos en java (EntityManager de JPA).

- *UsuarioEventoParticipantesDAO:*

Se encarga del acceso a la base de datos cuando se necesite conocer todo lo referente a los usuarios que votaron a un determinado evento. Se podrá crear la relación entre un usuario y el evento en que votó así como eliminarla. Como en los demás objetos DAO, también obtendrá el manejador que hace posible la manipulación de la base de datos con objetos en java (EntityManager de JPA).

2. Usuarios:

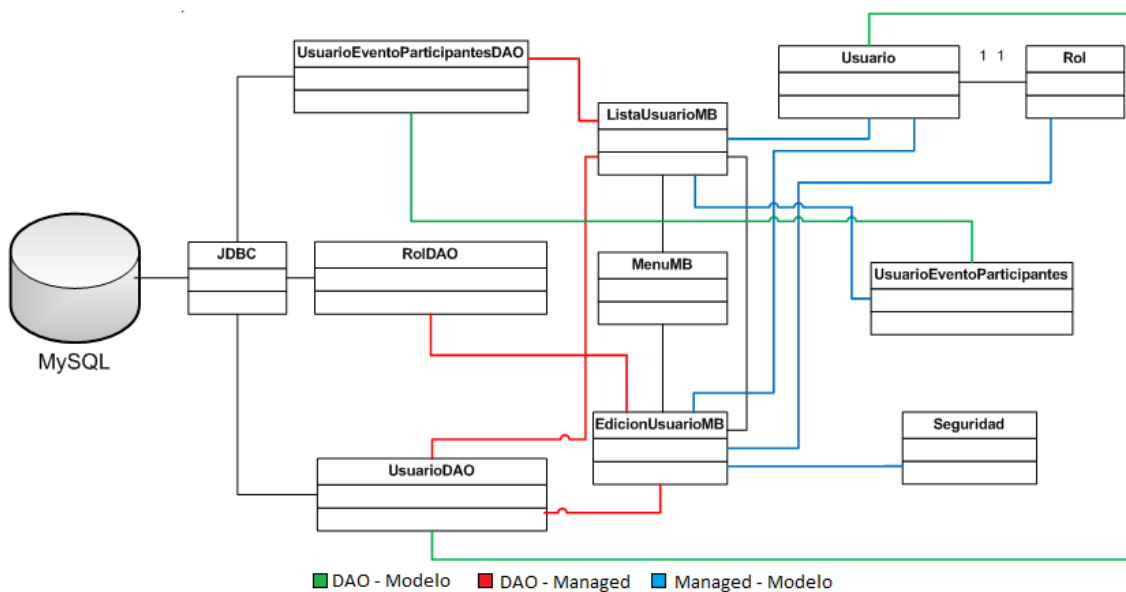


Figura 13: Diagrama de clases: usuarios.

- *ListaUsuarioMB:*

Es el encargado de manejar la lista de los usuarios disponibles registrados en el sistema. Gestionará la selección de un usuario determinado o bien dar paso al siguiente managed bean para la creación de uno nuevo.

- *EdicionUsuarioMB:*

Se encarga de la creación de un usuario nuevo y almacenarlo en la base de datos una vez rellenado los datos necesarios. También se encarga de la modificación de un usuario existente.

- *Seguridad:*

Se encarga de aportar la seguridad que requiere la contraseña de usuario. En ella se utiliza un algoritmo criptográfico para cifrar la contraseña. El algoritmo utilizado es SHA (Secure Hash Algorithm).

3. Roles:

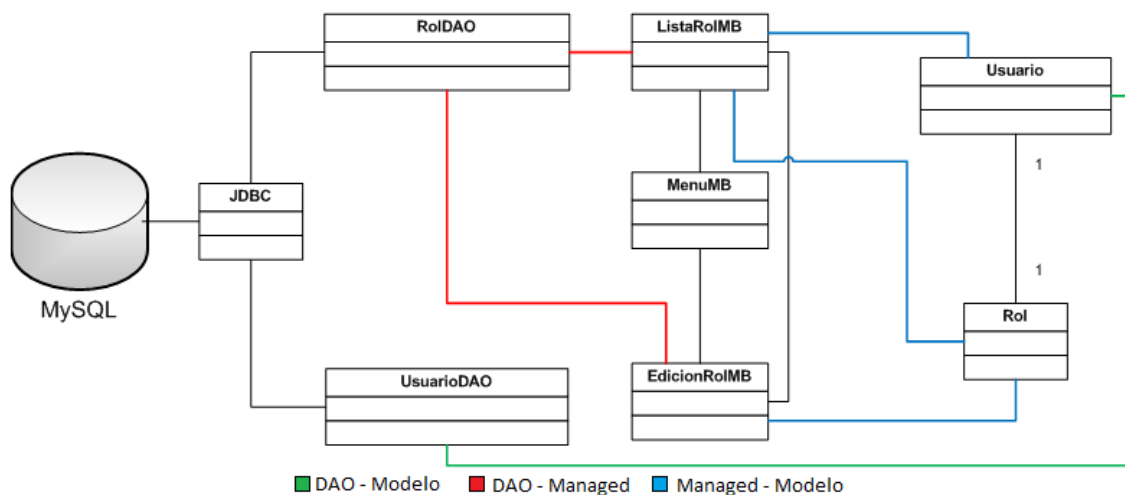


Figura 14: Diagrama de clases: roles.

- *ListaRolMB:*

Es el encargado de manejar la lista de los roles disponibles registrados en el sistema. Gestionará la selección de un rol determinado o bien dar paso al siguiente managed bean para la creación de uno nuevo.

- *EdicionRolMB:*

Se encarga de la creación de un rol nuevo y almacenarlo en la base de datos, una vez rellenado los datos necesarios. También se encarga de la modificación de un rol existente.

4. Votación:

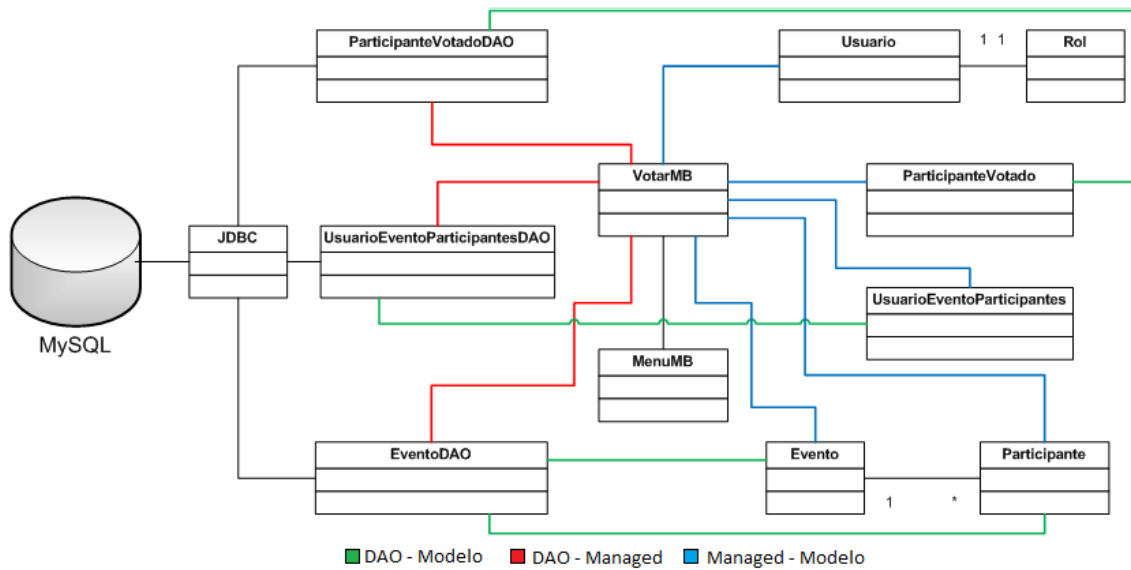


Figura 15: Diagrama de clases: votación.

- **VotarMB:**

Contiene toda la lógica a la hora de emitir un voto. La clase se encarga de distinguir el sistema de votación al que pertenece un determinado evento y en función de esto, el almacén de votos en la base de datos se hace de una manera u otra. Por tanto, en esta clase podemos encontrar los diferentes métodos que corresponden a cada sistema de votación: mayoría simple, mayoría cualificada, sistema acumulativo, sistema ponderado, voto preferencial y voto por aprobación.

En resumen, gestiona todo lo referente a la emisión de votos de los usuarios hacia su candidato o candidatos favoritos.

5. Resultados:

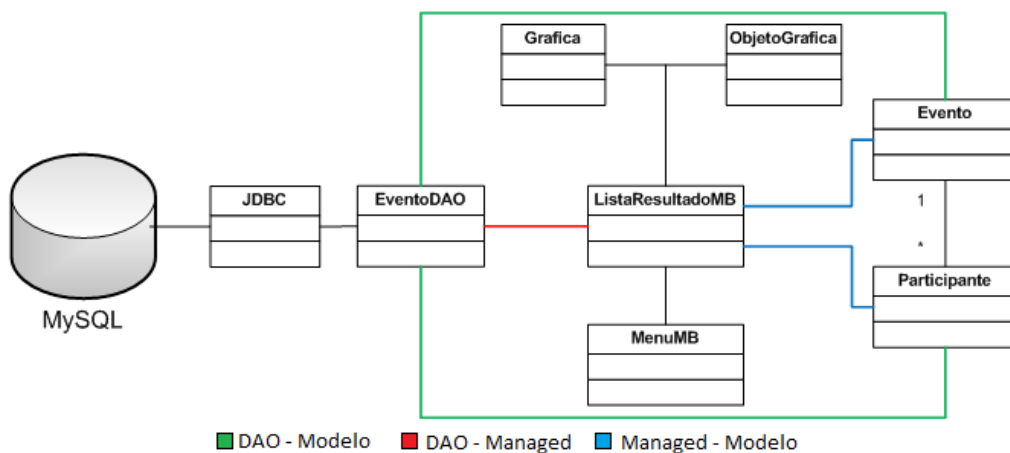


Figura 16: Diagrama de clases: resultados.

- *ListaResultadoMB:*

Es el encargado de manejar la lista de los resultados disponibles registrados en el sistema. Gestionará la selección de un evento determinado para obtener la gráfica que muestre el resultado del acontecimiento.

- *ObjetoGrafica:*

Esta clase se utilizará para representar el resultado de las votaciones. La gráfica hace uso de esta clase para su construcción. Aquí se especifica el nombre del candidato y el número de votos que tiene.

- *Grafica:*

Es el objeto a través del cual se podrá ver la gráfica representada. Hace uso de una colección de ObjetoGrafica para construir la gráfica al completo. Esto es posible gracias a JFreeChart, una librería para gráficos escrita en Java que facilita la creación de gráficos de calidad profesional. Además es un API consistente y bien documentado con soporte para un amplio rango de tipos de gráficas. Se ha optado por utilizar esta tecnología puesto que es flexible y fácilmente extendible y tiene la posibilidad de ser usado indistintamente en tecnologías de servidor (como nuestra aplicación) y de cliente.

6. Login:

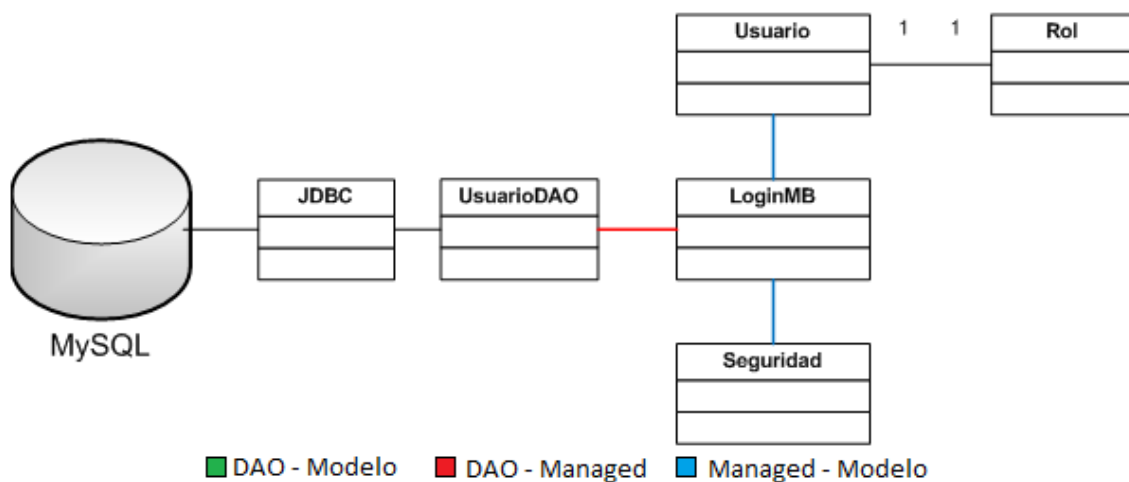


Figura 17: Diagrama de clases: login.

3.3.2.3.- Diagrama de base de datos.

A continuación mostramos el diagrama Entidad/Relación de la base de datos. Hemos elegido este diagrama ya que son modelos que expresan las entidades relevantes para un sistema de información, así como sus interrelaciones y propiedades. La base de datos está compuesta de siete tablas, todas ellas indexadas por la clave primaria (especificada con una llave):

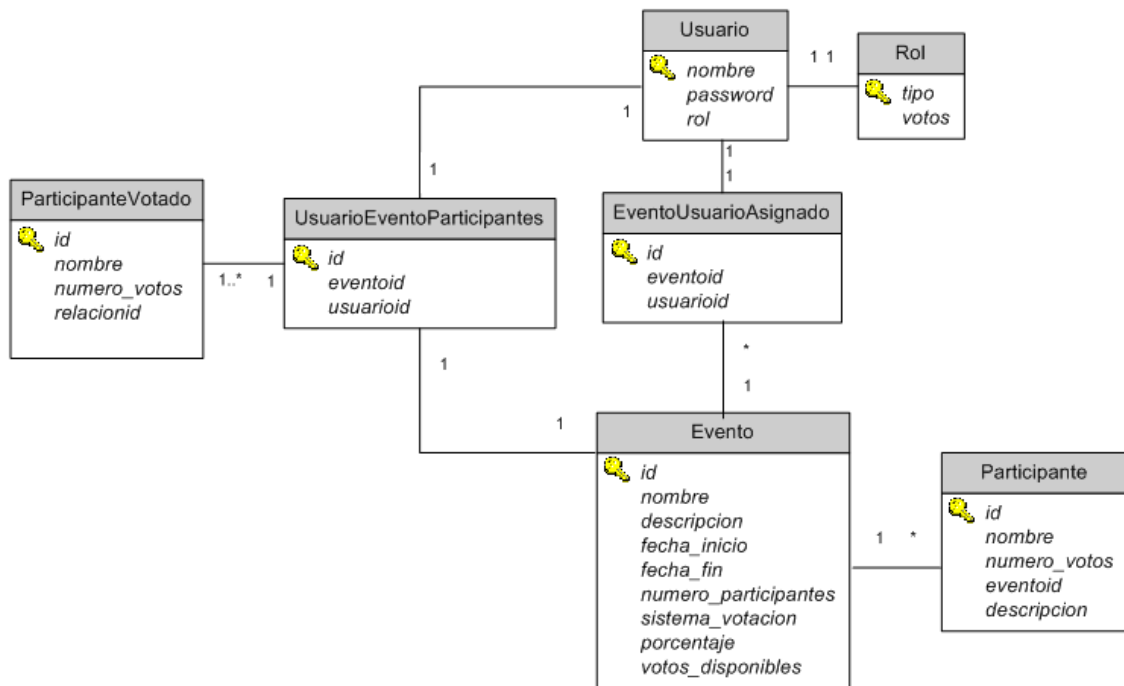


Figura 18: Diagrama de base de datos.

- **Usuario:**

En esta tabla de la base de datos se almacenarán todos los usuarios que pueden utilizar la aplicación. También está incluido el administrador. En ésta, se almacenará el nombre y contraseña del usuario, así como el rol que desempeña. Cabe destacar que es obligatorio que un usuario tenga un rol, para ser guardado con éxito.

- **Rol:**

En esta tabla de la base de datos se almacenarán todos los roles disponibles que hay en la aplicación. Serán creados por el administrador y habrá que especificar el nombre del rol y el número de votos de los que dispone ese rol para que el sistema de votación ponderado haga uso de ello.

- *Evento:*

En esta tabla de la base de datos se almacenarán todos los eventos disponibles en el sistema. Se guardará el nombre del evento, una descripción, la fecha de inicio y la de finalización, el número de participantes, el sistema de votación, el porcentaje mínimo de votos para ser el candidato ganador (mayoría cualificada) y los votos disponibles que tiene ese evento (sistema de votación acumulativo).

- *Participante:*

En esta tabla de la base de datos se almacenarán todos los participantes que componen un evento. Es importante saber que no puede haber un participante que no corresponda a ningún evento. Esta tabla contiene la información del nombre del candidato, el número de votos que tiene, el identificador del evento al que pertenece y una pequeña descripción acerca de él.

- *EventoUsuarioAsignado:*

En esta tabla de la base de datos se almacenarán todos los usuarios que están autorizados para votar a determinados eventos. En ella se almacena el identificador del evento y el del usuario al que corresponde.

- *UsuarioEventoParticipantes:*

En esta tabla de la base de datos se almacenarán todas las relaciones que haya entre el voto de un usuario y el evento. Es decir, al emitir el voto, tanto el usuario como el evento se almacenarán, llevando así un control exhaustivo de la aplicación. Gracias a esta tabla, sabremos los usuarios que han votado en un evento.

- *ParticipanteVotado:*

En esta tabla de la base de datos y, en complemento con la tabla anterior, se almacenarán el número de votos que ha emitido un usuario en un evento y a los candidatos a los que votó. Esto es necesario para poder modificar un voto, si fuese necesario. Por tanto, tendremos que especificar el nombre del candidato al que se votó, el número de votos que le otorgó y el identificador de la tabla anterior, para poderlo relacionar con un usuario. Se crearán tantos registros en esta tabla como votos a cada candidato.

3.3.2.4.- Diagrama de interfaz de usuario.

A continuación mostraremos el diagrama de clases de todas las páginas que nos podemos encontrar en la aplicación. Se ha optado por indicar el flujo que sigue la aplicación. Es importante decir que desde todas las páginas se puede salir de la sesión, al igual que se puede acceder al menú desde todas ellas. Esto no ha sido reflejado puesto que recargaría mucho el diagrama y lo haría menos legible.

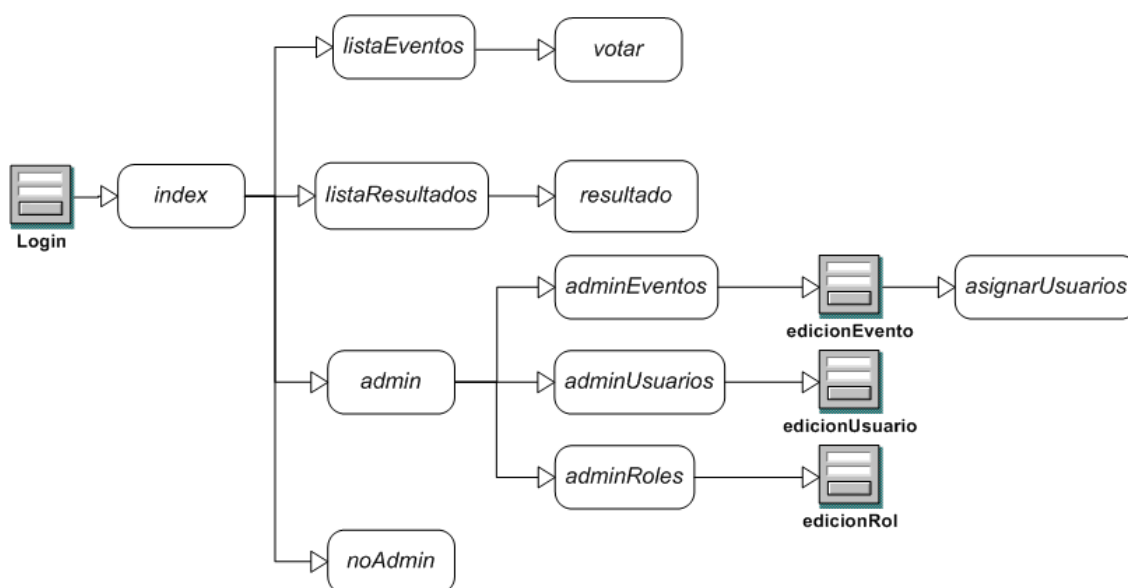


Figura 19: Diagrama de interfaz.

- **Login:**

Este formulario se utiliza para recoger los datos que el usuario ha de introducir para acceder a la aplicación.

- **Index:**

Esta página contiene una breve descripción para que el usuario tenga conocimiento de lo que trata la aplicación. Además, desde ésta se podrá acceder al resto de páginas que forman la aplicación.

- **ListaEventos:**

En esta página está ubicada una lista de todos los eventos que están registrados en el sistema. Se utiliza para que un usuario pueda seleccionar uno de ellos y poder votar. Sin embargo, si el evento ha finalizado, se le mostrará un mensaje de error dónde diga que no es posible emitir ningún voto.

- **ListaResultados:**

En esta página está ubicada una lista de todos los eventos que están registrados en el sistema. Se utiliza para que un usuario pueda seleccionar uno de ellos y poder ver

el resultado de dicho evento. Sin embargo, si el evento no ha finalizado, se le mostrará un mensaje de error dónde diga que no es posible ver el resultado hasta que no haya finalizado el evento.

- *Admin:*

Esta página contiene una serie de opciones para que el administrador pueda manipular y mantener el sistema. Las opciones serán: edición de eventos, edición de usuarios y edición de roles.

- *NoAdmin:*

A esta página se accede cuando un usuario, sin ser administrador, intenta acceder al menú de administrador ("Admin"). Se mostrará un mensaje de error informándole de que no tiene privilegios suficientes para acceder al recurso.

- *Votar:*

Esta página es la más importante para el usuario. Aquí, una vez elegido un evento determinado, se podrá votar al candidato o candidatos. Dependiendo del sistema de votación, la página mostrará uno u otro aspecto ya que hay grandes diferencias a la hora de emitir un voto entre los distintos sistemas de votación.

- *Resultado:*

Esta página se encarga de mostrar el resultado del evento seleccionado. Podrá visualizarse una gráfica con el número de votos de los candidatos que participan, así como la fecha de inicio y finalización del evento, el nombre de éste y el ganador.

- *AdminEventos:*

En esta página está ubicada una lista de todos los eventos que están registrados en el sistema. Se utiliza para que el administrador pueda editar uno de ellos, eliminar o añadir un evento nuevo.

- *AdminUsuarios:*

En esta página está ubicada una lista de todos los usuarios que están registrados en el sistema. Se utiliza para que el administrador pueda editar uno de ellos, eliminar o añadir un usuario nuevo.

- *AdminRoles:*

En esta página está ubicada una lista de todos los roles que están registrados en el sistema. Se utiliza para que el administrador pueda editar uno de ellos, eliminar o añadir un rol nuevo.

- *EdicionEventos:*

Este formulario recoge todos los datos que rellena el administrador al crear o modificar un evento.

- *EdicionUsuarios:*

Este formulario recoge todos los datos que rellena el administrador al crear o modificar un usuario.

- *EdicionRoles:*

Este formulario recoge todos los datos que rellena el administrador al crear o modificar un rol.

- *AsignarUsuarios:*

En esta página se elige a los usuarios que estarán autorizados para votar en un evento determinado. Mostraremos una lista seleccionable de usuarios y además se incluirá una "seleccionar todos" y "quitar selección" para que el proceso de selección sea más cómodo.

3.4.- Implementación

Este apartado permitirá mostrar una visión global de la construcción final de la aplicación. Analizaremos, con detalle, la disposición del servidor y toda la funcionalidad que tanto éste como el cliente deben cumplir para el correcto funcionamiento de la aplicación.

Es importante describir las tecnologías utilizadas para la creación del software, por tanto, se hará un breve resumen acerca de todas ellas.

3.4.1.- Funcionalidad

- *Funcionalidad del servidor*

El servidor es el encargado de albergar toda la lógica de negocio de la aplicación. Con esto conseguimos aumentar la ligereza del cliente, lo que se traduce en una mejora de rapidez y fluidez. El servidor, a su vez, es el encargado de repartir las peticiones de respuesta que se hagan en la aplicación. La comunicación entre cliente y servidor se realiza mediante un navegador web y cada usuario tendrá un identificador de sesión independiente para poder ofrecer un sistema multiusuario.

El funcionamiento real del servidor está compuesto por un servidor Tomcat (servidor web) y una base de datos MySQL. En la base de datos MySQL almacenaremos todos aquellos datos sensibles al almacenamiento, como datos de los eventos, participantes, etc.

- *Funcionalidad del cliente*

El cliente será el navegador que tenga instalado en el equipo. La aplicación es compatible con todos los navegadores: Explorer, Firefox, Google Chrome, Safari, Opera, Lobo ... Al ser la aplicación muy ligera, el cliente no tendrá que disponer de grandes recursos para conseguir gran fluidez.

3.4.2.- Tecnologías

- *Eclipse:*

Para la implementación de la aplicación en Java hemos utilizado el entorno eclipse. Este entorno de programación [14] es una comunidad de código abierto, cuyos proyectos se centran en construir un entorno de desarrollo integrado (IDE) compuesto de librerías, herramientas y compiladores extensibles para construir, desarrollar y mantener el software durante su ciclo de vida.

- *MySQL:*

Para el desarrollo de la aplicación se ha utilizado el sistema de gestión de bases de datos MySQL [15], porque nos da la rapidez y la capacidad de una base de datos relacional capaz de albergar cientos de registros sin ralentizar el sistema, algo muy importante para esta aplicación en concreto. Además contiene múltiples motores de almacenamiento permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.

- *Tomcat:*

El contenedor de servlets utilizado es el que ha sido desarrollado bajo el proyecto Jakarta en la Apache Software Foundation [16]. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Tomcat es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. Además, puede funcionar como servidor web por sí mismo, es más, es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Es necesario decir que Tomcat fue escrito en Java,

y, por tanto, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

- *Hibernate:*

Es una herramienta de mapeo objeto-relacional [17] para la plataforma que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Para limar la diferencia entre los datos usados en la aplicación con los de la base de datos, el desarrollador debe detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información, Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

- *JPA:*

La API de persistencia desarrollada para la plataforma Java EE [18] e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos y permitir usar objetos regulares.

- *JSF:*

Es una tecnología [19] y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas. Además incluye un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

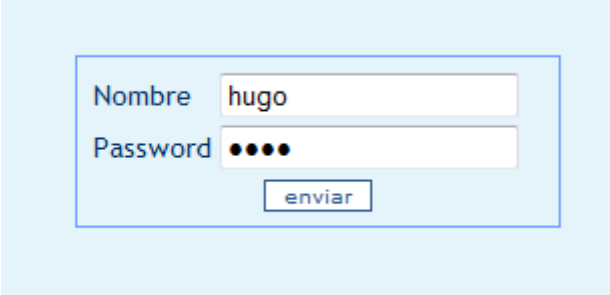
- *HeidiSQL:*

Es un interfaz gráfico para administrar MySQL [20]. Permite navegar y gestionar las bases de datos MySQL de una forma intuitiva y con un interfaz Windows. Es un programa gratuito y tiene un tamaño de descarga inferior a los 2Mb y de ejecución rápida y liviana.

3.4.3.- Interfaz gráfica

A continuación, se mostrará el aspecto que tienen las diferentes páginas explicadas en el punto 3.3.2.4 (Diagrama de interfaz).

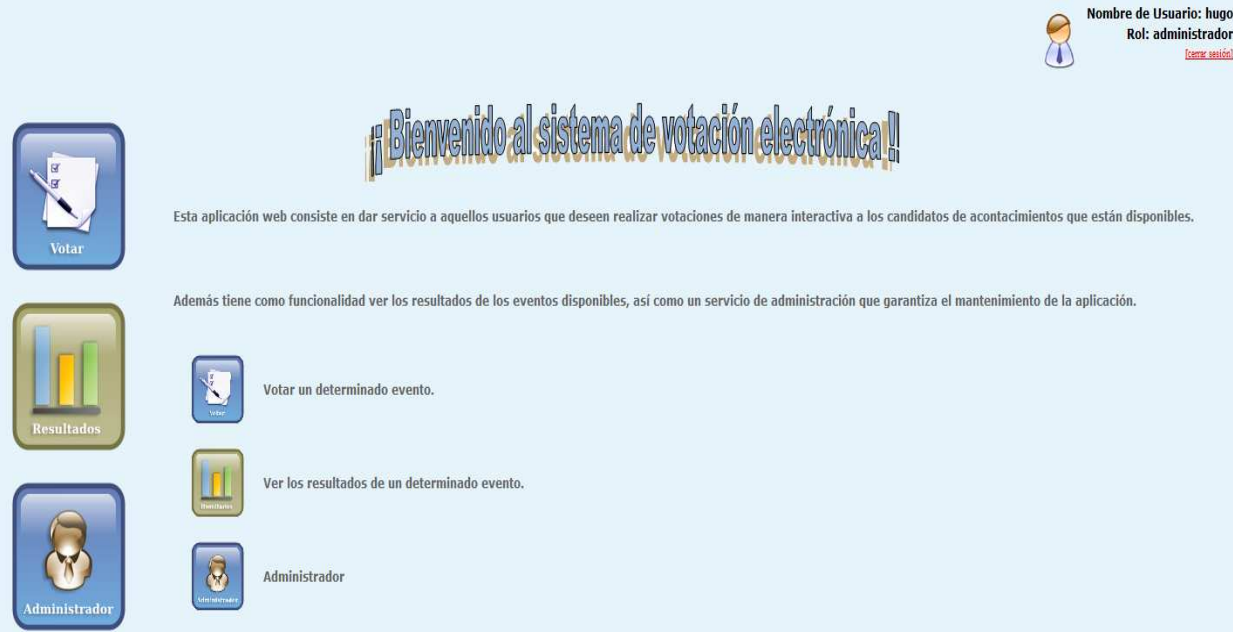
- *Login:*



A screenshot of a login form. It features two input fields: 'Nombre' with the text 'hugo' and 'Password' with four black dots. Below the fields is a button labeled 'enviar'.

Figura 20: Interfaz gráfica de login.

- *Index:*



The index page features a light blue background. At the top right, it displays 'Nombre de Usuario: hugo' and 'Rol: administrador' next to a user icon, with a 'Cerrar sesión' link. The main heading is '¡Bienvenido al sistema de votación electrónica!!'. Below this, a paragraph explains the application's purpose. A sidebar on the left contains three large icons: 'Votar' (document with pen), 'Resultados' (bar chart), and 'Administrador' (user icon). To the right of these icons are three smaller buttons with corresponding labels: 'Votar un determinado evento.', 'Ver los resultados de un determinado evento.', and 'Administrador'.

Figura 21: Interfaz gráfica de index.

- Admin:



Figura 22: Interfaz gráfica de admin.

- AdminEventos:



Figura 23: Interfaz gráfica de adminEventos.

- EdicionEvento:



Figura 24: Interfaz gráfica de edicionEvento.

- *AsignarUsuarios:*



Figura 25: Interfaz gráfica de asignarUsuarios.

- *AdminUsuarios:*



Figura 26: Interfaz gráfica de adminUsuarios.

- *EdicionUsuario:*



Figura 27: Interfaz gráfica de edicionUsuario.

- *AdminRoles:*

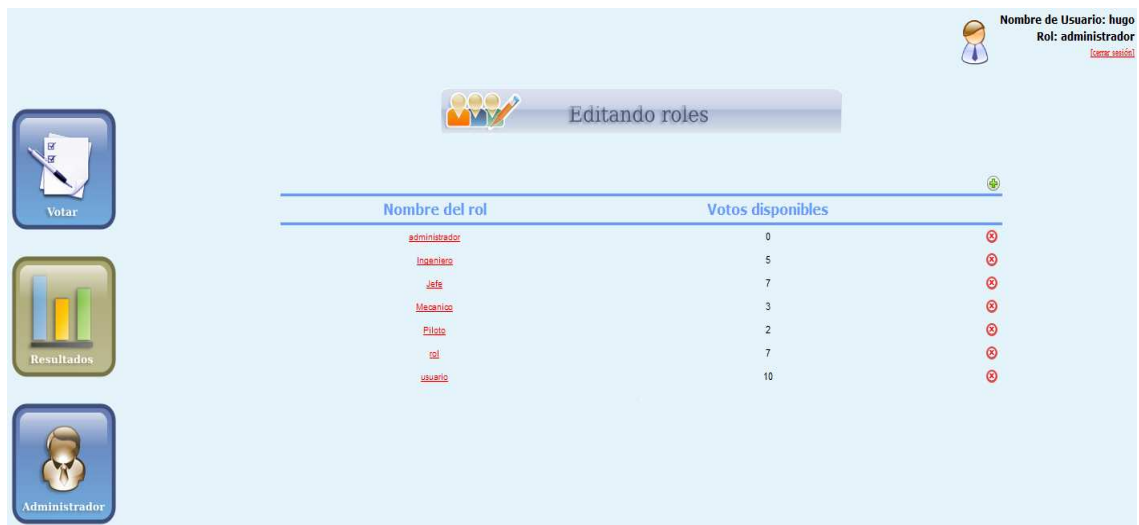


Figura 28: Interfaz gráfica de adminRoles.

- *EdicionRol:*

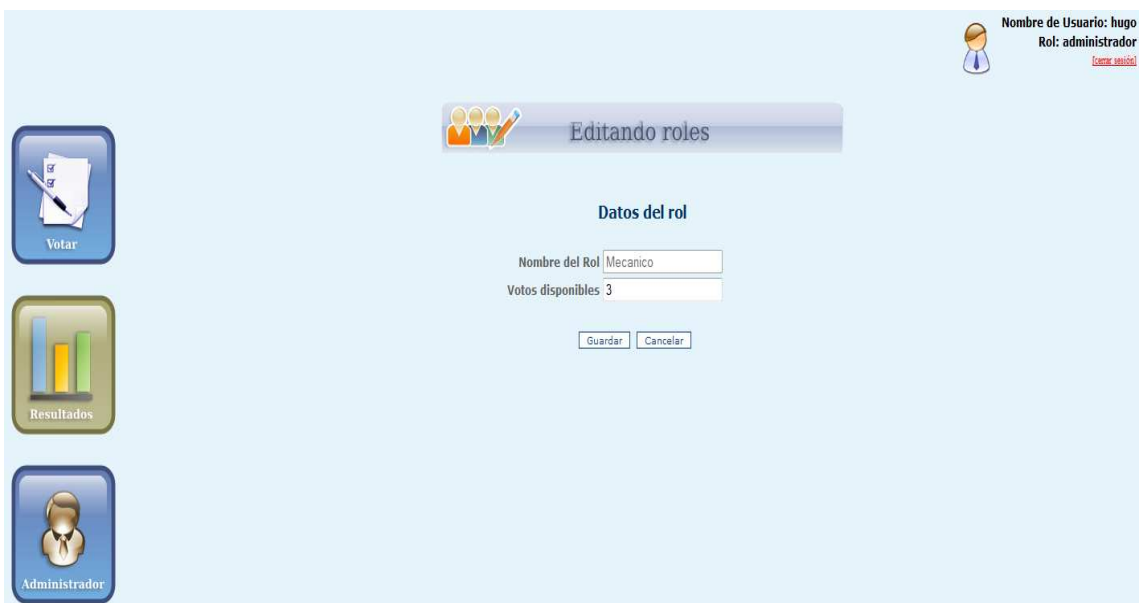


Figura 29: Interfaz gráfica de edicionRol.

- *ListaResultados:*



Figura 30: Interfaz gráfica de listaResultados.

- *Resultado:*

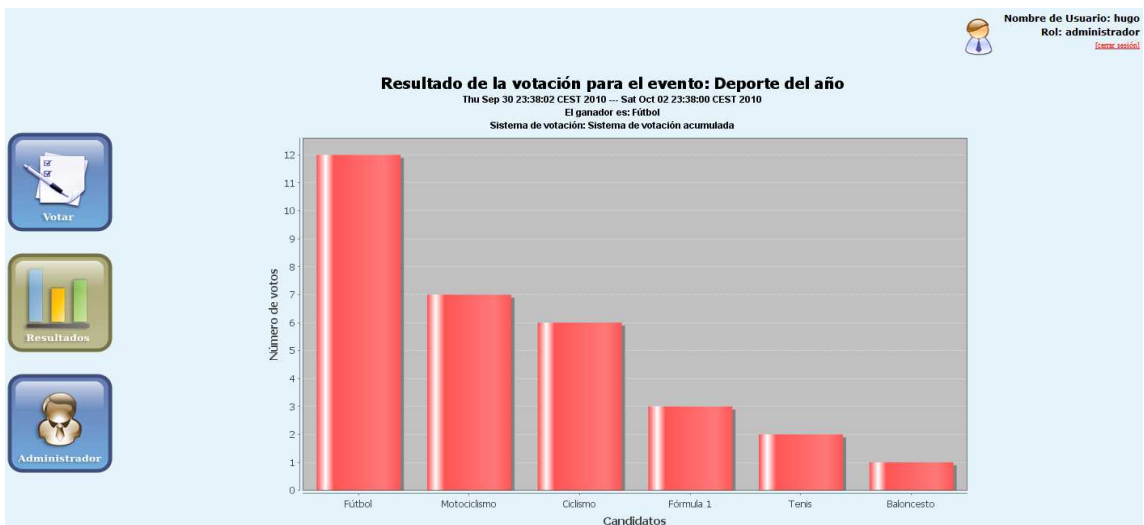


Figura 31: Interfaz gráfica de listaResultados.

- *ListaEventos:*

Nombre de Usuario: hugo
Rol: administrador

Votación

Nombre del evento	Sistema de votación	Número de participantes	Fecha de Inicio	Fecha Fin
Mejor Piloto MotoGP	Preferencial	3	30/09/2010 23:43:21	02/10/2010 23:43:19
Mejor Piloto F1	Ponderada	5	30/09/2010 23:40:02	02/10/2010 23:40:00
¿Qué debería hacer Ronaldo?	Aprobación	4	30/09/2010 23:54:51	02/10/2010 23:54:49
Deporte del año	Acumulativa	6	30/09/2010 23:38:02	02/10/2010 23:38:00
Balón de oro	Mayoría Simple	5	01/10/2010 15:52:47	04/10/2010 15:52:44
¿Silva triunfó en el Metacritic City?	Mayoría Simple	2	30/09/2010 23:51:05	02/10/2010 23:51:03
Mejor Justicia de Baloncesto	Aprobación	4	30/09/2010 23:44:41	02/10/2010 23:44:39
¿Hacen bien Raúl marchándose del Real Madrid?	Mayoría Simple	2	30/09/2010 23:48:04	02/10/2010 23:48:02
¿Es justa la sanción de la FIA a Ferrari?	Mayoría Simple	2	26/07/2010 23:49:15	27/07/2010 23:49:14
Mejor tenista	Mayoría cualificada	4	30/09/2010 23:35:05	01/10/2010 23:35:04

Votar Resultados Administrador

Figura 32: Interfaz gráfica de listaEventos.

- *Votar:*

Nombre de Usuario: Pedro
Rol: Ingeniero

Votación

Nombre del evento: [Mejor Piloto F1](#)
Sistema de votación: [Ponderada](#)
Votos disponibles: 5

Nombre	Descripción	Votos
Fernando Alonso	Ver detalle	2
Sebastian Vettel	Ver detalle	1
Lewis Hamilton	Ver detalle	0
Felipe Massa	Ver detalle	1
Mark Webber	Ver detalle	0

Votar Resultados Administrador

Figura 33: Interfaz gráfica de votar.

- *NoAdmin:*

Nombre de Usuario: Pedro
Rol: Ingeniero

Votar Resultados Administrador

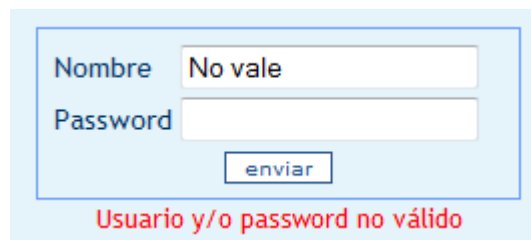
¡No tienes privilegios suficientes para acceder!!

Figura 34: Interfaz gráfica de noAdmin.

3.5.- Pruebas

Para asegurarnos que la aplicación web satisface todos los requisitos del cliente, hemos decidido realizar diversas pruebas para ratificar el buen comportamiento del sistema frente a distintas situaciones.

1. **PRUEBA 1:** dar de alta un rol con datos correctos.
2. **PRUEBA 2:** dar de alta un usuario con datos correctos y con el rol que hemos creado anteriormente.
3. **PRUEBA 3:** dar de alta un evento con datos correctos y asignando el usuario que hemos creado anteriormente.
4. **PRUEBA 4:** salir de la sesión y entrar con el usuario creado.
5. **PRUEBA 5:** votar en el evento que hemos creado anteriormente.
6. **PRUEBA 6:** ver el resultado de la votación que acabamos de realizar.
7. **PRUEBA 7:** introducir un usuario y/o contraseña erróneos.



Nombre

Password

Usuario y/o password no válido

Figura 35: prueba no válida: login y/o password.

8. **PRUEBA 8:** dar de alta un rol o modificarlo con datos no permitidos.



 Editando roles

Datos del rol

Nombre del Rol

Votos disponibles

Error al introducir los votos: no pueden ser negativos.

Figura 36: prueba no válida: editando roles 1.

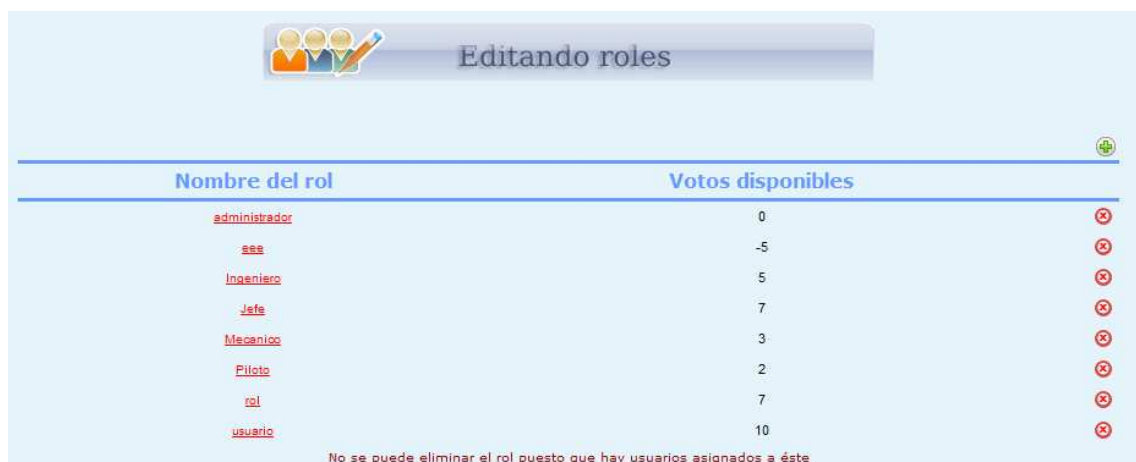
9. **PRUEBA 9:** dar de alta un rol ya existente.



The screenshot shows a web interface titled "Editando roles". Under the heading "Datos del rol", there are two input fields: "Nombre del Rol" with the value "administrador" and "Votos disponibles" with the value "0". Below these fields are two buttons: "Guardar" and "Cancelar". A red error message at the bottom states: "Ya hay un rol con este nombre."

Figura 37: prueba no válida: editando roles 2.

10. **PRUEBA 10:** eliminar un rol que pertenece a un usuario de la aplicación.



The screenshot shows a table titled "Editando roles" with two columns: "Nombre del rol" and "Votos disponibles". Each row has a red 'X' icon in the right margin. A red error message at the bottom states: "No se puede eliminar el rol puesto que hay usuarios asignados a éste".

Nombre del rol	Votos disponibles
administrador	0
aaa	-5
Ingeniero	5
Jefe	7
Mecanico	3
Piloto	2
rol	7
usuario	10

Figura 38: prueba no válida: editando roles 3.

11. **PRUEBA 11:** dar de alta un usuario con datos no permitidos.



The screenshot shows a web interface titled "Editando usuarios" (Editing users). Below the title is a section "Datos del usuario" (User data) with three input fields: "Nombre" (Name) containing "hugo", "Password" (empty), and "Rol" (Role) set to "usuario". At the bottom are "Guardar" (Save) and "Cancelar" (Cancel) buttons. A red error message "Debes introducir un password" (You must enter a password) is displayed below the buttons.

Figura 39: prueba no válida: editando usuarios 1.

12. **PRUEBA 12:** dar de alta un usuario con un nombre que ya existe.



The screenshot shows the same "Editando usuarios" interface. The "Nombre" field contains "hugo", the "Password" field is masked with four black dots, and the "Rol" field is set to "usuario". The "Guardar" and "Cancelar" buttons are present. A red error message "Ya hay un usuario con ese nombre" (There is already a user with that name) is displayed below the buttons.

Figura 40: prueba no válida: editando usuarios 2.

13. **PRUEBA 13:** eliminar un usuario que ya votó en algún evento.

Editando usuarios

Nombre del usuario	Rol del usuario	
Alberto	rol	✘
hugo	administrador	✘
ii	rol	✘
Juan	Piloto	✘
Pedro	Ingeniero	✘
Pape	Jefe	✘
Rodrigo	usuario	✘
Tuercas	Mecanico	✘

Este usuario no es posible eliminarlo puesto que ya votó

Figura 41: prueba no válida: editando usuarios 3.

14. **PRUEBA 14:** dar de alta o modificar un evento con datos no permitidos.

Editando eventos

Datos del evento

Nombre: No válido

Descripción: Fechas erróneas

Fecha de Inicio: ago 31, 2010

Fecha fin: ago 30, 2010

Sistema de votación: Mayoría simple

Lista de candidatos

Nombre	Número de votos	Descripción	
Un candidato	0	Ver	✘

Asignar usuarios

Guardar Cancelar

La fecha de inicio no puede ser posterior a la fecha de fin

Figura 42: prueba no válida: editando eventos 1.

15. **PRUEBA 15:** eliminar un evento que ya ha comenzado.



Nombre del evento	Sistema de votación	Número de participantes	Fecha de Inicio	Fecha Fin
Mejor Piloto MotoGP	Preferencial	3	03/08/2010 23:43:21	03/10/2010 23:43:18
Mejor Piloto F1	Ponderada	5	30/09/2010 23:40:02	02/10/2010 23:40:00
¿Qué debería hacer Ronaldinho?	Aprobación	4	30/09/2010 23:54:51	02/10/2010 23:54:49
Deporte del año	Acumulativa	6	30/09/2010 23:38:02	02/10/2010 23:38:00
Balón de oro	Mayoría Simple	5	01/10/2010 15:52:47	04/10/2010 15:52:44
¿Silva triunfará en el Manchester City?	Mayoría Simple	2	30/09/2010 23:51:05	02/10/2010 23:51:03
Mejor Jugador de Baloncesto	Aprobación	4	30/09/2010 23:44:41	02/10/2010 23:44:39
¿Hace bien Raúl marchándose del Real Madrid?	Mayoría Simple	2	30/09/2010 23:48:04	02/10/2010 23:48:02
Mejor tenista	Mayoría cualificada	4	30/09/2010 23:35:05	01/10/2010 23:35:04
¿Dónde debería jugar LeBron James para el 2011?	Acumulativa	4	30/09/2010 23:52:27	01/10/2010 23:52:26

No se puede eliminar el evento puesto que ya comenzó.

Figura 43: prueba no válida: editando eventos 2.

16. **PRUEBA 16:** ver los resultados de un evento cuando aún no ha finalizado.

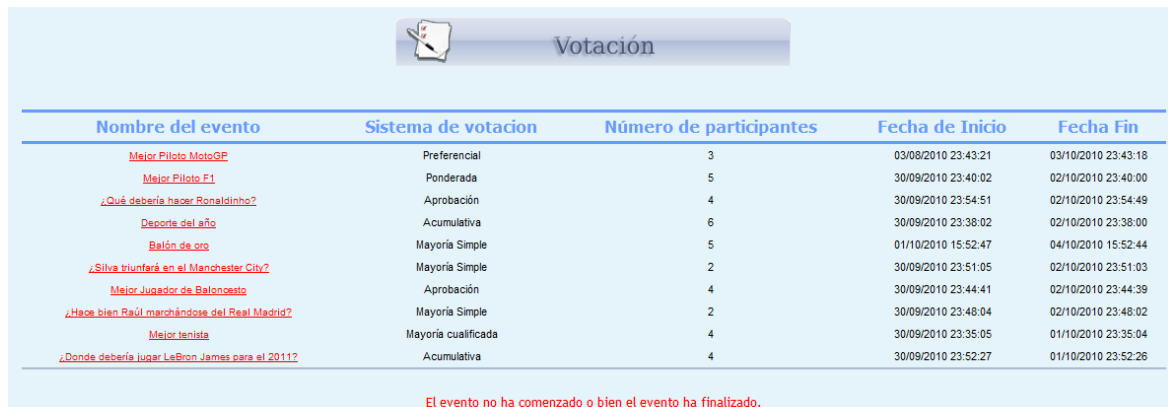


Nombre del evento	Sistema de votación	Número de participantes	Fecha de Inicio	Fecha Fin
Mejor Piloto MotoGP	Preferencial	3	03/08/2010 23:43:21	03/10/2010 23:43:18
Mejor Piloto F1	Ponderada	5	30/09/2010 23:40:02	02/10/2010 23:40:00
¿Qué debería hacer Ronaldinho?	Aprobación	4	30/09/2010 23:54:51	02/10/2010 23:54:49
Deporte del año	Acumulativa	6	30/09/2010 23:38:02	02/10/2010 23:38:00
Balón de oro	Mayoría Simple	5	01/10/2010 15:52:47	04/10/2010 15:52:44
¿Silva triunfará en el Manchester City?	Mayoría Simple	2	30/09/2010 23:51:05	02/10/2010 23:51:03
Mejor Jugador de Baloncesto	Aprobación	4	30/09/2010 23:44:41	02/10/2010 23:44:39
¿Hace bien Raúl marchándose del Real Madrid?	Mayoría Simple	2	30/09/2010 23:48:04	02/10/2010 23:48:02
Mejor tenista	Mayoría cualificada	4	30/09/2010 23:35:05	01/10/2010 23:35:04
¿Dónde debería jugar LeBron James para el 2011?	Acumulativa	4	30/09/2010 23:52:27	01/10/2010 23:52:26

No está permitido ver el resultado del evento porque no ha finalizado.

Figura 44: prueba no válida: viendo resultado.

17. **PRUEBA 17:** votar cuando un evento ha finalizado o bien no ha comenzado.



Nombre del evento	Sistema de votación	Número de participantes	Fecha de Inicio	Fecha Fin
Mejor Piloto MotoGP	Preferencial	3	03/08/2010 23:43:21	03/10/2010 23:43:18
Mejor Piloto F1	Ponderada	5	30/09/2010 23:40:02	02/10/2010 23:40:00
¿Qué debería hacer Ronaldinho?	Aprobación	4	30/09/2010 23:54:51	02/10/2010 23:54:49
Deporte del año	Acumulativa	6	30/09/2010 23:38:02	02/10/2010 23:38:00
Balón de oro	Mayoría Simple	5	01/10/2010 15:52:47	04/10/2010 15:52:44
¿Silva triunfará en el Manchester City?	Mayoría Simple	2	30/09/2010 23:51:05	02/10/2010 23:51:03
Mejor Jugador de Baloncesto	Aprobación	4	30/09/2010 23:44:41	02/10/2010 23:44:39
¿Hace bien Raúl marchándose del Real Madrid?	Mayoría Simple	2	30/09/2010 23:48:04	02/10/2010 23:48:02
Mejor tenista	Mayoría cualificada	4	30/09/2010 23:35:05	01/10/2010 23:35:04
¿Dónde debería jugar LeBron James para el 2011?	Acumulativa	4	30/09/2010 23:52:27	01/10/2010 23:52:26

El evento no ha comenzado o bien el evento ha finalizado.

Figura 45: prueba no válida: votando 1.

18. **PRUEBA 18:** votar cuando un usuario no está autorizado.



Nombre del evento	Sistema de votación	Número de participantes	Fecha de Inicio	Fecha Fin
Mejor Piloto MotoGP	Acumulativa	3	03/08/2010 23:43:21	03/10/2010 23:43:18
Mejor Piloto F1	Ponderada	5	30/09/2010 23:40:02	02/10/2010 23:40:00
¿Qué debería hacer Ronaldinho?	Aprobación	4	30/09/2010 23:54:51	02/10/2010 23:54:49
Deporte del año	Acumulativa	6	30/09/2010 23:38:02	02/10/2010 23:38:00
Balón de oro	Mayoría Simple	5	01/10/2010 15:52:47	04/10/2010 15:52:44
¿Silva triunfará en el Manchester City?	Mayoría Simple	2	30/09/2010 23:51:05	02/10/2010 23:51:03
Mejor Jugador de Baloncesto	Aprobación	4	30/09/2010 23:44:41	02/10/2010 23:44:39
¿Hace bien Raúl marchándose del Real Madrid?	Mayoría Simple	2	30/09/2010 23:48:04	02/10/2010 23:48:02
Mejor tenista	Mayoría cualificada	4	30/09/2010 23:35:05	01/10/2010 23:35:04

Figura 46: prueba no válida: votando 2.

19. **PRUEBA 19:** votar de manera no permitida.



Nombre del evento: **Mejor Piloto MotoGP**
 Sistema de votación: **Acumulativa**
 Votos disponibles: **10**

Nombre	Descripción	Votos
Dani Pedrosa	Ver detalle	9
Jorge Lorenzo	Ver detalle	7
Valentino Rossi	Ver detalle	3

¡¡Vota!!

No dispones de suficientes votos para realizar la votación

Figura 47: prueba no válida: votando 3.

4. Conclusiones y mejoras

4.1.- Conclusión

En conclusión, los objetivos del proyecto se han cumplido, ya que en él nos hemos enfrentado al desarrollo de un software bastante costoso por la gran cantidad de funcionalidad que debía ofrecer. No obstante, se ha conseguido desarrollar, por completo, la aplicación, por lo que el trabajo ha resultado satisfactorio.

Después de seis meses trabajados, el resultado ha sido espléndido. El proyecto supone la culminación a todo el esfuerzo realizado durante elaboración de la aplicación y en el curso de mi formación académica.

Por otro lado, la elaboración de este proyecto, nos ha permitido conocer nuevas herramientas de trabajo como *JSF* y *programación web*, así como tecnologías de las cuales no teníamos conocimiento hasta que no empezamos a trabajar en él, por ejemplo todo lo referente a *BBDD*.

Por primera vez en la carrera, he tenido que enfrentarme a la gestión de un proyecto completo, es decir, un proyecto que tiene un ciclo de vida y que se va desarrollando desde la captura de requisitos hasta obtener el producto final. Además ha sido un reto enfrentarme a un proyecto de tanta envergadura en relación con lo que estaba habituado hasta el momento.

El objetivo de la aplicación es que tenga gran éxito, pudiendo competir, en un futuro, con las aplicaciones similares que hay en Internet.

4.2.- Mejoras

Debido a la falta de tiempo que conlleva hacer una aplicación web profesional, se fueron anotando algunas de las mejoras que sería interesante llevar a cabo en el futuro, sin contar con un estrecho margen de tiempo y así poder documentarse, más ampliamente, sobre las tecnologías que han sido utilizadas en la elaboración de este proyecto.

- *Diseño y estilo de la aplicación*

El diseño web es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web. No es simplemente una aplicación del

diseño convencional, ya que requiere tener en cuenta cuestiones tales como navegabilidad, interactividad, usabilidad, arquitectura de la información y la interacción de medios como el audio, texto, imagen y vídeo.

Consideramos que la aplicación web no está enfocada al diseño, sino a la funcionalidad que tiene. Sin embargo, la unión de un buen diseño con una jerarquía bien elaborada de contenidos aumenta la eficiencia de la web como canal de comunicación e intercambio de datos, que brinda posibilidades como el contacto directo entre el productor y el consumidor de contenidos, característica destacable del medio.

- *Diseño de la BBDD*

Puesto que en el futuro la aplicación manejará una gran cantidad de datos sería muy saludable optimizarla todo cuanto se pueda. En la aplicación hay algunas relaciones entre las tablas, pero no todas las que nos hubiera gustado incluir. No han sido implementadas por desconocimiento en esta rama y por la dificultad que esto suponía debido al estrecho margen de tiempo del que disponía.

Otro punto muy importante es la velocidad de acceso a los datos, con la creación de índices. Éstos son un sistema especial que utilizan las bases de datos para mejorar su rendimiento global. Dado que los índices hacen que las consultas se ejecuten más rápido, podemos estar incitados a indexar todas las columnas de nuestras tablas.

Además, como colofón, y para que sea un programa profesional, habría que registrar la base de datos en la Agencia de Protección de Datos y que cumpla con la reglamentación correspondiente al nivel Básico, ya que es una exigencia de la reglamentación española. La primera condición legal para un sistema de voto electrónico es, pues, su inscripción y reconocimiento oficial

- *Seguridad*

Quizás sea el tema más importante para una aplicación de este calibre. En primer lugar, cabe destacar que la codificación de las contraseñas están implementadas de manera correcta. A pesar de esto, hemos de matizar que no es suficiente. Se requiere, además, tener unas condiciones mínimas de privacidad y seguridad:

1. La autenticación y certificación de la máquina desde la que opera el votante.
2. La autenticación y certificación del servidor del sistema de votación.
3. La identificación del votante.
4. La verificación del sistema de voto.

Por último, sería importante conocer en cada momento todos los movimientos que se vayan produciendo en la aplicación. Con la creación de ficheros (logs) podría subsanarse este aspecto. El fichero de logs es un fichero de texto cuyas líneas son ciertos eventos registrados por el servidor, indicando, para cada uno de ellos, la fecha y hora en que se han producido. En éste, podría haber información de activación del servidor, desactivación del servidor, altas de usuarios, bajas de usuarios, aperturas de sesiones, cierres de sesiones, excepciones del programa...

Referencias

- [1]: **Voto:** http://es.wikipedia.org/wiki/Voto_%28elecciones%29
- [2]: **Participación ciudadana:**
http://es.wikipedia.org/wiki/Participaci%C3%B3n_ciudadana
- [3]: **Definiciones:** <http://www.rae.es/rae.html>
- [4]: **Scytl:** <http://www.scytl.com/>
- [5]: **M-Tempus:** <http://egovflash.wordpress.com/>
- [6]: **E-Vote:** <http://www.swissinfo.ch/ger/index.html>
- [7]: **MyPools:** <http://mypools.com/>
- [8]: **BVE:** http://www.msa.com.ar/productos_dcs_2.htm
- [9]: **Extreme Programming:** <http://www.extremeprogramming.org/>
- [10]: **Kent Beck:** http://es.wikipedia.org/wiki/Kent_Beck
- [11]: **Arquitectura software:** http://es.wikipedia.org/wiki/Arquitectura_de_software
- [12]: **MVC:** http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- [13]: **UML:** http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado
- [14]: **Eclipse:** <http://www.eclipse.org/>
- [15] **MySQL:** <http://www.mysql.com/>
- [16] **Tomcat:** <http://tomcat.apache.org/>
- [17] **Hibernate:** <http://www.hibernate.org/>
- [18] **JPA:** <http://www.oracle.com/technetwork/java/index.html>
- [19] **JSF:** <http://www.oracle.com/technetwork/java/index.html>
- [20] **HeidiSQL:** <http://www.heidisql.com/>