

UNIVERSIDAD DE VALLADOLID

INGENIERÍA INFORMÁTICA

PLANIFICACIÓN Y DISEÑO DE SISTEMAS COMPUTACIONALES

Plan de Desarrollo de Software

Autores:

Jose Maria Pinilla Gimenez

Daniel González Alonso

Vicente Martínez Franco

2 de diciembre de 2016

Diagrama de revisión

Versión	Autor/es	Descripción	Fecha
Inicial	Vicente Martínez Daniel González Jose Maria Pinilla	Primera versión del documento, con la planificación inicial del pro- yecto	28/10/2016

Índice

1. Introducción	3
1.1. Descripción general	3
1.2. Entregables	4
1.3. Evolución del plan de gestión	4
2. Organización del proyecto	5
2.1. Modelo de proceso	5
2.2. Estructura organizacional	5
3. Proceso de gestión	7
3.1. Objetivos y prioridades	7
3.2. Supuestos, restricciones y dependencias	7
3.3. Gestión de riesgos	8
3.4. Mecanismos de monitorización y control	9
4. Procedimiento Técnico	10
4.1. Herramientas, técnicas y métodos	10
5. Tareas, Recursos y Programación	11
5.1. Tareas	11
5.2. Dependencias	14
5.3. Presupuesto y necesidades de Recursos	14
5.4. Programación	15
5.5. Planificación del primer Sprint	16

1. Introducción

Este Plan de Desarrollo de Proyecto (SDP) establece el plan a seguir para el desarrollo de una aplicación que simule el sistema electoral español para la práctica de la asignatura Planificación y Diseño de Sistemas Computacionales.

1.1. Descripción general

El objetivo principal de este proyecto es elaborar una aplicación para simular los resultados electorales con distintos datos y parámetros en el sistema electoral español. Este sistema ha de poder ser usado para distintos tipos de elecciones, ya sea al congreso de los diputados, elecciones autonómicas, municipales o al parlamento europeo. Para ello el sistema ha de permitirle al usuario configurar, ya sea mediante la interfaz o mediante un archivo de configuración (además, el sistema ha de permitir exportar la configuración actual), los siguientes parámetros:

- Tipo de elección, año y mes
- Número total de representantes
- Numero de circunscripciones (además del nombre y número de representantes de cada una)
- Número de candidaturas (además de su nombre largo y abreviado así como su color en el sistema)

Los datos del escrutinio se han de poder introducir tanto por un archivo como mediante la interfaz de usuario. Para ello en la interfaz de usuario se ha de poder cambiar parámetros como los votos en blanco, votos nulos, abstenciones, umbral mínimo para tener representación...

La visualización del resultado final de la simulación ha de ser tanto gráfica como mediante tablas. Además la visualización del reparto de escaños ha de realizarse de uno en uno para facilitar la comprensión por parte del usuario.

1.2. Entregables

Durante el desarrollo del proyecto se van a elaborar la siguiente lista de entregables:

- Plan de proyecto (Entrega 28 de octubre de 2016): Este documento
- Documento de diseño (Entrega 2 de diciembre de 2016): En este se incluirán el conjunto de modelos y diagramas que describan la forma en que la herramienta ha sido diseñada (patrones arquitectónicos, diagramas de casos de uso, etc)
- Documento de implementación (Entrega 10 de enero de 2017): Incluye la descripción de la implementación final del sistema: la estructura de directorios, los paquetes necesarios, sistemas operativos con los que es compatible, etc..
- Documento de seguimiento del proyecto (Entrega 10 de enero de 2017): En este documento se incluirán las incidencias ocurridas durante el desarrollo del mismo así como las medidas tomadas para su solución.
- Manual de usuario y manual de instalación (Entrega 10 de enero de 2017)
- Herramienta final (Entrega 10 de enero de 2017)

Puede ser necesaria la inclusión de más entregables a lo largo del desarrollo del proyecto pero estos son los principales.

1.3. Evolución del plan de gestión

A lo largo del desarrollo del proyecto este documento sufrirá ciertas modificaciones como consecuencia de las distintas decisiones tomadas. La evolución del documento se puede dividir en distintas fases:

- Fase inicial: Incluye la descripción general del proyecto así como la planificación del tiempo, mecanismos de gestión, etc.
- Decisiones de diseño: Se incluirán las distintas decisiones realizadas con respecto al diseño de la aplicación como pueden ser el sistema operativo, el lenguaje, IDE, etc.
- Actualizaciones en la programación: Se incluirá las planificaciones de los distintos Sprints según se vayan realizando.
- Distintas correcciones y actualizaciones en la planificación que se vayan realizando a lo largo del tiempo.
- Versión final del documento para su entrega.

2. Organización del proyecto

2.1. Modelo de proceso

Para el desarrollo de este proyecto usaremos un plan de desarrollo basado en metodología ágil, específicamente SCRUM.

Este proyecto dispone de tres hitos principales que son:

- HITO 1: La entrega de la primera versión de este documento, con lo cual se dispondrá ya del plan de desarrollo general.
- HITO 2: Entrega del documento de diseño. Este hito marca el inicio del proceso de implementación del sistema.
- HITO 3: Entrega del proyecto final. Aquí se presentarán, además de las revisiones de los documentos anteriores, el resto de entregables mencionados en la sección 1.2. Este hito supondrá además el final del proyecto de desarrollo.

2.2. Estructura organizacional

Como hemos señalado anteriormente, para el desarrollo de la practica vamos a seguir la metodología Scrum. En este tipo de metodología existe un único equipo llamado *Equipo Scrum* el cual cuenta de los siguientes roles:

- Dueño del Producto: el dueño del producto (o *Product Owner*) es el responsable a nivel de ganancias o pérdidas del producto y por tanto busca maximizar el valor final del mismo. Dentro del equipo scrum, el dueño del producto participa activamente en cada Sprint escogiendo los elementos que más le interesan. En nuestro caso, los dueños del producto son los profesores Pablo de la Fuente Redondo y Yania Crespo González-Carvajal.
- Equipo de Desarrollo: el equipo de desarrollo es el encargado de construir lo que el Dueño del Producto indica, este equipo es auto-organizado y ninguno de sus miembros tiene un título fijo (el equipo es “*cross-funcional*”). Este equipo es también el encargado de decidir cuantos elementos se van a desarrollar durante cada Sprint y la mejor forma de lograr tal objetivo. En nuestro caso, este equipo estará formado por los tres autores de este documento, todos estudiantes de Ingeniería Informática.
- Scrum Master: el *Scrum Master* es una persona dedicada a entrenar y guiar al Dueño del Producto y al Equipo de Desarrollo a aplicar Scrum y a adoptar

técnicas de desarrollo modernas. En nuestro caso, el profesor Pablo de la Fuente Redondo ocupa el puesto de Scrum Master.

3. Proceso de gestión

3.1. Objetivos y prioridades

Dado que el proyecto se engloba en la asignatura Planificación y Diseño de Sistemas Computacionales, entendemos que el principal objetivo del mismo es el aprendizaje y la obtención de experiencia en las materias relativas a la asignatura. Estas son, por un lado la planificación y gestión de proyectos de software, y por otro el diseño de software.

Entrando más en detalle, encontramos las siguientes prioridades:

- Tenemos unas restricciones de tiempo claras que debemos anteponer a la funcionalidad. Además, el presupuesto es cero, por lo que toda la tecnología utilizada será de licencia libre.
- A lo largo del desarrollo, los “*Sprint Meetings*” nos permitirán ir evaluando la marcha del trabajo y planificar la realización de las tareas. Por otro lado, durante las sesiones de prácticas, los profesores podrán ayudarnos con la evaluación del proyecto.
- Por la naturaleza del trabajo, los riesgos cobran una importancia más baja y su gestión resultará más sencilla. Trabajaremos mano a mano en ellos y procuraremos solucionarlos con reuniones extraordinarias.
- Como hemos comentado, no se adquirirá ningún tipo de software, ya que el presupuesto para el proyecto es cero. Se utilizarán algunas herramientas vistas en clase como el Project o Pivotal Tracker, así como otras con las que el equipo se siente cómodo trabajando como es Overleaf para la edición de documentos en Latex.

3.2. Supuestos, restricciones y dependencias

En este punto vamos a señalar aquellas restricciones y suposiciones en las que se basa el plan de desarrollo del proyecto:

- La principal restricción con la que nos encontramos es el tiempo. Las fechas de los hitos están marcadas y son inamovibles, por lo que es prioridad absoluta cumplirlas.
- Aunque de menor importancia, existe una segunda restricción clave en el proyecto: la funcionalidad. La entrega debe cumplir unos requisitos mínimos para

ser considerada válida.

3.3. Gestión de riesgos

En la realización de este proyecto se ha detectado una serie de posibles riesgos. A continuación pasaremos a enumerarlos y describir las posibles medidas a tomar en el caso de producirse alguno de ellos. La probabilidad prevista para cada riesgo así como su incidencia y la valoración de la importancia de este están indicadas en una escala del 1 al 10, siendo el 1 el de menor importancia y 10 el de mayor.

Descripción	Probabilidad	Incidencia	Valoración
Perdida de datos del programa (errores en los equipos informáticos, borrado accidental...)	2	8	4
Diseño mal elaborado	3	7	4
Imposibilidad de realizar reuniones presenciales entre los miembros del equipo	6	1	3
Falta de familiaridad con las herramientas	6	1	3
Miembro no disponible temporalmente	6	3	5
Retraso en la realización de alguna tarea respecto al tiempo estimado	4	2	3

Frente a estos posibles riesgos se han elaborado las siguientes medidas de acción:

- Para reducir el riesgo de la posible pérdida de datos se dispondrá de varias copias del proyecto en desarrollo en cada una de las máquinas de los miembros del equipo así como al menos otra copia en algún sistema de almacenamiento online.
- En caso de no poder realizar reuniones del equipo de forma presencial se recurrirá a la realización de reuniones a distancia mediante el uso de distintos medios de comunicación como puede ser Skype.
- En caso de la indisponibilidad de algún miembro del equipo por cualquier causa (enfermedad, acumulación de trabajo debido a otras asignaturas, ...) se procederá a la realización de una reunión, ya sea presencial o a distancia, entre los restantes miembros para dividir el trabajo correspondiente al miembro faltante

entre ellos. En el siguiente sprint deberá tenerse en cuenta como puede afectar al desarrollo general del proyecto y realizar la planificación de los siguientes de forma acorde.

- Para intentar evitar la realización de un diseño mal elaborado se realizarán varias revisiones conjuntas del mismo y se contrastarán con los modelos elegidos antes de aprobarlo. Si durante la implementación se detecta algún posible error en el mismo se deberá informar al resto de miembros y se llevará a cabo una revisión conjunta del mismo.

3.4. Mecanismos de monitorización y control

En el seguimiento y monitorización del proyecto dispondremos de las herramientas típicas de scrum además de otras que utilizaremos nosotros para facilitar nuestra organización, todas ellas detalladas a continuación.

En cuanto a las herramientas de control típicas de scrum se encuentran las siguientes:

- Sprint planning: Una reunión al comienzo de cada Sprint donde se fijan los objetivos del mismo. Se identifican los requisitos y las tareas a realizar para completar cada requisito.
- Daily Scrum: reuniones diarias del equipo de desarrollo en las que se comente las tareas en que se ha trabajado el día anterior y en que se va a continuar trabajando ese día.
- Sprint Review: Una reunión al final de cada Sprint, iniciada por el “Product Owner”, donde se analiza la funcionalidad del producto conseguida durante el sprint.
- Sprint Retrospective: Una reunión al final de cada Sprint, donde el equipo del Sprint analiza qué fue bien, qué debe cambiar y cómo realizar los cambios.

Además se utilizarán las siguientes herramientas auxiliares:

- Pivotal tracker: Una herramienta online para la planificación de proyectos, que permite visualizar las distintas tareas a realizar, quien se encuentra trabajando en cada una y el progreso de las mismas.

4. Procedimiento Técnico

4.1. Herramientas, técnicas y métodos

El simulador se alojará en un servicio web, de forma que así sea accesible desde distintas plataformas independientemente del sistema operativo. Las distintas herramientas que se emplearán para el desarrollo del simulador son las siguientes:

- **Elaboración de documentos:** Para la elaboración de los distintos documentos, como los documentos de planificación, diseño, manual de usuario, etc. emplearemos Latex, exactamente para colaborar bajo el mismo documento emplearemos la página web Overleaf.
- **Elaboración de diagramas de diseño:** Para la elaboración de los distintos diagramas emplearemos Astah, puesto que es una herramienta ya conocida por los distintos miembros del equipo de desarrollo.
- **Herramientas de planificación:** para mantener un control sobre las tareas, Sprints y la velocidad a la que desarrolla el equipo se ha decidido emplear *Pivotaltracker*.
- **Sistema Operativo del servidor:** El servidor con el que trabajaremos funciona bajo Linux.
- **Lenguajes de programación:** Como es un servicio web, en la parte del servidor hemos decidido emplear Java, la razón de haber escogido este lenguaje es debido a que los miembros del equipo de desarrollo ya han trabajado con anterioridad con este lenguaje para la elaboración de servicios web, lo que facilita y ahorra tiempo durante el desarrollo.
- **Bases de datos:** En el caso de tener que emplear bases de datos emplearemos MySQL debido a que es el único gestor de bases de datos al que tenemos acceso.
- **Librerías:** para la elaboración de los distintos gráficos emplearemos la librería *d3* puesto que los miembros del equipo ya han trabajado con anterioridad con ésta para la elaboración de gráficos.
- **Frameworks:** Para el diseño de la aplicación web hemos decidido emplear el framework *Bootstrap* para facilitar así el desarrollo de los distintos elementos de la página.
- **Control de versiones:** Para cuando empecemos la implementación del simulador, hemos decidido emplear la herramienta de control de versiones *GIT*.

5. Tareas, Recursos y Programación

5.1. Tareas

Para la estimación de las distintas tareas hemos escogido una escala de Fibonacci, la lista de tareas es la siguiente:

ID: 1	Nombre: Elaborar diagramas de descomposición modular y dependencias
Descripción: Elaborar diagramas que muestran los estilos arquitectónicos universales de descomposición modular y dependencias.	
Condiciones de satisfacción:	
Estimación: 5	

ID: 2	Nombre: Elaborar diagramas de clases
Descripción: Elaborar el diagrama de clases diseño detallado.	
Condiciones de satisfacción:	
Estimación: 3	

ID: 3	Nombre: Elaborar diagramas de secuencia
Descripción: Elaborar los diagramas de secuencia.	
Condiciones de satisfacción: <ul style="list-style-type: none">■ Elaborar al menos 3 casos de uso	
Estimación: 3	

ID: 4	Nombre: Elaborar documento de diseño
Descripción: Elaborar el documento de diseño.	
Condiciones de satisfacción:	
Estimación: 3	

ID: 5	Nombre: Configuración del simulador manualmente
Descripción: Como usuario he de poder configurar el simulador para poder utilizarlo en diferentes tipos de elecciones mediante una interfaz gráfica.	
Condiciones de satisfacción: <ul style="list-style-type: none">■ Disponer de una interfaz de usuario con la que introducir o modificar los datos de configuración■ Permitir configurar tipo de elección, año y mes■ Permitir la configuración del número total de representantes■ Permitir la configuración del número de circunscripciones (por cada una nombre y numero de representantes)■ Permitir la configuración del número de candidaturas (por cada una el color asignado, nombre abreviado y completo)	
Estimación: 3	

ID: 6	Nombre: Exportar configuración
Descripción: Como usuario he de poder exportar los datos de configuración actuales a un archivo de configuración.	
Condiciones de satisfacción:	
Estimación: 1	

ID: 7	Nombre: Importar configuración
Descripción: Como usuario he de poder importar una configuración en el simulador mediante un archivo de configuración.	
Condiciones de satisfacción:	
Estimación: 1	

ID: 8	Nombre: Introducción de datos de escrutinio manualmente
Descripción: Como usuario he de poder introducir o modificar los datos de escrutinio manualmente mediante una interfaz gráfica.	
Condiciones de satisfacción: <ul style="list-style-type: none">■ Disponer de una interfaz gráfica donde poder introducir o modificar los datos del escrutinio manualmente	
Estimación: 3	

ID: 9	Nombre: Importar datos de escrutinio
Descripción: Como usuario he de poder importar un archivo con los datos de escrutinio en el simulador.	
Condiciones de satisfacción:	
Estimación: 5	

ID: 10	Nombre: Introducción de parámetros para repartir escaños
Descripción: Como usuario he de poder modificar fácilmente los parámetros empleados para repartir escaños.	
Condiciones de satisfacción: <ul style="list-style-type: none">■ Disponer de una interfaz de usuario mediante la cual introducir o modificar los datos empleados para repartir los escaños (el umbral mínimo para obtener representación, el número de votos nulos, abstenciones, votos en blanco)	
Estimación: 3	

ID: 11	Nombre: Visualización del reparto de escaños
Descripción: Como usuario he de poder ver el reparto de escaños de la simulación paso a paso.	
Condiciones de satisfacción: <ul style="list-style-type: none">■ La simulación del reparto de escaños debe realizarse de uno en uno para facilitar la comprensión del sistema D'Hont.	
Estimación: 8	

ID: 12	Nombre: Visualización de los resultados en forma de tabla
Descripción: Como usuario debo ser capaz de visualizar los resultados de la simulación mediante una tabla.	
Condiciones de satisfacción:	
Estimación: 3	

ID: 13	Nombre: Visualización de los resultados de la simulación mediante gráficos
Descripción: Como usuario debo ser capaz de ver el resultado de la simulación de forma gráfica.	
Condiciones de satisfacción:	
Estimación: 5	

ID: 14	Nombre: Manual de usuario
Descripción: Como usuario debo de disponer de un manual de usuario.	
Condiciones de satisfacción:	
Estimación: 1	

5.2. Dependencias

Las dependencias encontradas entre las tareas expuestas anteriormente son las siguientes:

- Para la elaboración del documento de diseño (ID 4) se necesita que primero estén elaborados todos los correspondientes diagramas de diseño (ID 1, 2 y 3).
- Tanto la importación como la exportación de los archivos de configuración (ID 6 y 7) dependen de que esté creado el modelo que almacena la información de la configuración, por ello hemos considerado que primero tenemos que tener creada la interfaz para introducir y modificar los datos de configuración manualmente (ID 5). Además, también para la importación también tiene que estar definido el formato de almacenamiento del archivo de configuración, por ello hemos considerado que primero tenemos que hacer la tarea de exportar la configuración (ID 6).
- Con la importación de los datos de escrutinio sucede algo similar al punto anterior (ID 9), por ello primero hemos de elaborar la interfaz para modificar e introducir los datos de escrutinio manualmente (ID 8).
- Para poder visualizar la tanto la simulación como los resultados de ésta (ID 11, 12 y 13) tenemos que tener introducidos todos los datos en el sistema, por ello primero hemos de tener tanto los datos de configuración (ID 5 y 7), como los datos de escrutinio (ID 8 y 9) como los parámetros de la configuración en el simulador (ID 10).

5.3. Presupuesto y necesidades de Recursos

Para el desarrollo del proyecto se dispone de los siguientes recursos:

- Trabajo: Se dispone de tres miembros dentro del equipo de desarrollo, quienes a su vez se encargarán de las correspondientes pruebas de calidad, así como de la planificación del proyecto y de su control. Cada miembro puede dedicar una media de unas 10 horas semanales a la realización del proyecto, 60 por sprint contando el total de los miembros, que pueden variar dependiendo de

la circunstancias, al tratarse los miembros de estudiantes con un calendario de trabajo muy irregular.

- Espacio: Las instalaciones donde se realizará el proyecto ya están disponibles puesto que se llevará a cabo en los edificios de la propia Universidad de Valladolid, por otro lado, para la parte del trabajo que se pueda desarrollar individualmente cada uno de los miembros podrá realizarlo desde su propia casa.
- Equipamiento: Los miembros del equipo disponen de sus propios ordenadores donde realizar el trabajo, en caso de fallo de alguno de estos, la escuela también dispone de ordenadores donde poder trabajar.
- Software: Para la elaboración de los distintos documentos se empleará *Overleaf*, una página donde el equipo puede colaborar en un mismo documento LaTeX. Para la planificación del proyecto se empleará *Microsoft Project*, programa para el cual los miembros del equipo ya disponen de licencia. Para el desarrollo de los diagramas necesarios se empleará *Astah*, programa para el cual los miembros ya disponen también de licencia *Professional*. Finalmente, para la implementación del simulador se empleará algún IDE o editor de archivos gratuito, en caso de necesidad de alguna librería externa se empleará alguna libre.
- Tiempo: Para el desarrollo del proyecto se tienen unos hitos prefijados, la entrega final se llevará a cabo el 10 de Enero de 2017.
- Presupuesto: No se dispone de dinero para el desarrollo del proyecto.

5.4. Programación

Durante este proyecto realizaremos Sprints de 2 semanas de duración, lo que con la fecha final de entrega nos lleva a la realización de 5 sprints a lo largo del desarrollo del proyecto. Además hemos establecido que los días inicio siempre sean los Jueves.

La planificación de los diversos Sprints se realizará durante el “*Sprint Planning*” correspondiente al mismo en relación a las tareas ya realizadas anteriormente y las prioridades de las restantes. Se asignarán tareas de forma que se intente optimizar las horas de trabajo disponibles del equipo, que como ya se ha dicho antes es de alrededor de 60 por Sprint por norma general.

5.5. Planificación del primer Sprint

Como la entrega del segundo Hito, en el cual tenemos que tener realizados los diagramas y el documento de diseño, se encuentra relativamente cerca en cuanto a tiempo, hemos decidido que en el primer Sprint vamos a dedicarnos a hacer varias de las tareas “Chore” con el fin de elaborar varios de estos diagramas, los cuales ocuparán una gran parte del tiempo del Sprint. Además como al final de cada Sprint tenemos que tener por lo menos algún entregable para poder recibir retroalimentación del Product Owner, hemos decidido hacer también alguna de las tareas más simples en cuanto a estimación se refiere. En resumen hemos decidido hacer las tareas:

- Elaborar los diagramas de descomposición modular y dependencias (ID 1).
- Elaborar los diagramas de clases (ID 2).
- Configuración del simulador manualmente (ID 5).

Para la realización de estas tareas, primero tenemos que descomponerlas aún más en subtareas. Las subtareas extraídas durante el Sprint planning son las siguientes:

- Para la realización de los diagramas:
 - Identificar las clases y subsistemas necesarios (5 horas).
 - Diseñar las clases, para ello hemos de identificar sus atributos, operaciones, relaciones, dependencias, estados y requisitos (10 horas).
 - Diseñar los subsistemas, para ello hay que identificar las dependencias, las clases que forman parte de cada subsistema, las interfaces y clases que se comunican con otros subsistemas (15 horas).
- Para la realización de la configuración del simulador:
 - Familiarizarse con el framework *Bootstrap* para poder crear las páginas web (solo lo necesitan 2 de los miembros, y se dedicarán aproximadamente 5 horas cada uno).
 - Diseñar la interfaz para poder introducir los datos de configuración en el simulador (5 horas).
 - Implementar la interfaz (10 horas).
 - Crear el modelo que almacena la información de configuración del simulador (5 horas).

Referencias

- [1] Software Research, *Software Development Plan*, 2009
- [2] Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde, *Scrum Primer, Una introducción básica a la teoría y práctica de Scrum*, 2012