

Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)

Grado en Ingeniería Informática de Servicios y Aplicaciones



APLICACIÓN WEB PARA LA PREPARACIÓN DE LAS OPOSICIONES DE TÉCNICO DE SOPORTE INFORMÁTICO DE LA JUNTA DE CASTILLA Y LEÓN

Alumno: Luis Orozco Zarzosa

Tutora: María Luisa Martín Pérez

BLOQUE 1

PRELIMINARES

TÍTULO

Aplicación web para la preparación de las oposiciones de Técnico de Soporte Informático de la Junta de Castilla y León

AUTOR

Luis Orozco Zarzosa

AÑO

Curso 2017 / 2018

TUTORA

María Luisa Martín Pérez

DEPARTAMENTO

Matemática Aplicada

CENTRO

Escuela de Ingeniería Informática de Segovia

AGRADECIMIENTOS

Durante todo el tiempo en el que he estado realizando este Trabajo Fin de Grado siempre he tenido personas a mi alrededor que me han apoyado y animado para seguir adelante. Ahora me gustaría agradecerles sus esfuerzos y consejos, porque sin su ayuda todo habría resultado mucho más difícil.

En primer lugar, quiero mencionar a mis padres y a mi pareja, que han estado preocupándose desde el primer día y me han ofrecido su apoyo incondicional.

Por último, agradecer a mi tutora María Luisa Martín Pérez su ayuda e interés constante durante todo este tiempo.

RESUMEN

El objetivo de este Trabajo Fin de Grado ha sido desarrollar una aplicación web que sirviese como complemento al estudio mediante la realización de test online para los estudiantes de las oposiciones de Técnico de Soporte Informático de la Junta de Castilla y León.

No viene a sustituir los libros de teoría ni las intensas e interminables horas de estudio, pues es una oposición y es lo que conlleva. La finalidad es complementar dicho estudio:

- Ganando tiempo. Al prepararse una oposición el tiempo es oro (como dijo Benjamin Franklin "Time is money") y una de las principales ventajas del presente trabajo es el ahorro de tiempo al corregir los test de manera instantánea y automática.
- Ganando espacio y mejorando tu organización, ya que en todo momento tienes toda la materia a tu alcance sin tener que llevarlo encima.
- Hacer el estudio más llevadero con algo diferente, que te permita desconectar cambiando la rutina y la forma de estudio.
- Todo ello de una forma eficiente y eficaz.

Utilizando tecnologías web disponibles en la actualidad (Html5, css3, JavaScript, Java, Servlets), entornos de trabajo como Netbeans, editores de texto y código fuente (Sublime Text), programas de edición de imágenes (Photoshop) y otros tantos programas que se detallan más adelante.

El proyecto ha sido desarrollado con diseño responsivo (Responsive Design), ya que actualmente hay multitud de dispositivos desde donde se consume contenido, cada uno con un tamaño y resolución.

Creo que se han conseguido las premisas y objetivos que me marqué al comienzo del trabajo. La aplicación es funcional, atractiva visualmente, brinda otra forma de estudio alternativa, ahorra tiempo y facilita la organización de contenidos.

ABSTRACT

The aim of this Final Degree Project has been to develop a new web application to work as a complement for those students who are studying to become computer support technicians in the Junta de Castilla y León.

This application will not substitute theory books or the intense and long study hours, since competitive exams are really challenging and that is what they need, but they will help us with the following:

- Saving time: When you are studying a competitive exam time is gold- Like Benjamin said "time is money"- and one of the main advantages of this project is that students will be able to save time correcting the test immediately and automatically.
- Saving space and improving your organization, since you have all the materials at your reach without carrying them with you.
- Studying becomes smooth when using these kind of materials. You are able to disconnect from your routine and from your way of studying.
- All the above being efficient and effective.

We will be using state-of-the-art technology like HTML5, CSS3, or JavaScript and an operating environment such as NetBeans, or text editors or source code (Sublime text), image editors as Photoshop and so many other programmes that will be detailed later on.

The project has been designed with a responsive design, given that there are currently many devices from which people access to streaming content, each of them with a different size and resolution.

I honestly believe that I reached the aims that I set for myself at the beginning of this project. The application is functional, visually attractive and it provides a different approach to traditional studying. It helps us save time and makes content organization easier.

ÍNDICE DE FIGURAS

FIGURA 1 - GUÍA SWEBOK	23
FIGURA 2 - ITERACIONES SCRUM	25
FIGURA 3 - CICLO SCRUM	26
FIGURA 4 - SPRINT BACKLOG SCRUM	27
FIGURA 5 - MATRIZ DE PROBABILIDAD E IMPACTO DE RIESGOS	49
FIGURA 6 - DATOS DIAGRAMA DE GANTT	55
FIGURA 7 - DIAGRAMA DE GANTT	56
FIGURA 8 - MODELO ARQUITECTÓNICO	62
FIGURA 9 - ESQUEMA ARQUITECTURA 3 CAPAS.....	63
FIGURA 10 - ESQUEMA MVC.....	65
FIGURA 11 - DIAGRAMA RESUMEN DEL FUNCIONAMIENTO DE LA APLICACIÓN	66
FIGURA 12 - MODELO DE DOMINIO	67
FIGURA 13 - MODELO DE COMPORTAMIENTO	68
FIGURA 14 - MODELO DE DESPLIEGUE	69
FIGURA 15 - BOCETO PANTALLA PRINCIPAL	73
FIGURA 16 - BOCETO COLOR PÁGINA PRINCIPAL CON PHOTOSHOP.....	74
FIGURA 17 - BOCETO A Y BOCETO B DEL MENÚ TEMARIO	74
FIGURA 18 - BOCETO A COLOR SEGUNDA VERSIÓN PÁGINA PRINCIPAL CON PHOTOSHOP	75
FIGURA 19 - BOCETO INICIAL TEST DE EJEMPLO	75
FIGURA 20 - TEST DE USABILIDAD	90
FIGURA 21 - CAPTURA PANTALLA PRINCIPAL VENTAJAS Y TEMARIO	92
FIGURA 22 - CAPTURA EXÁMENES TIMELINE Y FOOTER.....	93
FIGURA 23 - CAPTURA MENÚ TEMARIO	94
FIGURA 24 - CAPTURA MENÚ MATERIAS COMUNES	95
FIGURA 25 - CAPTURA DE TEST.....	96
FIGURA 26 - CAPTURA RESULTADO CORREGIDO	97
FIGURA 27 - ESQUEMA DE LA APLICACIÓN	105
FIGURA 28 - INSTRUCCIONES DE DESCARGA DEL JDK.....	106
FIGURA 29 - INSTRUCCIONES DE DESCARGA DE NETBEANS	107
FIGURA 30 - INSTRUCCIONES INSTALACIÓN BUNDLE	108
FIGURA 31 - INSTRUCCIONES INSTALACIÓN TOMCAT EN NETBEANS.....	109
FIGURA 32 - TOMCAT PANTALLA DE BIENVENIDA.....	110
FIGURA 33 - POOL DE CONEXIONES	111
FIGURA 34 - CONEXIÓN APACHE DERBY	112
FIGURA 35 - INSTRUCCIONES CREAR TABLA A PARTIR DE SENTENCIA SQL.....	112
FIGURA 36 - ESQUEMA TABLA PREGUNTA	112
FIGURA 37 - CONTENIDO CD-ROM	120

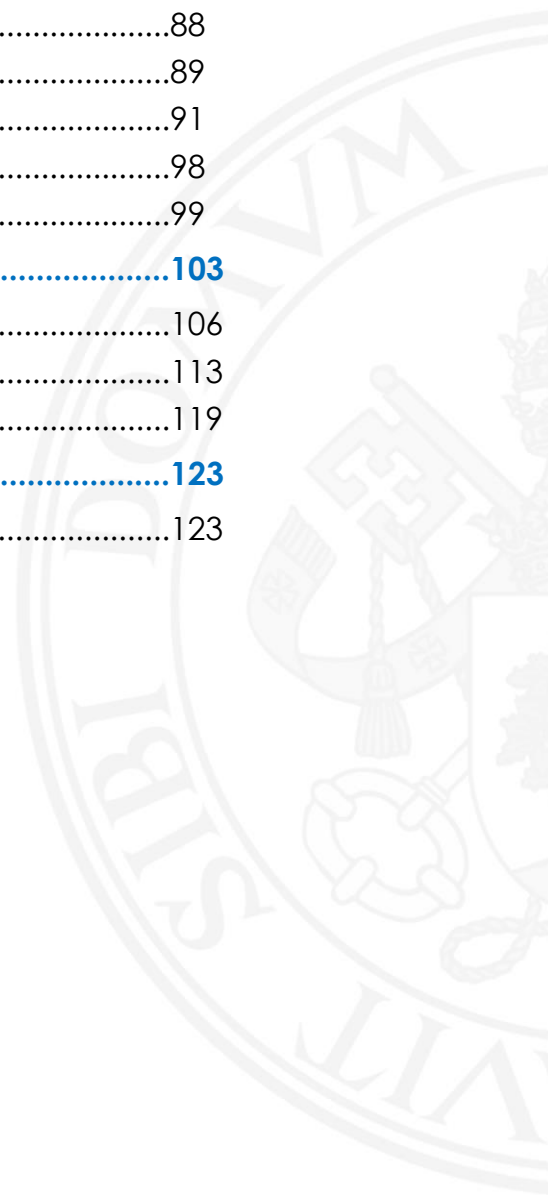
ÍNDICE DE TABLAS

TABLA 1 - PRODUCT BACKLOG	36
TABLA 2 - PLANTILLA HISTORIAS DE USUARIO	37
TABLA 3 - HISTORIA DE USUARIO 1	38
TABLA 4 - HISTORIA DE USUARIO 2	38
TABLA 5 - HISTORIA DE USUARIO 3	39
TABLA 6 - HISTORIA DE USUARIO 4	39
TABLA 7 - HISTORIA DE USUARIO 5	40
TABLA 8 - HISTORIA DE USUARIO 6	40
TABLA 9 - HISTORIA DE USUARIO 7	41
TABLA 10 - HISTORIA DE USUARIO 8	41
TABLA 11 - HISTORIA DE USUARIO 9	42
TABLA 12 - HISTORIA DE USUARIO 10	42
TABLA 13 - HISTORIA DE USUARIO 11	43
TABLA 14 - HISTORIA DE USUARIO 12	43
TABLA 15 - HISTORIA DE USUARIO 13	44
TABLA 16 - HISTORIA DE USUARIO 14	44
TABLA 17 - HISTORIA DE USUARIO 15	45
TABLA 18 - HISTORIA DE USUARIO 16	45
TABLA 19 - HISTORIA DE USUARIO 17	46
TABLA 20 - TABLA TAREAS POR CADA ITERACIÓN	48
TABLA 21 - RIESGO 001	50
TABLA 22 - RIESGO 002	51
TABLA 23 - RIESGO 003	51
TABLA 24 - RIESGO 004	52
TABLA 25 - RIESGO 005	52
TABLA 26 - RIESGO 006	53
TABLA 27 - RIESGO 007	53
TABLA 28 - RIESGO 008	54
TABLA 29 - COSTES HARDWARE	58
TABLA 30 - COSTES SOFTWARE	58
TABLA 31 - ROLES PARA COSTES HUMANOS	59
TABLA 32 - COSTES HUMANOS	60
TABLA 33 - COSTES TOTALES ESTIMADOS	61
TABLA 34 - TABLA PREGUNTA DB	70
TABLA 35 - ESTIMACIÓN HORAS POR ITERACIÓN	76
TABLA 36 - DESGLOSE HORAS POR CADA ROL	77
TABLA 37 - CALENDARIZACIÓN	80
TABLA 38 - HORAS ESTIMADAS VS HORAS REALES	88

ÍNDICE DE CONTENIDOS

I. PRELIMINARES.....	7
RESUMEN.....	8
ABSTRACT.....	9
ÍNDICE DE FIGURAS.....	10
ÍNDICE DE TABLAS.....	11
ÍNDICE DE CONTENIDOS.....	12
II. CONTENIDO	17
1. INTRODUCCIÓN.....	17
1.1. Motivación.....	18
1.2. Objetivos.....	18
2. ESTADO DEL ARTE.....	20
2.1. Propuesta.....	21
3. METODOLOGÍA.....	23
3.1. Scrum.....	24
3.2. Herramientas utilizadas.....	29
4. ANÁLISIS.....	34
4.1. Product Backlog.....	34
4.1.1. Historias de Usuario.....	36
4.2. Sprint Backlog.....	46
4.3. Plan de riesgos.....	48
4.4. Diagrama de Gantt.....	54
4.5. Estimación de los costes.....	57
5. DISEÑO.....	62
5.1. Modelo Arquitectónico.....	62
5.1.1. Arquitectura en Capas.....	63
5.1.2 Patrón MVC.....	64
5.2. Modelo del dominio.....	66
5.3. Modelo de comportamiento.....	67
5.4. Diagrama de despliegue.....	69
5.5. Diccionario de datos.....	69
5.6. Diseño centrado en el usuario.....	70
5.7. Bocetos de la interfaz de usuario.....	72
6. IMPLEMENTACIÓN.....	76
6.1. Iteración 1. Inicio del TFG.....	77
6.2. Iteración 2. Planificación.....	78

6.3. Iteración 3. Desarrollo I.....	81
6.4. Iteración 4. Desarrollo II.....	82
6.5. Iteración 5. Pruebas.....	84
6.5.1. Test de usabilidad.....	84
6.6. Iteración 6. Entrega final.....	86
7. RESULTADOS.....	88
7.1. Costes reales.....	88
7.2. Resultados test de usabilidad.....	89
7.3. Capturas de la aplicación.....	91
8. CONCLUSIONES.....	98
9. TRABAJOS FUTUROS.....	99
III. APÉNDICES.....	103
APÉNDICE 1: MANUAL DE INSTALACIÓN.....	106
APÉNDICE 2: MANUAL DE USUARIO.....	113
APÉNDICE 3: CONTENIDO DEL CD-ROM.....	119
IV. BIBLIOGRAFÍA.....	123
WEBGRAFÍA.....	123



BLOQUE 2

CONTENIDO

II . CONTENIDO

En este segundo bloque se muestra el cuerpo principal o puntos principales de nuestro TFG. Será el bloque más destacado tanto en extensión como en importancia.

I . INTRODUCCIÓN

Este trabajo surgió por pura necesidad personal a la hora de preparar unas oposiciones similares. A la hora de la preparación, el estudio se hacía monótono, se malgastaban horas corrigiendo tests, y se manejaba una cantidad ingente de apuntes. Echaba en falta una herramienta que sirviera de ayuda ante dichos inconvenientes. Esta fue la génesis del estudio que se plantea en este Trabajo Fin de Grado (TFG).

Plantea el desarrollo de una aplicación web que sirva de ayuda a los usuarios que planeen estudiar las oposiciones de Técnico de Soporte Informático de la Junta de Castilla y León.

El público objetivo serían personas con un amplio rango de edad: desde jóvenes, con estudios de bachillerato, buenos conocimientos ofimáticos y desenvueltos en el mundo web, hasta personas con alguna dificultad para realizar todas las acciones con soltura. Por consiguiente, la sencillez era una premisa imprescindible por cumplir. Menús simples e intuitivos que hagan fácil navegar y llegar a la información buscada. Otro factor para tener en cuenta era el diseño. Si la página no es atractiva ni funcional, no invita a la práctica. Por ello, se ha intentado hacer la web de forma que sea sugestiva para el usuario.

1.1. MOTIVACIÓN

El camino del opositor es una auténtica montaña rusa: hay días que te levantas motivado y que crees que ya estás acariciando la plaza, y, sin embargo, otros días la ves tan lejana que se piensa en abandonar. Por tanto, no es fácil mantener esa motivación día tras día. A lo largo del camino vamos a encontrar tantos obstáculos que más de una vez dudaremos de nuestras habilidades y de las posibilidades de llegar a la meta.

Es necesario buscar motivación para el día a día. Personalmente, observé que hacer tests de forma interactiva sería una buena idea, una buena forma de romper la monotonía.

La situación laboral estos últimos años no ha sido la mejor y mucha gente ha decidido opositar. Se ha investigado y para la oposición de Técnico de Soporte Informático no hay precedentes de ninguna aplicación similar que sirva para preparar tests de forma interactiva. Esta es la razón por la que se desarrolla este TFG, cubrir el vacío encontrado y ser un apoyo para el mayor número posible de personas.

1.2. OBJETIVOS

La idea principal es crear una aplicación web que complemente el estudio de las oposiciones de la Junta de Castilla y León de Técnico de Soporte Informático. Los principales objetivos, considerados coherentes y alcanzables son:

- Realizar tests online.
- Recopilar y mostrar toda la información disponible.
- Actualizar noticias y novedades.
- Descargar exámenes de años anteriores.

Una vez que decidí desarrollar dicha idea, era importante fijar unos objetivos para implementarlo todo sin perderlos de vista y no desviarme del camino.

No se pretende sustituir la forma de estudiar, ni los contenidos. Debe ser un complemento, una herramienta de ayuda con objetivos claros.

El primero de los objetivos es permitir romper la monotonía del estudio cambiando la forma de realizar los test, del papel a una forma interactiva de forma similar a como se hace actualmente con los test del carné de conducir en las autoescuelas.

Se dan varias situaciones a la hora de preparar oposiciones: bien lo haces a tiempo completo, bien tras una jornada laboral o educativa. El principal inconveniente en ambas es la falta de tiempo así que todo lo que podamos ahorrar es bienvenido. Hacer los tests y corregirlos al instante, de forma automática, es la principal ventaja implícita en el primer objetivo.

El segundo objetivo es recopilar toda la información relativa a la oposición, tomadas de fuentes como el Boletín Oficial del Estado, como las plazas, anexos, bases, plazos o un resumen con la información más relevante.

Una vez desarrollada la aplicación web, otro de los objetivos es ir actualizando toda la información a medida que se haga pública, de forma que al usuario le sea fácil enterarse de las novedades.

Además, se permite al usuario la opción de descargar exámenes de años anteriores, plantillas, y todo tipo de información.

La aplicación se ha desarrollado de una forma intuitiva, fácil de entender, simple y eficiente, atractiva para la vista y productiva con todo lo que el usuario puede necesitar sin ocupar más espacio que un portátil, móvil, Tablet o cualquier dispositivo electrónico.

2 . ESTADO DEL ARTE

En este punto se analizan las distintas aplicaciones que existen actualmente o han existido en el mercado que realicen funcionalidades similares a las que se propone desarrollar en este TFG.

Empecemos comentando una por una, con sus respectivas ventajas e inconvenientes:

a. <http://www.testsoposicionesgratis.com>.

Este portal web tiene la opción de realizar tests online que es lo que buscamos nosotros, pero no consigue ni ser atractiva visualmente ni dispone de test para la oposición de Técnico de Soporte Informático. Es muy básica.

b. <http://www.opositatest.com/>

Dicho portal web es algo más moderno que el anterior. Tiene ideas buenas que no se me habían ocurrido hasta este momento de investigación, como por ejemplo justificar las respuestas. Saber el motivo por el que has fallado hace que luego lo memorices mejor.

También te muestra un análisis de tus resultados y estadísticas y tendencias personalizadas. Como desventajas tenemos que no tiene soporte para nuestra oposición y que es de pago (entre 3 y 33€ cada mes).

c. <http://www.hacertest.com/oposiciones-tecnicos-auxiliares-informatica-estado/>

En este portal web podemos ver que el método de hacer los tests es muy similar a la idea que ya se tenía y servirá de apoyo para la fase de implementación. Solo dispone de temas para la oposición de Técnico Auxiliar de Informática (mismas oposiciones, pero de nivel nacional), y los temas que tiene están algo desactualizados.

d. <http://www.baquedano.es/>

Desde que planteamos la web, con la experiencia de preparar unas oposiciones a las espaldas, ya contábamos con la ayuda de este portal web. Fue de gran ayuda, ya que tienes todos o gran cantidad de temas para descargar, todos relacionados con las TIC¹.

La desventaja es que no se tiene la opción de realizar test online, ni la apariencia es demasiado amigable.

e. <https://www.taiestado.com/>

Otro de los portales web que realmente sirven de ayuda, en este caso para las oposiciones de Técnico Auxiliar de Informática. Cuenta con ventajas como informar de todas las novedades en un timeline y con un diseño bastante moderno.

La posibilidad de añadir comentarios entre los diferentes usuarios es otra de las funcionalidades que me han gustado. Por el contrario, no dispone de test ni exámenes, sino que hace una redirección a www.baquedano.com.

Después de una búsqueda exhaustiva comprobamos que es bastante difícil encontrar portales web similares al nuestro. Es verdad que existen otras parecidas, pero ninguna con el objetivo de preparar las oposiciones de Técnico de Soporte Informático.

Así, con la falta de portales webs que se dediquen exclusivamente a la preparación de oposiciones relacionadas con el mundo de las TIC mediante test online, como es la de la Junta de Castilla y León, creo que el desarrollo de este TFG rellenará un hueco de conocimiento hasta ahora vacío.

2.1. PROPUESTA

En este TFG se propone una aplicación web relacionada con la preparación de oposiciones de las TIC, centrada en las de Técnico de Soporte Informático, que permita la realización de tests online y su corrección de forma instantánea y automática.

El trabajo estará enfocado en todo momento en el usuario, con un diseño intuitivo, eficiente y atractivo. Intuitivo ya que el público objetivo tendrá unos conocimientos de informática muy variados, eficiente

¹ TIC es una sigla que significa Tecnología de la Información y la Comunicación. Son un conjunto de tecnologías aplicadas para proveer a las personas de la información y comunicación a través de medios tecnológicos de última generación.

porque si la página no hace bien lo que tiene que hacer no valdrá para nada pues no será visitada, y atractiva ya que tiene que ofrecer y transmitir algo a los “clientes” de la web.

En este trabajo se combinarán las ideas que ya teníamos, con las características que más nos han gustado de los portales listados anteriormente y con todos los conocimientos aprendidos durante estos años de carrera.

Además, podemos observar en el apartado 9 futuras ampliaciones o trabajos futuros en caso de ampliar dicho proyecto. Inicio de sesión, resultados guardados para cada usuario, más oposiciones, nuevos apartados, etc.



3 . METODOLOGÍA

Ya en 1945, mucho antes de que existiese la Ingeniería del Software, el matemático George Polya describió este proceso en su libro "How to solve it" (el primero que describe la utilización de técnicas heurísticas en la resolución de problemas).

Básicamente, resolver un problema requiere:

- Comprender el problema (análisis).
- Plantear una posible solución, considerando soluciones alternativas (diseño).
- Llevar a cabo la solución planteada (implementación).
- Comprobar que el resultado obtenido es correcto (pruebas).
- Mantenimiento de dicho software en el tiempo.

2

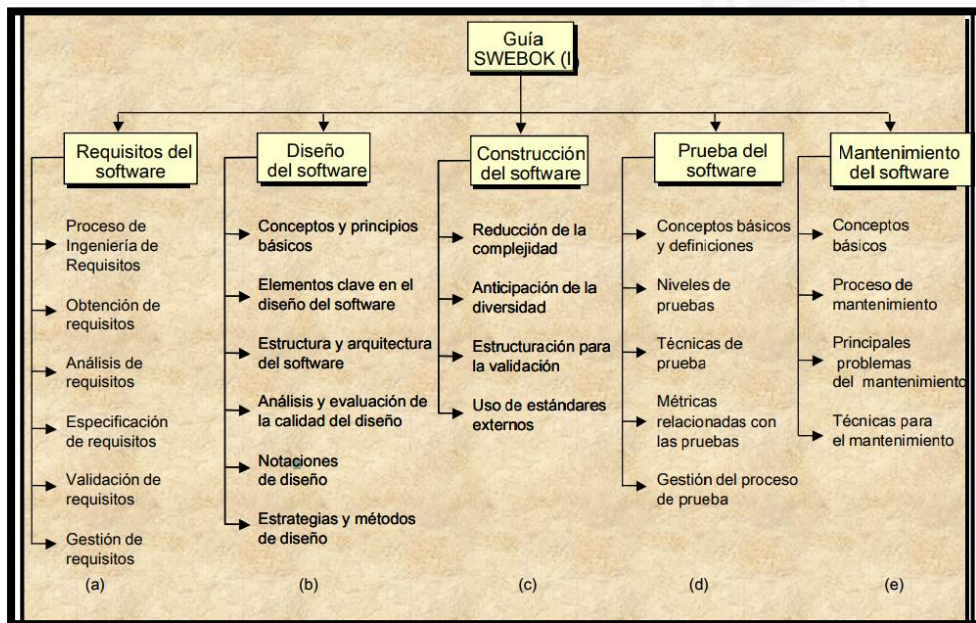


Figura 1 - Guía SWEBOOK

² SWEBOOK, Software Engineering Body of Knowledge, es un documento del área de la Ingeniería del Software que representa un amplio consenso respecto a los contenidos de la disciplina.

Sabiendo qué es lo que se va a hacer, el siguiente paso es definir cómo vamos a hacer todo lo relativo al desarrollo y gestión del proyecto en sí. Para ello, aplicando los conocimientos de Ingeniería del Software, se ha llevado a cabo una investigación sobre las diferentes alternativas con el fin de elegir aquella metodología más conveniente para este TFG.

Para llevar a cabo la investigación se ha tenido en cuenta que nuestro TFG precisa de una metodología ágil, ligera y sobre todo flexible pues se va a compaginar la realización del TFG con las prácticas de empresa y quizás algún trabajo temporal, lo cual dificulta la predicción de objetivos, plazos y demás.

Al tener en cuenta la opinión de la tutora se prefiere una metodología iterativa e incremental que se pueda ir modificando, que una metodología secuencial donde los cambios de diseño, código y pruebas serían más perjudiciales.

En lo relativo a la gestión, por tanto, se necesita flexibilidad en los plazos de entregas o flexibilidad a cambios. En lo relativo al desarrollo en sí, la metodología debe ser apropiada para facilitar el trabajo con el tipo de lenguajes a utilizar.

Así pues, basándose en estas ideas o restricciones se ha elegido la metodología de trabajo **Scrum**.

3.1. SCRUM

Este modelo fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, al analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Nonaka & Takeuchi, The New New Product Development Game, 1986).

Scrum significa Melé, que es un tipo de jugada del deporte rugby en la que todos los jugadores de ambos equipos se agrupan en una formación en la cual luchan por el balón que se introduce por el centro. Si un miembro se viene abajo, se cae toda la melé. En consecuencia, todos los jugadores deben estar muy bien coordinados para apoyarse y empujar al mismo tiempo.

El marco técnico de scrum está formado por un conjunto de prácticas y reglas que dan respuesta a los siguientes principios de desarrollo ágil:

- Gestión evolutiva del producto, en lugar de la tradicional o predictiva.
- Calidad del resultado basado en el conocimiento tácito de las personas, antes que en el explícito de los procesos y la tecnología empleada.
- Estrategia de desarrollo incremental a través de iteraciones (sprints).

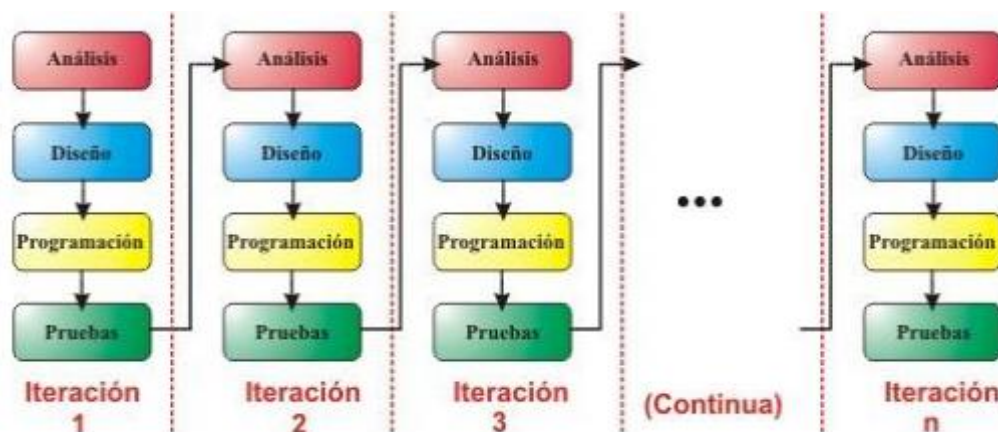


Figura 2 - Iteraciones SCRUM

Se comienza con la visión general del resultado que se desea, y a partir de ella se especifica y da detalle a las funcionalidades que se desean obtener en primer lugar. Cada ciclo de desarrollo o iteración (sprint) finaliza con la entrega de una parte operativa del producto (incremento). La duración de cada sprint puede ser desde una, hasta seis semanas, aunque se recomienda que no exceda de un mes.

Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Es por tanto la duración del sprint, el período de tiempo máximo para descubrir planteamientos erróneos, mejorables o malinterpretaciones en las funcionalidades del producto.

Su principal objetivo es obtener resultados (normalmente prototipos) cuanto antes y adaptarse a los cambios (normalmente, los cambios en los requisitos).

En un caso práctico los pasos y diferentes roles dentro de Scrum, serían los siguientes:

- El cliente se reúne con el **Dueño de producto** ("Product Owner") que toma nota de lo que el cliente quiere ("es la voz del

cliente”), divide el proyecto en historias que componen la pila de producto y prioriza las tareas.

- El **Scrum Manager** (“Scrum Master”) estima el coste de creación del producto con su equipo y asegura que el trabajo vaya bien.
- El cliente aprueba el presupuesto y reordena la pila de producto para que el equipo trabaje prioritariamente en lo más importante.
- Los **Miembros del equipo de desarrollo** desglosan la primera historia en tareas menores para crear la pila de sprint.
- Finalizado el sprint, el Dueño de Producto le muestra al cliente el resultado del trabajo realizado. Así el cliente tiene el primer contacto con su encargo, puede volver a priorizar la pila, cambiar cosas, etc.
- Se repite todo hasta el último sprint y fin del trabajo.

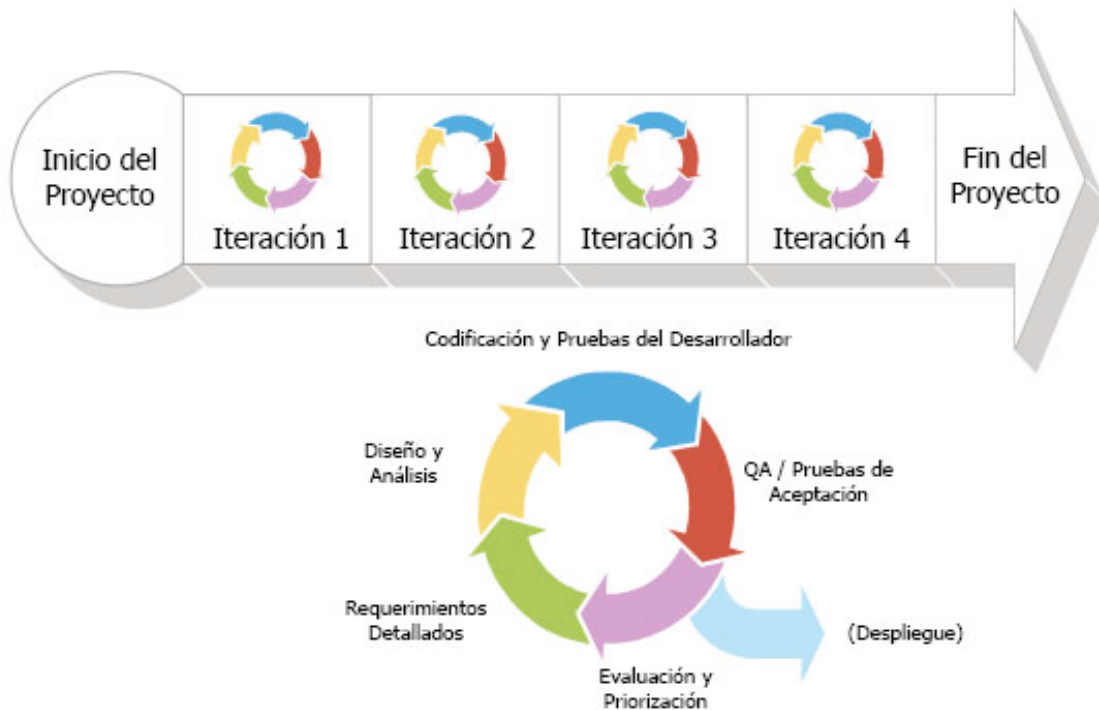


Figura 3 - Ciclo SCRUM

En este caso, al tratarse de un proyecto llevado a cabo por una sola persona, todos los roles serán interpretados por el propio autor y la tutora de este.

Es un proceso iterativo e incremental donde cada iteración debe tener como salida software funcional y de valor para el cliente, permitiendo ver el avance en el proyecto.

Esto difiere del ciclo de vida en cascada, en el que las fases del ciclo de vida (requisitos, análisis, diseño, etc.) se realizan (en teoría) una única vez, y el inicio de una fase no comienza hasta que termina la fase que le precede.

Los Sprints (iteraciones) de este TFG se explican y documentan detalladamente en el apartado 4 (Análisis) y en el apartado 6 (Implementación).

Lo que se va a implementar en cada Sprint, la funcionalidad del prototipo proviene de la Pila del Producto (Product Backlog), que contiene un conjunto de items, normalmente requisitos funcionales también llamados en scrum historias de usuario.

Una vez seleccionadas las historias de usuario que se van a desarrollar en cada Sprint, el equipo técnico las descompone en tareas, las cuales forman la Pila del Sprint (Sprint Backlog), que será inamovible durante el Sprint.

- **Pila del producto (product backlog):** lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.
- **Pila del sprint (sprint backlog):** lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- **Incremento:** resultado de cada sprint.

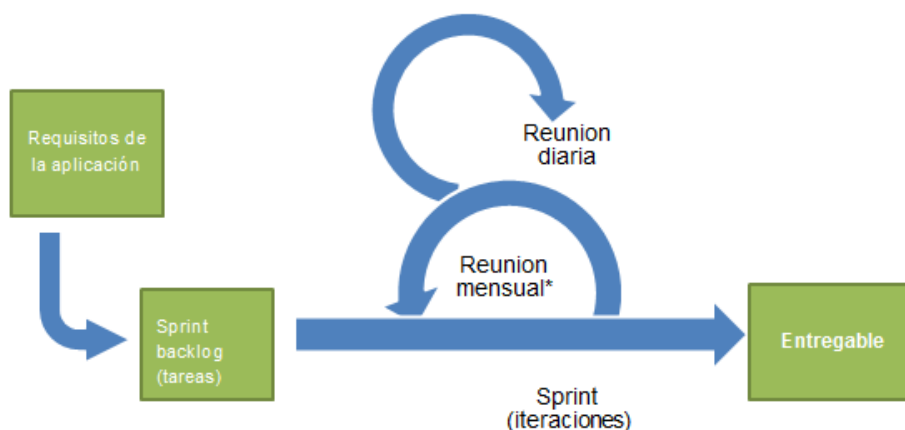


Figura 4 – Sprint Backlog Scrum

El segundo pilar más importante de Scrum son las reuniones (el primero es el ciclo de vida iterativo e incremental, los Sprints). Su importancia reside en que las reuniones son la base para lograr

transparencia y comunicación, y posibilitan algo característico en un equipo ágil: que sea autogestionado y multifuncional.

En este caso las reuniones serán con la tutora del TFG al principio de cada Sprint, para decidir que se va a realizar en ese Sprint.

Así pues, analizados los aspectos importantes de Scrum, se pueden definir cuáles son los principales beneficios que ofrece la metodología:

- a. **Cumplimiento de expectativas.** Durante cada demostración se comprueba que efectivamente se van cumpliendo sus expectativas.
- b. **Flexibilidad a cambios.** La metodología está diseñada para adaptarse a los cambios de requisitos del proyecto.
- c. **Reducción del Time to Market.** Se pueden empezar a utilizar las funcionalidades más básicas incluso antes de que el proyecto esté finalizado.
- d. **Mayor calidad del software.** El trabajo metódico, iterativo e incremental, y la necesidad de obtener una versión funcional al final de cada iteración, ayuda a que el software que se obtiene sea de una calidad superior.
- e. **Mayor productividad.** La eliminación de burocracia, con equipos pequeños y trabajo concurrente, repercute en menor tiempo y gasto dedicado al proyecto.
- f. **Reducción de riesgos.** Al llevar a cabo inicialmente las funcionalidades de mayor valor y conocer la velocidad con la que se avanza en el proyecto, se pueden despejar riesgos eficazmente antes de que estos ocurran.

En base a esta metodología para este proyecto se ha decidido hacer Sprints de una duración de cuatro semanas con sus consiguientes reuniones. Para el desarrollo completo del TFG, se han estimado necesarias seis iteraciones (seis Sprints) explicadas al detalle en el apartado seis.

A continuación, se muestran las **fases** en las que se va a organizar el desarrollo del TFG de una manera más general:

1. Estudio del problema. Lo primero es conocer la complejidad y magnitud del problema, así como las necesidades reales y objetivos para satisfacerlas.

Para ello se considera la existencia de aplicaciones similares, el conocimiento de la tecnología necesaria para llevarlo a cabo o aspectos relacionados con el campo de la IPC³.

2. Análisis y decisiones iniciales. Se realiza un análisis de todo lo recopilado en el estudio previo con el fin de determinar qué tipo de software o herramientas se van a necesitar. Tras dicho análisis se establecerán objetivos y requisitos necesarios para el desarrollo del TFG.

3. Definición de estructura y diseño. Se documentarán los aspectos clave que guiarán el posterior desarrollo, como son la estructura principal y bocetos del funcionamiento de la aplicación web, los componentes o implementaciones a usar.

4. Formación. Una vez establecidos y documentados los objetivos, se llevará a cabo el proceso de formación en torno a las herramientas a utilizar. Se debe tener en cuenta que el autor del TFG no ha desarrollado de forma individual una página web tan completa, con uso de técnicas Responsive Design⁴, Servlets, JSP o Javascript a un nivel avanzado por lo que la curva de aprendizaje inicial llevará tiempo.

5. Desarrollo Software. Una vez analizado y comprendido el problema en su conjunto, se inicia el desarrollo software que seguirá una metodología de desarrollo iterativa e incremental de 6 Sprints, centrada en el usuario y que será la que más tiempo lleve.

6. Prueba y Documentación del TFG. Por último, realizaremos pruebas de funcionamiento y respuesta del sistema descrito. Y no menos importante, al compás del desarrollo software se irá documentando todo el proceso, recopilando todos los documentos necesarios y añadiendo algunos puntos de interés en esta última fase del TFG.

3.2. HERRAMIENTAS UTILIZADAS

Durante todo el proceso de trabajo hemos utilizado multitud de herramientas. Desde las herramientas software más básicas de ofimática,

³ **IPC** (Interacción Persona Computador): Disciplina relacionada con el diseño, implementación y evaluación de sistemas informáticos interactivos.

⁴ **Responsive Design:** El diseño web responsivo o adaptativo es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos.

hasta páginas con descargas o información que podrían ser de interés para los lectores, pasando por navegadores, frameworks⁵, etc.

A continuación, se detallan las principales características de cada una de ellas:



Typecast es una aplicación que permite trabajar con más de 5.000 fuentes en un solo panel de control. El objetivo es poder decidir la tipografía que utilizarás en los textos que publicarás en la web de forma sencilla y rápida.



Sublime text es un editor de código multiplataforma, ligero y con pocas concesiones a las florituras. Es una herramienta concebida para programar sin distracciones y bastante intuitiva.



Front-end ⁶ es el lado del cliente. Los frontends son los encargados de maquetar la estructura semántica del contenido (HTML), codificar el diseño en hojas de estilo

(CSS) y agregar la interacción con el usuario (Javascript).

Con **HTML5** es posible hacer geolocalización, dibujo vectorial, guardar datos en el disco del usuario, insertar audio y video, entre otras cosas.

Con **CSS3**, se pueden crear diseños altamente complejos sin la necesidad de imágenes cortadas, sólo usando código. Bordes redondeados, sombras, degradados, fondos múltiples, entre otros.

Por último, **Javascript** (lenguaje de script portable de lado del cliente) y sus Frameworks añaden el componente de interactividad y conexión al servidor.

⁵ **Framework**. Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

⁶ **Frontend** son todas aquellas tecnologías que corren del lado del cliente, es decir, todas aquellas tecnologías que corren del lado del navegador web, generalizándose más que nada en tres lenguajes, Html, CSS Y JavaScript.



Bootstrap es un framework para HTML5 para el diseño responsivo o adaptativo y desarrollo de páginas web en HTML, CSS y JavaScript. Bootstrap es un framework muy utilizado en la actualidad por los desarrolladores de front-end. El aumento del consumo de contenidos en dispositivos móviles es uno de los motivos fundamentales en la expansión de este framework.

Back-end⁷ es el lado del servidor. Usa herramientas como editores de código, compiladores, algunos debuggeadores para revisar errores y seguridad, gestores de bases de datos y algunas otras cosas.



Un programador tiende a ser backend, y su labor se centra en el acceso a bases de datos y generación de plantillas del lado del servidor.

En backend se encargan de implementar cosas como **Apache Derby** (es la que usaremos en nuestro proyecto), Postgres, SQL Server o MongoDB. Apache Derby es un sistema gestor de base de datos relacional escrito en Java.



El **servlet** es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java (pequeños programas que se ejecutan en el contexto de un navegador web) que se ejecutan en servidores en vez de navegadores web.

El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.



La tecnología **JSP** se crea especialmente para producir contenido dinámico que sólo cambia en los valores de datos y no en la estructura básica. Una página JSP es un documento que contiene una plantilla de texto fija, junto a unas etiquetas específicas para incluir texto dinámico o ejecutar la lógica embebida en el documento. Puesto que es un lenguaje de servidor, una página JSP es siempre servida al cliente como código HTML tradicional.

⁷ El Back-End es el área que se dedica a la parte lógica de un sitio web, es la parte de atrás que de alguna manera no es visible para el usuario ya que no se trata de diseño, o elementos gráficos, se trata de programar las funciones que tendrá un sitio.



Tomcat es un contenedor web que implementa solo partes de J2EE, notablemente las especificaciones de servlets y JSP. Si se necesita utilizar Enterprise Java Beans, se necesita un servidor de aplicaciones. Si solo se van a usar servlets y JSP, solo se necesita un contenedor web.



Netbeans IDE es un entorno de desarrollo gratuito y de código abierto que permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones web, o para dispositivos móviles.

Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario.



Adobe Acrobat es una familia de programas o aplicaciones informáticas desarrollados por Adobe Systems, diseñados para visualizar, crear y modificar archivos con el formato Portable Document Format, más conocido como PDF.



Para todo lo comentado se han estado usando tres navegadores diferentes (**Safari, Google Chrome y Mozilla Firefox**).

El primero para hacer pruebas desde el dispositivo móvil, el segundo para navegar por internet, depurar código y probar estilos. Y el último para comprobar compatibilidades.



A lo largo del proceso se han utilizado tres sistemas operativos distintos, **iOS** (versión 11) en el dispositivo móvil (iPhone 7), **Android** en una tablet Samsung Galaxy

Tab4 y **Windows 10** en su última versión en el ordenador personal (Lenovo U31-70).



Microsoft Office 2016 es una Suite ofimática que recopila aplicaciones informáticas para ser utilizadas en oficinas realizando diferentes funciones sobre archivos y documentos, como crear, modificar, organizar, escanear, imprimir, entre otros.

En este trabajo se ha utilizado el Word 2016 para la memoria y el PowerPoint para la presentación.



StarUML es una herramienta para el modelamiento de software basado en los estándares UML⁸. Con él se han realizado los diagramas de secuencia del TFG en el que nos encontramos.



Adobe Photoshop es un editor de gráficos rasterizados desarrollado por Adobe Systems Incorporated. Usado principalmente para la creación de bocetos e imágenes del proyecto.

⁸ El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

4 . ANÁLISIS

La ingeniería del software es el proceso formal de desarrollo de software en el que las necesidades del usuario se traducen en requisitos (apartado 4. Análisis), estos requisitos se transforman en diseño (apartado 5. Diseño), que se implementa en código y finalmente se prueba (apartado 6. Desarrollo).

En este apartado trataremos todo lo relacionado con el análisis. Se empieza realizando la primera reunión con el cliente, con el objetivo de conocer sus necesidades y definir los requisitos.

El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables y sin ambigüedades o contradicciones.

Scrum, emplea dos formatos para registrar los requisitos:

- Product Backlog (Pila del producto)
- Sprint Backlog (Pila del sprint)

La **pila del producto** registra los requisitos vistos desde el punto de vista del **cliente**. Un enfoque similar al de los requisitos del sistema de la ingeniería de software tradicional. Está formada por la lista de funcionalidades o **historias de usuario** que desea obtener el cliente, ordenadas por la prioridad que el mismo da a cada una.

La **pila del sprint** refleja los requisitos vistos desde el punto de vista del **equipo de desarrollo**. Está formada por la **lista de tareas** en las que se descomponen las historias de usuario que se van a llevar a cabo en el sprint.

4.1. PRODUCT BACKLOG

El Product Backlog es una lista de requisitos funcionales y no funcionales priorizados por su valor para el cliente. Los requisitos que en

ella aparecen y el orden de estos es cambiante a lo largo de la vida del proyecto.

Construir y mantener el Product Backlog es mucho más simple que hacer una captura de requisitos tradicional. El motivo es simple, solo hay que expresar a grandes rasgos en que consiste el requisito, no es necesario un nivel de detalle muy elevado. Se pospone la explicación detallada de cada uno hasta el momento de la planificación del sprint correspondiente.

Utilizando el Product Backlog, cada vez que el cliente cambia sus prioridades o el alcance de un requisito ve de manera explícita y visual (se trata de una lista ordenada) que otros requisitos ven afectada su prioridad, de manera que serán implementados más tarde o incluso quedarán fuera del alcance del proyecto si este tiene precio o fecha cerrados. En el caso de que algo quede fuera del ámbito del proyecto, serán aquellos requisitos que menor valor tienen para el cliente.

A los requisitos en las metodologías ágiles se les llama **historias de usuarios** (User Stories) aunque son descripciones más cortas y simples. Están escritas en un lenguaje coloquial al ser, simplemente, el recordatorio de la conversación con el cliente. Una buena historia de usuario sigue el modelo INVEST: Independiente, Negociable, Valiosa, Estimable, Pequeña y Testeable.

A continuación, se muestran las principales historias de usuario de este TFG ordenadas por prioridad. Todas ellas serán detalladas en el apartado siguiente 4.1.1.

PRODUCT BACKLOG			
USER STORY	NOMBRE	PRIORIDAD	ESFUERZO
US-001	Conseguir una interfaz amigable e intuitiva	Alta	Alto
US-002	Realizar Test Online de Materias Comunes	Alta	Alto
US-003	Realizar Test Online de Ámbito Organizativo	Alta	Medio
US-004	Realizar Test Online de Tecnología Básica	Alta	Medio
US-005	Realizar Test Online de Seguridad	Alta	Medio

US-006	Realizar Test Online de Comunicaciones y Redes	Alta	Medio
US-007	Realizar Test Online de Simulación de examen	Alta	Alto
US-008	Cargar datos dinámicamente de una BBDD	Alta	Medio
US-009	Leer resumen con las novedades más importantes	Alta	Medio
US-010	Ampliar información de las novedades	Media	Bajo
US-011	Acceso a exámenes y soluciones de convocatorias anteriores	Media	Bajo
US-012	Ver cada apartado con sus temas y test correspondientes	Media	Bajo
US-013	Ver información relevante	Baja	Bajo
US-014	Acceso a diferentes elementos del footer	Baja	Bajo
US-015	Visitar Redes Sociales de Técnicos de Soporte Informático	Baja	Bajo
US-016	Visitar página personal del Autor de la aplicación web	Baja	Bajo
US-017	Diseño adaptado a todo tipo de dispositivos	Baja	Medio

Tabla 1 - Product Backlog

4.1.1. Historias de Usuario

En este apartado, para una mejor organización, se muestra información detallada de cada una de las historias de usuario que aparecen en la tabla superior.

Dentro de las historias de usuario tenemos varios campos necesarios para describirlas de una manera adecuada. Típicamente se describen unas características básicas para cada una: título, identificador, prioridad, estimación de las horas necesarias y la definición con la plantilla Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [descripción de la consecuencia].

El último de los campos son los llamados criterios de validación o de aceptación, que son las pruebas que el código debe superar para dar como finalizada la implementación de la historia de usuario.

Estos criterios de aceptación se escriben con la técnica del comportamiento por escenarios propia de **BDD** (Behavior Driven Development) y **gherkin**, un lenguaje específico creado especialmente para las descripciones de comportamiento de software. Dichos criterios de aceptación han de cumplir en lo máximo posible los criterios conocidos como SMART⁹.

En la figura inferior podemos ver la plantilla seguida (formato propuesto por Mike Cohn) para representar las historias de usuario de nuestro proyecto:

# - ID USER STORY	Identificador único	# - TÍTULO	Título descriptivo		
		# - PRIORIDAD	Para determinar el orden de implementación	# - ESTIMACIÓN	Estimación en tiempo ideal de implementación
# - DEFINICIÓN					
Como...			Rol del usuario interesado en la acción		
Quiero...			Acción que queremos que ocurra		
Para...			El beneficio que queremos lograr (opcional)		
# - CRITERIO DE ACEPTACIÓN					
Número			Título del escenario		
Dado que			Precondición / Descripción		
Cuando			Acción que el usuario ejecuta		
Entonces			Comportamiento del sistema		

Tabla 2 - Plantilla Historias de Usuario

La sintaxis de los criterios de aceptación con gherkin, como podemos comprobar en la plantilla de la parte superior, es la siguiente:

(Scenario)	Escenario [Número de escenario] [Título del escenario]
(Given)	Dado que [Contexto] y adicionalmente [Contexto]
(When)	Cuando [Evento]
(Then)	Entonces el sistema [Resultado / Comportamiento]

⁹ SMART: S - Specific (Específicos), M - Measurable (Medibles), A - Achievable (Alcanzables), R - Relevant (Relevantes) y T - Time-boxed (Limitados en el tiempo).

US-001	TÍTULO Conseguir una interfaz amigable e intuitiva
	PRIORIDAD ALTA ESTIMACIÓN 40h
DEFINICIÓN	
Como...	Usuario
Quiero...	Conseguir navegar por una interfaz visualmente atractiva e intuitiva
Para...	Interactuar de una forma fácil y exitosa
CRITERIO DE ACEPTACIÓN	
1	Navegación satisfactoria
Dado que	Visito la aplicación web
Cuando	Quiero navegar entre los diferentes apartados
Entonces	Encuentro rápido todo y entiendo como navegar por la aplicación web fácilmente

Tabla 3 – Historia de Usuario 1

US-002	TÍTULO Realizar Test Online de Materias Comunes
	PRIORIDAD ALTA ESTIMACIÓN 40h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder realizar test de forma interactiva
Para...	Preparar los temas correspondientes a materias comunes
CRITERIO DE ACEPTACIÓN	
1	Realización del Test
Dado que	Estoy en un test de materias comunes
Cuando	Relleno todos los datos y doy a corregir
Entonces	El test se da por concluido y se corrige automáticamente mostrando los resultados

Tabla 4 – Historia de Usuario 2

US-003	TÍTULO Realizar Test Online de Ámbito Organizativo
	PRIORIDAD ALTA ESTIMACIÓN 25h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder realizar test de forma interactiva
Para...	Preparar los temas correspondientes a ámbito organizativo
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en un test de ámbito organizativo
Cuando	Relleno todos los datos y doy a corregir
Entonces	El test se da por concluido y se corrige automáticamente mostrando los resultados

Tabla 5 - Historia de usuario 3

US-004	TÍTULO Realizar Test Online de Tecnología Básica
	PRIORIDAD ALTA ESTIMACIÓN 30h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder realizar test de forma interactiva
Para...	Preparar los temas correspondientes a tecnología básica
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en un test de tecnología básica
Cuando	Relleno todos los datos y doy a corregir
Entonces	El test se da por concluido y se corrige automáticamente mostrando los resultados

Tabla 6 - Historia de usuario 4

US-005	TÍTULO	Realizar Test Online de Seguridad
	PRIORIDAD	ALTA
	ESTIMACIÓN	25h
DEFINICIÓN		
Como...	Usuario	
Quiero...	Poder realizar test de forma interactiva	
Para...	Preparar los temas correspondientes a seguridad	
CRITERIO DE ACEPTACIÓN		
1	1	
Dado que	Estoy en un test de seguridad	
Cuando	Relleno todos los datos y doy a corregir	
Entonces	El test se da por concluido y se corrige automáticamente mostrando los resultados	

Tabla 7 - Historia de usuario 5

US-006	TÍTULO	Realizar Test Online de Comunicaciones y Redes
	PRIORIDAD	ALTA
	ESTIMACIÓN	25h
DEFINICIÓN		
Como...	Usuario	
Quiero...	Poder realizar test de forma interactiva	
Para...	Preparar los temas correspondientes a comunicaciones y redes	
CRITERIO DE ACEPTACIÓN		
1	1	
Dado que	Estoy en un test de comunicaciones y redes	
Cuando	Relleno todos los datos y doy a corregir	
Entonces	El test se da por concluido y se corrige automáticamente mostrando los resultados	

Tabla 8 - Historia de usuario 6

US-007	TÍTULO Realizar Test Online de Simulación de examen
	PRIORIDAD ALTA ESTIMACIÓN 40h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder realizar test de forma interactiva
Para...	Preparar los temas correspondientes a simulación de examen
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en un test de simulación de examen
Cuando	Relleno todos los datos y doy a corregir
Entonces	El test se da por concluido y se corrige automáticamente mostrando los resultados

Tabla 9 - Historia de usuario 7

US-008	TÍTULO Cargar datos dinámicamente de una BBDD
	PRIORIDAD ALTA ESTIMACIÓN 40h
DEFINICIÓN	
Como...	Usuario
Quiero...	Cargar datos desde una BBDD
Para...	Poder realizar los test cargando preguntas de forma dinámica
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en una página para test online
Cuando	Se carga la página y aparecen los datos
Entonces	Se cargan 20 preguntas fijas para los test normales, y 55 preguntas (50 de ellas aleatorias) para los test de examen.

Tabla 10 - Historia de usuario 8

US-009	TÍTULO Leer resumen con las novedades más importantes
	PRIORIDAD ALTA ESTIMACIÓN 15h
DEFINICIÓN	
Como...	Usuario
Quiero...	Leer las últimas novedades de una forma rápida
Para...	Ahorrarme el tiempo que tardaría en comprobar el JCYL, BOE y otras páginas.
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el apartado de novedades
Cuando	Miro la información
Entonces	De un vistazo pueden leerse las novedades más importantes, ordenadas por fecha en un orden descendente

Tabla 11 - Historia de usuario 9

US-010	TÍTULO Ampliar información de las novedades
	PRIORIDAD MEDIA ESTIMACIÓN 5h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder ampliar la información con las últimas novedades
Para...	Enterarme de todos los detalles en caso de estar interesado
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el apartado de novedades
Cuando	Amplio la información
Entonces	Se abre la fuente que ha servido de base para realizar el resumen y que contiene toda la información más detallada

Tabla 12 - Historia de usuario 10

US-011	TÍTULO Acceso a exámenes y soluciones de convocatorias anteriores
	PRIORIDAD MEDIA ESTIMACIÓN 5h
DEFINICIÓN	
Como...	Usuario
Quiero...	Tener rápido acceso a exámenes de convocatorias anteriores
Para...	Estar mejor preparado una vez que llegue la fecha del examen
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el apartado de exámenes
Cuando	Abro un examen
Entonces	Se abre un documento con el examen o respuestas correspondientes

Tabla 13 - Historia de usuario 11

US-012	TÍTULO Ver cada apartado con sus temas y test correspondientes
	PRIORIDAD MEDIA ESTIMACIÓN 5h
DEFINICIÓN	
Como...	Usuario
Quiero...	Tener acceso a todos los test online organizados por temas
Para...	Preparar el test correspondiente al tema que se esté estudiando
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el menú de los distintos apartados
Cuando	Abro un apartado cualquiera
Entonces	Se abre una página con los temas y test correspondientes a ese apartado

Tabla 14 - Historia de usuario 12

US-013	TÍTULO Ver información relevante de la oposición
	PRIORIDAD BAJA ESTIMACIÓN 15h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder enterarme de información relevante de la oposición
Para...	Estar al tanto de cómo se realiza el proceso selectivo, cómo funciona el pago de sueldos, requisitos, etc.
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en la página principal en el apartado ventajas
Cuando	Hago clic en el apartado de Información que interesa
Entonces	Se abre una nueva página con toda la información.

Tabla 15 - Historia de usuario 13

US-014	TÍTULO Acceso a diferentes elementos del footer
	PRIORIDAD BAJA ESTIMACIÓN 10h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder acceder a diferentes elementos del footer
Para...	Ver sobre nosotros, el Mapa del sitio web, Leer los términos y condiciones, las preguntas frecuentes o la privacidad.
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el pie de página o footer
Cuando	Entro en cada uno de los enlaces
Entonces	Se muestra la información correspondiente

Tabla 16 - Historia de usuario 14

US-015	TÍTULO Visitar Redes Sociales de Técnicos de Soporte Informático
	PRIORIDAD BAJA ESTIMACIÓN 1h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder acceder a diferentes redes sociales relacionadas
Para...	Interactuar con otros estudiantes de la oposición
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el pie de página o footer
Cuando	Entro en los enlaces de redes sociales
Entonces	Se muestra la información correspondiente

Tabla 17 - Historia de usuario 15

US-016	TÍTULO Visitar página personal del Autor de la aplicación web
	PRIORIDAD BAJA ESTIMACIÓN 1h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder acceder a la página personal del autor del sitio web
Para...	Conocer al autor, sus trabajos o contactar con él
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	Estoy en el pie de página o footer
Cuando	Entro en el enlace a su página personal
Entonces	Se muestran la página del autor del proyecto

Tabla 18 - Historia de usuario 16

US-017	TÍTULO Diseño adaptado a todo tipo de dispositivos
	PRIORIDAD BAJA ESTIMACIÓN 25h
DEFINICIÓN	
Como...	Usuario
Quiero...	Poder acceder a la aplicación web desde cualquier tipo de dispositivo
Para...	Poder realizar test incluso cuando no se tenga acceso a un ordenador
CRITERIO DE ACEPTACIÓN	
1	1
Dado que	No dispongo de mi ordenador, pero sí de mi teléfono móvil
Cuando	Entro en la página web del presente proyecto
Entonces	Se muestro la página adaptada al tamaño del dispositivo

Tabla 19 - Historia de usuario 17

4.2. SPRINT BACKLOG

La pila del sprint descompone las historias de usuario en unidades llamadas tareas, con un tamaño adecuado para monitorizar el avance a diario e identificar riesgos sin necesidad de procesos de gestión complejos.

En la tabla inferior se encuentra la tabla Sprint Backlog de nuestro TFG, con las tareas necesarias para construir las historias de usuario que se van a realizar en cada una de las iteraciones.

SPRINT BACKLOG				
SPRINT	TAREA	DESCRIPCIÓN	HORAS ESTIMADAS	ESTADO
1	T-001	Documentación estructura y bloques I y II	30h	Completada
	T-002	Configurar el entorno de desarrollo.	4h	Completada
	T-003	Diseño de imagen de fondo	2h	Completada
	T-004	Diseño imágenes temarios	8h	Completada
	T-005	Especificar historias de usuario y criterios de aceptación.	6h	Completada

SPRINT	TAREA	DESCRIPCIÓN	HORAS ESTIMADAS	ESTADO
	T-006	Crear modelo de dominio.	2h	Completada
	T-007	Plantear la arquitectura software.	2h	Completada
2	T-008	Documentación Bloque II	10h	Completada
	T-009	Bocetos para la interfaz de usuario.	10h	Completada
	T-010	Establecer presupuesto.	5h	Completada
	T-011	División en tareas independientes.	10h	Completada
	T-012	Calendarización.	10h	Completada
	T-013	Elaborar plan de riesgos.	5h	Completada
3	T-014	Documentación Bloque II (continuación)	5h	Completada
	T-015	Estructura de la página principal.	5h	Completada
	T-016	Implementar menú barra de navegación	5h	Completada
	T-017	Implementar sección Ventajas	5h	Completada
	T-018	Implementar diseño test online	5h	Completada
	T-019	Implementar mecanismo responder preguntas	5h	Completada
	T-020	Implementar porcentaje de aciertos	5h	Completada
	T-021	Implementar colores en las preguntas	5h	Completada
	T-022	Implementar preguntas falladas / no contestadas	10h	Completada
	T-023	Implementar corregir test	10h	Completada
	T-024	Implementar ver nota de examen	5h	Completada
	T-025	Implementar timeline de novedades	20h	Completada
	T-026	Implementar apartado exámenes	5h	Completada
	T-027	Test de usabilidad	5h	Completada
4	T-028	Documentación Bloque II (continuación)	5h	Completada
	T-029	Implementar página inicio temario	4h	Completada
	T-030	Implementar página materias comunes	10h	Completada
	T-031	Implementar página ámbito organizativo	5h	Completada
	T-032	Implementar página tecnología básica	5h	Completada
	T-033	Implementar página seguridad	4h	Completada
	T-034	Implementar página comunicaciones y redes	4h	Completada
	T-035	Implementar página simulación de examen	6h	Completada
	T-036	Conectar BBDD e introducir datos	30h	En progreso

	T-037	Detalles de análisis y diseño	10h	Completada
	T-038	Implementar sección pie de página o footer	2h	Completada
	T-039	Implementar About	2h	Completada
	T-040	Implementar Sitemap	2h	Completada
	T-041	Implementar FAQ	2h	Completada
	T-042	Implementar Términos y condiciones	2h	Completada
	T-043	Implementar Privacidad	2h	Completada
	T-044	Primeras pruebas y mejoras	5h	Completada
5	T-045	Documentación Bloque II (continuación)	5h	Completada
	T-046	Diseño responsivo css media queries	10h	En progreso
	T-047	Pruebas y mejoras correspondientes	20h	Completada
6	T-048	Documentación final	15h	Completada
	T-049	Pruebas / Test de usabilidad	5h	Completada
	T-050	Mejoras finales y ajustes visuales	10h	Completada

Tabla 20 - Tabla tareas por cada iteración

Como podemos observar, las iteraciones o sprints 3 y 4 son los que más tiempo nos han llevado, pues es en los que se ha centrado la parte más complicada del desarrollo.

Si sumamos el tiempo estimado en realizar las tareas en un entorno de desarrollo ideal para la iteración 1 sería de unas 50 horas, la iteración 2 unas 50 horas (durante las dos primeras iteraciones se compagina el TFG con las prácticas de empresa), la iteración 3 unas 95 horas, la iteración 4 en unas 100h, la quinta en unas 35h y la sexta en 30h.

Todo ello suma un total de 360 horas, que en el caso de nuestro título "Título de Graduado en Ingeniería Informática de Servicios y Aplicaciones" donde la carga ECTS de la asignatura TFG es de 12 ECTS, se ajusta correctamente a la horquilla de 300 a 360 horas aconsejadas.

4.3. PLAN DE RIESGOS

Es prácticamente imposible abordar todos y cada uno de los riesgos existentes en un proyecto, sin mencionar lo costoso que sería en términos de tiempo y recursos. Por esta razón, existen herramientas de priorización, tales como la **matriz de probabilidad e impacto de riesgos**.

La matriz de probabilidad e impacto es una herramienta de análisis cualitativo de riesgos que te permite establecer prioridades en cuanto a los posibles riesgos de un proyecto en función tanto de la probabilidad de que ocurran como del impacto o repercusiones que podrían tener en caso de que ocurrieran.

El resultado de esta matriz es una calificación global para cada uno de los riesgos identificados en el proyecto. A partir de esta calificación es posible tomar decisiones con relación a la cantidad de tiempo y atención que requerirá cada riesgo.

Como se puede observar en la figura inferior, un riesgo con una alta probabilidad de que ocurra y un alto impacto demandaría la preparación de un plan de respuesta, mientras que los riesgos con baja probabilidad y bajo impacto podrían requerir sólo supervisión o monitoreo superficial.

Impacto	Alto	Considerar	Planificar Respuesta	Planificar Respuesta
	Medio	Desatender pero monitorizar	Considerar	Planificar Respuesta
	Bajo	Desatender pero monitorizar	Desatender pero monitorizar	Considerar
		Baja	Media	Alta
		Probabilidad		

Figura 5 – Matriz de probabilidad e impacto de riesgos

Para proyectos de mayor envergadura que el aquí tratado, podríamos refinar más la matriz organizando una lluvia de ideas identificando los riesgos y colocándolos en la cuadrícula (NxN) mayor.

Para gestionar los riesgos, se establecen un plan de mitigación y un plan de contingencia:

- **Plan de mitigación:** Se trata de lo que en el PMBOK ¹⁰ se denomina Estrategia de Respuesta a los Riesgos, es decir, las acciones que intentan reducir la probabilidad de ocurrencia de un riesgo o el impacto en caso de que se produzca, por tanto, su objetivo es reducir la exposición al riesgo.
- **Plan de contingencia:** Se trata de las respuestas que se dan cuando efectivamente se produce un riesgo. Por lo tanto, se puede decir que el plan de contingencia es un plan reactivo que se ocupa de los eventos que pueden o no ocurrir.

A continuación, se plantean algunos de los posibles riesgos que pueden surgir a lo largo del desarrollo del proyecto, así como los planes de contingencia y mitigación para cada uno de ellos.

RISK-001	Riesgo de retraso en la planificación
Probabilidad	Alta
Impacto	Alto
Descripción	Pueden producirse retrasos debido a una mala estimación de la calendarización o de una mala planificación, lo que causaría no finalizar el proyecto en las fechas previstas.
Gestión del riesgo	
Estrategia	Evitar el riesgo.
Plan de mitigación	Se ha realizado la planificación con días a mayores en cada iteración, para en caso de que se produzcan retrasos, finalizar el proyecto a tiempo.
Plan de contingencia	Incrementar el trabajo diario.

Tabla 21 - Riesgo 001

¹⁰ PMBOK: La Guía de los fundamentos para la dirección de proyectos (del inglés Guide to the Project Management Body of Knowledge o PMBOK por sus siglas) es un libro en el que se presentan estándares, pautas y normas para la gestión de proyectos.

RISK-002	Riesgo por falta de conocimiento de las tecnologías empleadas
Probabilidad	Alta
Impacto	Alto
Descripción	Algunas de las tecnologías usadas en el proyecto son desconocidas como Servlets de Java, páginas JSP o Javascript a alto nivel.
Gestión del riesgo	
Estrategia	Prevenir el riesgo.
Plan de mitigación	Realizar pruebas por medio de tutoriales y documentación existente antes y durante del proyecto.
Plan de contingencia	Pedir ayuda sobre posibles problemas que puedan surgir.

Tabla 22 - Riesgo 002

RISK-003	Riesgo por problemas en el diseño
Probabilidad	Baja
Impacto	Medio
Descripción	Un mal diseño de la aplicación podría generar problemas en la implementación.
Gestión del riesgo	
Estrategia	Protegerse del riesgo.
Plan de mitigación	Dar importancia suficiente a las fases de diseño.
Plan de contingencia	Replantear el diseño.

Tabla 23 - Riesgo 003

RISK-004	Riesgo por modificaciones en los requisitos
Probabilidad	Media
Impacto	Medio
Descripción	Posible modificación de los requisitos, es decir, de las historias de usuario una vez comenzado el proyecto.
Gestión del riesgo	
Estrategia	Protegerse del riesgo.
Plan de mitigación	Empleo de metodología ágil con buena adaptación a cambios en los requisitos.
Plan de contingencia	Realizar cambios en los requisitos.

Tabla 24 - Riesgo 004

RISK-005	Riesgo por problemas con los recursos software y hardware
Probabilidad	Baja
Impacto	Bajo
Descripción	Un fallo en un recurso software o hardware podría provocar retrasos en la entrega del proyecto.
Gestión del riesgo	
Estrategia	Protegerse del riesgo.
Plan de mitigación	Puesta a punto de los equipos antes del comienzo del proyecto, labores de mantenimiento y realización de copias de seguridad periódicamente.
Plan de contingencia	Reparar recurso o sustituirlo por otro.

Tabla 25 - Riesgo 005

RISK-006	Riesgo debido a una mala elección en la arquitectura del proyecto
Probabilidad	Baja
Impacto	Medio
Descripción	Una mala elección en la arquitectura del proyecto conllevaría un aumento en los costes de desarrollo.
Gestión del riesgo	
Estrategia	Prevenir el riesgo.
Plan de mitigación	Planear la arquitectura minuciosamente e investigar desde el inicio si es la idónea para nuestro proyecto.
Plan de contingencia	Decidir si se asume el sobrecoste y se continua con la misma arquitectura o se replantea una nueva.

Tabla 26 - Riesgo 006

RISK-007	Riesgo debido a la falta de experiencia del equipo de desarrollo
Probabilidad	Media
Impacto	Medio
Descripción	La falta de experiencia de los recursos humanos del equipo de desarrollo puede suponer retrasos y sobrecostes.
Gestión del riesgo	
Estrategia	Prevenir el riesgo.
Plan de mitigación	Incluir tiempo de formación a la hora de realizar la planificación del proyecto.
Plan de contingencia	Aumentar los tiempos o recursos económicos destinados a formación.

Tabla 27 - Riesgo 007

RISK-008	Riesgo de baja laboral en algún miembro del equipo de desarrollo
Probabilidad	Baja
Impacto	Alto
Descripción	En este proyecto realizado por una sola persona, la baja por enfermedad o cualquier otra causa que le impidiese realizar sus funciones correctamente conllevaría retrasos con la fecha de entrega.
Gestión del riesgo	
Estrategia	Aceptar el riesgo.
Plan de mitigación	Ya que la probabilidad es baja, y sería difícil prevenir o evitar dicho riesgo más allá de hacer una vida sana y realizar las actividades cotidianas.
Plan de contingencia	Aceptar los retrasos producidos a consecuencia de esa baja.

Tabla 28 - Riesgo 008

4.4. DIAGRAMA DE GANTT

Se trata de una herramienta cuyo objetivo es exponer de una forma gráfica el tiempo de dedicación previsto para las diferentes tareas de nuestro TFG a lo largo de un tiempo determinado, en nuestro caso de septiembre de 2017 a marzo de 2018.

En el apartado 4.2 Sprint Backlog se obtienen de las historias de usuario 50 tareas (de menor tamaño) para distribuir en el tiempo y en cada iteración. Con dichas tareas se ha realizado un Diagrama de Gantt para ver la evolución de estas durante el tiempo que ha tenido de duración el proyecto.

Tarea	Sprint	Fecha de inicio	Fecha final	Duración (días)
Tarea 1	1	17/09/2017	05/10/2017	18
Tarea 2		25/09/2017	27/09/2017	2
Tarea 3		27/09/2017	30/09/2017	3
Tarea 4		01/10/2017	05/10/2017	4
Tarea 5		06/10/2017	10/10/2017	4
Tarea 6		11/10/2017	13/10/2017	2
Tarea 7		14/10/2017	18/10/2017	4
Tarea 8	2	19/10/2017	25/10/2017	6
Tarea 9		26/10/2017	02/11/2017	7
Tarea 10		03/11/2017	05/11/2017	2
Tarea 11		05/11/2017	14/11/2017	9
Tarea 12		15/11/2017	19/11/2017	4
Tarea 13		20/11/2017	22/11/2017	2
Tarea 14	3	23/11/2017	26/11/2017	3
Tarea 15		26/11/2017	05/12/2017	9
Tarea 16		27/11/2017	29/11/2017	2
Tarea 17		29/11/2017	30/11/2017	1
Tarea 18		30/11/2017	01/12/2017	1
Tarea 19		01/12/2017	03/12/2017	2
Tarea 20		03/12/2017	05/12/2017	2
Tarea 21		05/12/2017	07/12/2017	2
Tarea 22		08/12/2017	10/12/2017	2
Tarea 23		11/12/2017	13/12/2017	2
Tarea 24		13/12/2017	15/12/2017	2
Tarea 25		12/12/2017	23/12/2017	11
Tarea 26		26/12/2017	27/12/2017	1
Tarea 27		26/12/2017	31/12/2017	5
Tarea 28	4	03/01/2018	06/01/2018	3
Tarea 29		07/01/2018	09/01/2018	2
Tarea 30		10/01/2018	11/01/2018	1
Tarea 31		11/01/2018	12/01/2018	1
Tarea 32		12/01/2018	13/01/2018	1
Tarea 33		13/01/2018	14/01/2018	1
Tarea 34		14/01/2018	15/01/2018	1
Tarea 35		15/01/2018	23/01/2018	8
Tarea 36		20/01/2018	01/02/2018	12
Tarea 37		02/02/2018	04/02/2018	2
Tarea 38		04/02/2018	06/02/2018	2
Tarea 39		05/02/2018	06/02/2018	1
Tarea 40		05/02/2018	06/02/2018	1
Tarea 41		05/02/2018	07/02/2018	2
Tarea 42		06/02/2018	08/02/2018	2
Tarea 43		08/02/2018	09/02/2018	1
Tarea 44		10/02/2018	11/02/2018	1
Tarea 45		5	12/02/2018	20/02/2018
Tarea 46	21/02/2018		25/02/2018	4
Tarea 47	25/02/2018		01/03/2018	4
Tarea 48	6	01/03/2018	10/03/2018	9
Tarea 49		05/03/2018	10/03/2018	5
Tarea 50		09/03/2018	15/03/2018	6

Figura 6 – Datos diagrama de Gantt

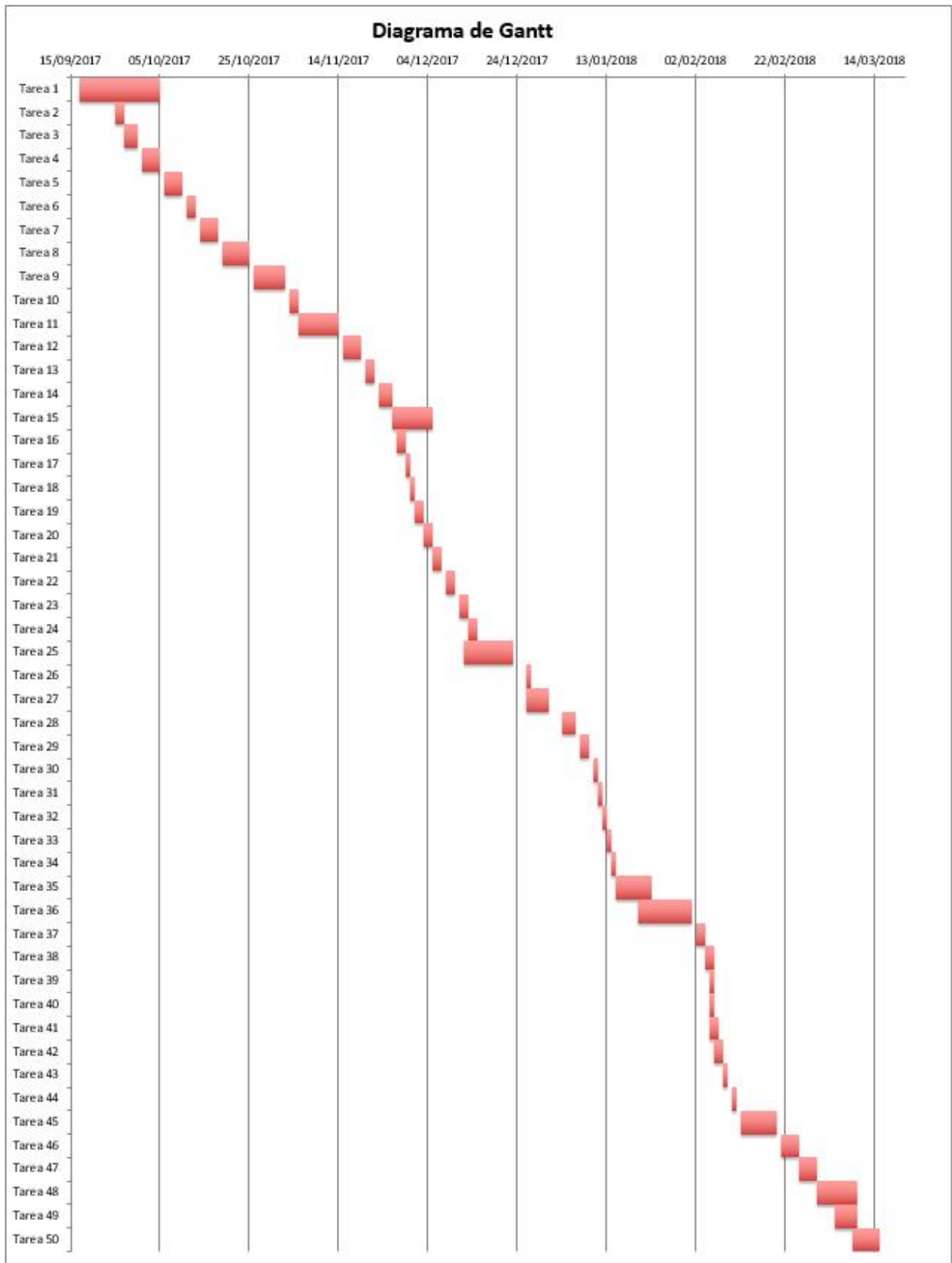


Figura 7 - Diagrama de Gantt

4.5. ESTIMACIÓN DE LOS COSTES

Hay varias propuestas de métricas para medir diversos aspectos de la productividad del software, en nuestro caso usaremos una métrica basada en volumen / unidad de tiempo (líneas / programador-mes) ya que no tendremos en cuenta factores como fiabilidad ni mantenimiento del software.

“Es imposible planificar sin estimar o estimar sin medir”

Con el fin de dar una idea aproximada sobre el coste que puede suponer el desarrollo de aplicación web, se lleva a cabo una estimación de los costes totales en base a los costes de hardware, costes de software, costes de recursos humanos y costes de oficina.

Costes de hardware:

Nuestro proyecto tendrá una duración estimada de unos 6 meses, así pues, contaremos la amortización de la parte proporcional a la vida útil de cada producto.

Respecto al material necesario, se ha requerido de la utilización de un ordenador portátil, en este caso un Lenovo U31-70 con el que poder desarrollar la aplicación y transportarlo fácilmente valorado en unos 700€. También se ha optado por el uso de un monitor externo marca LG de 32 pulgadas para usarlo como doble pantalla y mejorar la productividad valorado en 360€.

La aplicación web está optimizada para todo tipo de dispositivos, por ello usamos para las pruebas un teléfono móvil modelo Apple iPhone 7 32Gb de 4,6 pulgadas y una Tablet Samsung Galaxy Tab 4 de 10,1 pulgadas.

Producto	Precio	Vida útil	Amortización
Lenovo U31-70	700,00€	5 años	70,00€
Teléfono móvil	639,00€	3 años	106,50€
Monitor LG 34"	360,00€	10 años	18,00€

Tablet	299,00€	5 años	29,90€
TOTAL	1998,00€		224,40€

Tabla 29 - Costes Hardware

Costes de software:

A parte del hardware, también se requiere de cierto software para poder diseñar y desarrollar la aplicación. La mayoría lo ofrecen gratuitamente las empresas propietarias, pero hay algunas licencias que hay que pagar.

En el caso de las de los productos de Adobe se han calculado proporcionalmente a los precios de las licencias mensuales.

Producto	Precio
Windows 10	145,00€
Microsoft Office	149,00€
Adobe Standard DC	94,00€
Adobe Photoshop	217,00€
Sublime Text	80,00€
StarUML	49,00€
TOTAL	734,00€

Tabla 30 - Costes Software

Costes de oficina:

Sumamos al coste de oficina el correspondiente gasto en material de oficina, el coste de impresión, luz eléctrica y el coste proporcional del servicio proveedor de internet.

También se incluye, el coste de contratar un servidor y un dominio durante 6 meses. Todo ello valorado en unos 600€.

Costes de recursos humanos:

Normalmente no solo los jefes de proyecto realizan estimaciones para el software, pues a cada desarrollador le asignan unas tareas y se le pregunta cuánto tiempo le llevará completarlo. Para este TFG, al realizarlo una sola persona, nos encargaremos de las estimaciones de cada uno de los roles.

Este gasto se ha calculado en función del rol del ejercicio y de las horas de trabajo necesarias.

Rol	Precio por hora
Jefe de Proyecto [JP]	50,00€
Diseñador gráfico [DG]	30,00€
Ingeniero Software [IS]	40,00€
Tester [T]	30,00€
Documentador [D]	30,00€

Tabla 31 - Roles para costes humanos

A continuación, se muestran los costes de recursos humanos según las tareas que se han estimado en cada uno de los sprints:

Tarea	Rol	Horas previstas	Coste
Análisis	IS	50h	2.000,00€
Diseño	IS, DG	30h	2.100,00€
Desarrollo		180h	8.200,00€
Iteración 1			
Reunión Cliente	JP, IS	4h	360,00€
Implementación	IS	20h	800,00€

Iteración 2			
Reunión Cliente	JP, IS	4h	360,00€
Implementación	IS	20h	800,00€
Iteración 3			
Reunión Cliente	JP, IS	4h	360,00€
Implementación	IS	35h	1400,00€
Iteración 4			
Reunión Cliente	JP, IS	4h	360,00€
Implementación	IS	35h	1400,00€
Iteración 5			
Reunión Cliente	JP, IS	2h	180,00€
Implementación	IS	30h	1200,00€
Iteración 6			
Reunión Cliente	JP, IS	2h	180,00€
Implementación	IS	20h	800,00€
Pruebas	T	30h	900,00€
Documentación	D	70h	2.100,00€
TOTAL		360h	15.300,00€

Tabla 32 - Costes humanos

La fase de análisis está estimada en unas 50h, la fase de diseño en 30h, la de implementación en 180h, la parte correspondiente a las pruebas 30 horas y la documentación en unas 70 horas.

Por lo tanto, el total de la estimación de los sueldos a pagar de los trabajadores asciende a 15.300,00€.

Costes totales:

Después de detallar anteriormente los diferentes costes del proyecto se ha calculado el coste total en la siguiente tabla:

Rol	Coste
Costes Hardware	224,40€
Costes Software	734,00€
Costes de Oficina	600,00€
Costes de Recursos Humanos	15.300,00€
COSTE TOTAL	16.858,40€

Tabla 33 - Costes totales estimados

Se trata de una estimación, por lo que el coste real puede variar. Dicho coste aparecerá en el apartado de resultados una vez se sepan los gastos y tiempos finales.

Se ha de tener en cuenta que los resultados estimados del coste total serían correctos si el proyecto se encargase de manera profesional a una empresa para un desarrollo completo que involucrase a los diferentes perfiles de personas especializadas en su materia como diseñadores, analistas o jefe de proyecto con el fin de remunerar su trabajo y proporcionar todos los materiales y herramientas que necesitasen.

Se trata en realidad de un Trabajo Fin de Grado con una duración de 6 meses realizado por una sola persona, la cual no ha cobrado nada por ello.

5 . DISEÑO

En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software.

De acuerdo con las decisiones tomadas en las fases anteriores, se razona la forma de realizar la solución al problema.

Para empezar, la topología que debe seguirse corresponde con un modelo distribuido pues existen ordenadores clientes que acceden a un servidor web en distintas partes del mundo mediante un navegador.

5.1. MODELO ARQUITECTÓNICO

La arquitectura de esta aplicación web Java es de tipo cliente-servidor con una arquitectura de tres capas que nos permite dividir la funcionalidad entre las distintas capas (servidor de datos, servidor de web o de aplicaciones y cliente).

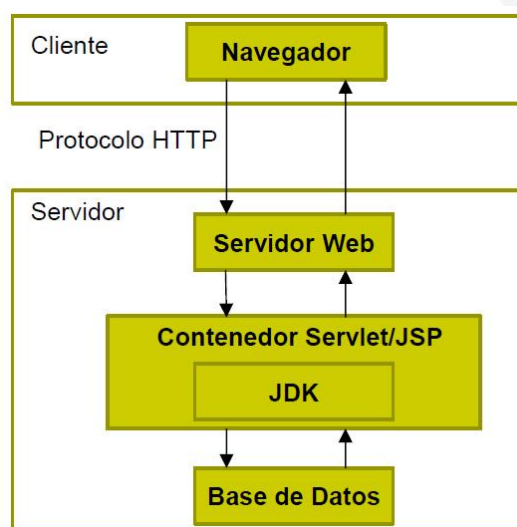


Figura 8 - Modelo Arquitectónico

En el diseño del proyecto se ha seguido el patrón arquitectónico MVC, explicado en detalle en el apartado 5.2.2.

5.1.1. Arquitectura en Capas

Se ha utilizado un diseño en tres capas:

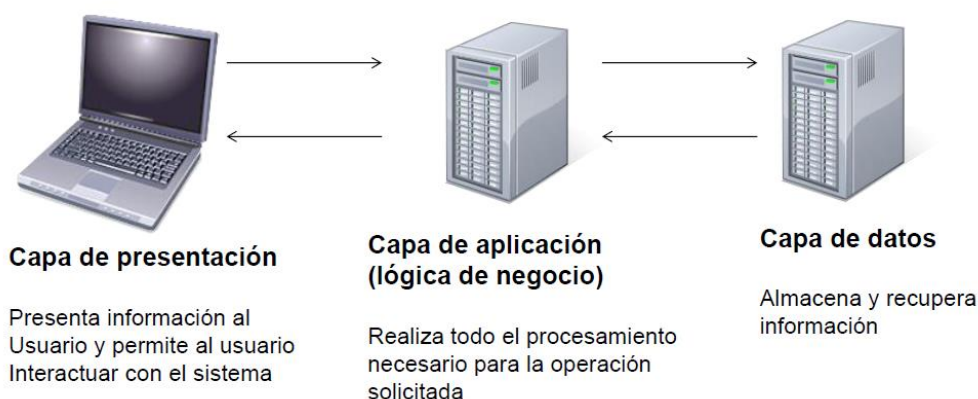


Figura 9 - Esquema arquitectura 3 capas

- **Capa de presentación:** la capa de presentación o interfaz gráfica es la capa con la que el usuario se comunica con la aplicación. Recoge y valida la información enviada por los usuarios y presenta los datos con el formato adecuado.

Las tecnologías implicadas en esta capa son HTML, CSS y JavaScript. Por una parte, HTML ofrece la estructura de la aplicación web, CSS aplica los estilos necesarios para hacer de la aplicación un entorno “amigable”, mientras que JavaScript es el encargado de realizar las operaciones necesarias en el navegador.

(Las peticiones AJAX se situarían entre la capa de presentación y la de negocio).

- **Capa de negocio:** Esta capa recibe las peticiones del usuario y envía las respuestas tras procesarlas. JSP ¹¹ y Servlets son las tecnologías implicadas en esta capa y proporcionadas por nuestro contenedor web Apache Tomcat que implementa

¹¹ **JavaServer Pages** (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

solo partes de J2EE, proporcionando un conjunto de funciones capaces de comunicarse con la capa de datos y la capa de presentación.

- **Capa de datos:** En esta capa se almacenan los datos, se reciben las peticiones de información y se devuelven con la información seleccionada. La tecnología empleada en esta capa es la base de datos Apache Derby, y las acciones realizadas en la base de datos se harán mediante el lenguaje SQL.

Los términos "capas" y "niveles" no significan lo mismo, el término "capa" hace referencia a la forma en que una solución es segmentada desde el punto de vista lógico (Presentación, negocio y datos) y el término "nivel" corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física (diferentes ordenadores).

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores.

5.1.2 Patrón MVC

Actualmente, en el desarrollo de software orientado a la Web, es imprescindible la utilización de patrones de diseño que permitan el mejoramiento de la calidad del software otorgando a las aplicaciones características que les permitan ser fácilmente mantenibles, flexibles y evolutivas. Uno de los patrones básicos para el desarrollo de aplicaciones orientadas a la Web es el patrón modelo vista controlador (MVC).

Este patrón separa la interfaz de los componentes de la abstracción interna de los datos y utiliza controladores que notifican del cambio de estado. La utilización de este patrón surge por la necesidad de independizar la representación externa de los datos de su tipo interno, y puesto que ambos conceptos son susceptibles de cambiar con frecuencia, se intentará que una variación en uno de ellos tenga una mínima repercusión en el otro.

Fue desarrollado a finales de los años 70 por Trygve M.H. Reenskaug, quien trabajaba en el proyecto SmallTalk de Xerox. El patrón MVC está compuesto por los siguientes elementos:

- **La vista:** Es la Interfaz gráfica, es con lo que el usuario de la aplicación interactúa y es donde se reciben las “órdenes” y se presentan los resultados de los procesos al usuario.
- **El controlador:** Es el modificador del modelo, es decir, es el que modifica los valores de las variables, objetos, datos en general, lo hace de acuerdo con lo solicitado por el usuario a través de la interfaz gráfica (vista).
- **El modelo:** Es la representación específica de la información con la que se está operando en el instante de la ejecución de la aplicación, representa las variables, objetos, datos en general que se están modificando de acuerdo con lo que el usuario solicite.

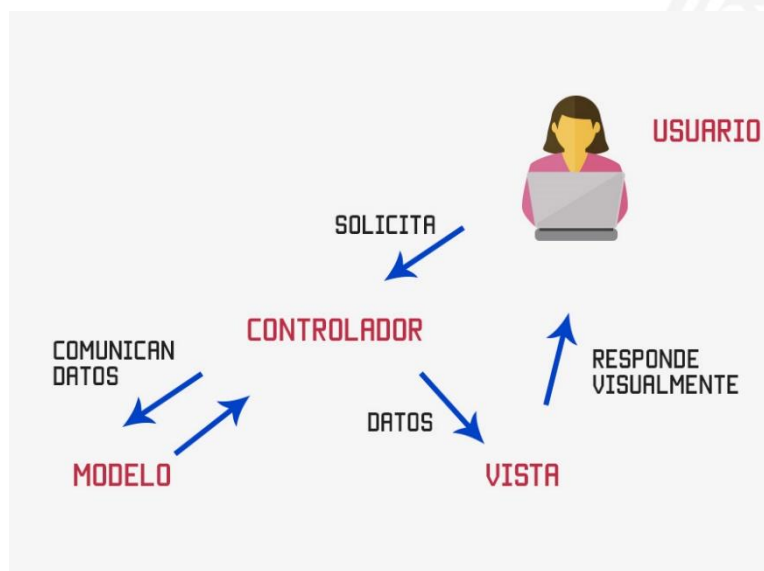


Figura 10 - Esquema MVC

Cuando se desarrolla una aplicación Web, surge la pregunta ¿Qué capa pertenece al cliente y qué capa pertenece al servidor? Con el “boom” de las nuevas tecnologías la respuesta puede ser un poco más compleja, pues el patrón MVC ha tenido que acomodarse no solo a la arquitectura Cliente-Servidor sino al paradigma de orientación a objetos.

Se han desarrollado multitud de frameworks que implementan este patrón MVC, estos se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

Los primeros frameworks planteaban un enfoque de cliente ligero en el que tanto el modelo como el controlador (y buena parte de la vista) están completamente alojados en el servidor. Actualmente existen

frameworks como JavaScriptMVC, Backbone o jQuery14 que permiten que ciertos componentes MVC se ejecuten parcial o totalmente en el cliente (véase AJAX).

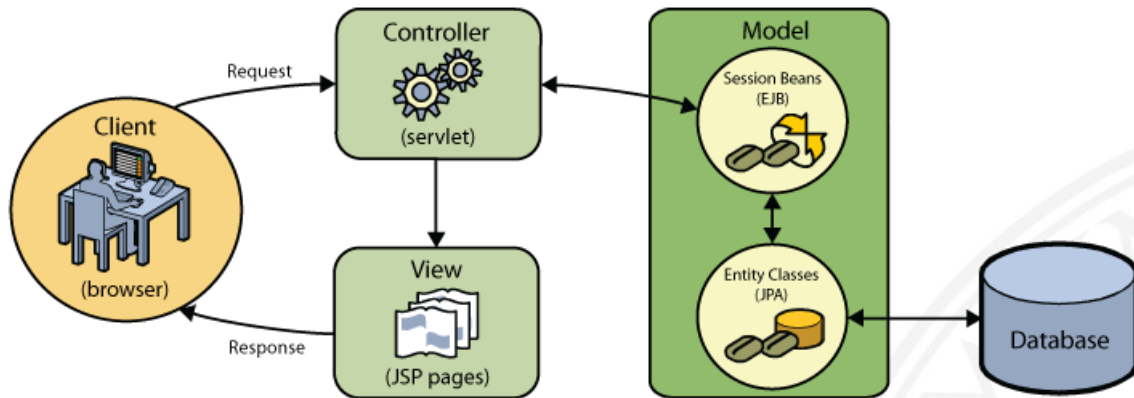


Figura 11 – Diagrama resumen del funcionamiento de la aplicación

Este diagrama nos muestra de forma simplificada cómo funciona el **patrón MVC en la aplicación**, aunque en nuestro caso dada la simplicidad del proyecto, no usaremos EJB. Los Servlets hacen la función de controlador, las clases de Servicio hacen la función de modelo y, por último, los ficheros JSP se encargan de la Vista. El resultado final de todas estas operaciones es generar un documento HTML que el navegador presenta.

La fuente de datos también es otro factor para tener en cuenta en el diseño de una aplicación. En este proyecto los datos se almacenan en una base de datos relacional, Apache Derby.

5.2. MODELO DEL DOMINIO

A la hora de diseñar el modelo del dominio se decidió trabajar sólo con una entidad, en este caso 'Pregunta'. Al comienzo del TFG se planteó la posibilidad de hacerlo con dos, 'pregunta tema' y 'pregunta examen', pero daba lugar a algunos problemas que eran fácilmente evitables trabajando sólo con una entidad.

Estos problemas eran:

- A la hora de diseñar la base de datos nos encontraríamos con 2 entidades representadas cada una en una tabla y toda la lógica implementada para construir las páginas donde el usuario ve los test tendría que ampliarse para tratar objetos de dos tipos.

- Derivado del punto anterior también habría que modificar algunos de los métodos de los servlets, e incluso tenerles por duplicado para que trabajasen con diferentes objetos.

La solución adoptada finalmente ha sido trabajar con una tabla. Aunque presenta el inconveniente de que para las preguntas de tipo Examen Final (con valor 'EX' en el atributo sección) algunos campos son 'null', es la decisión elegida porque nos permite unificar la lógica para cualquier tipo de test.

La tabla se muestra en la figura siguiente:

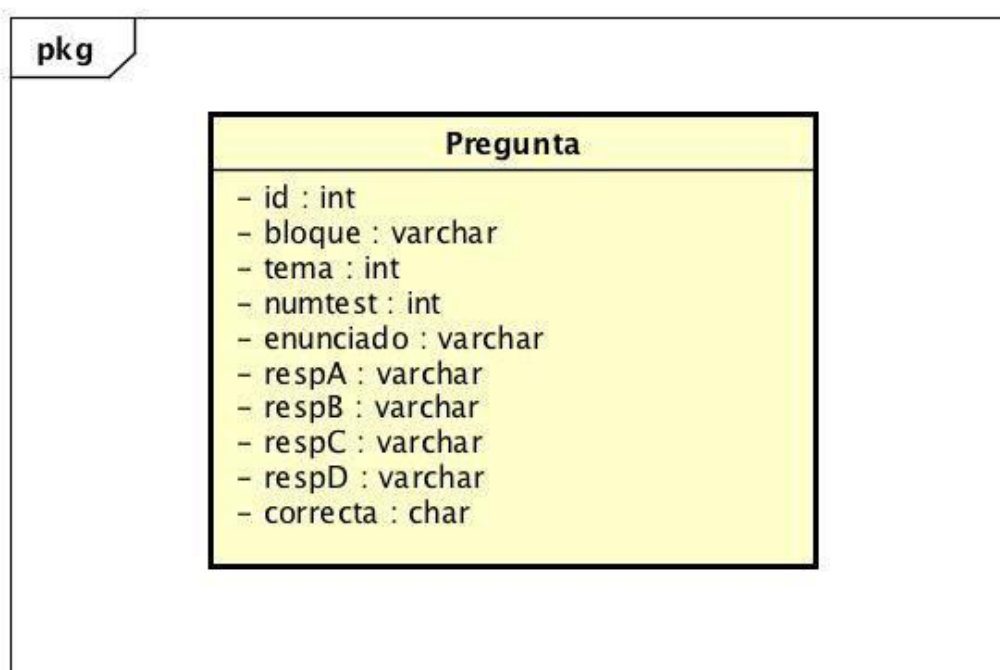


Figura 12 - Modelo de dominio

5.3. MODELO DE COMPORTAMIENTO

En esta sección se incluye un diagrama de secuencia, que permite reflejar el comportamiento de la aplicación en cuanto al flujo de operaciones.

El diagrama de secuencia contiene detalles de implementación, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos a través del tiempo.

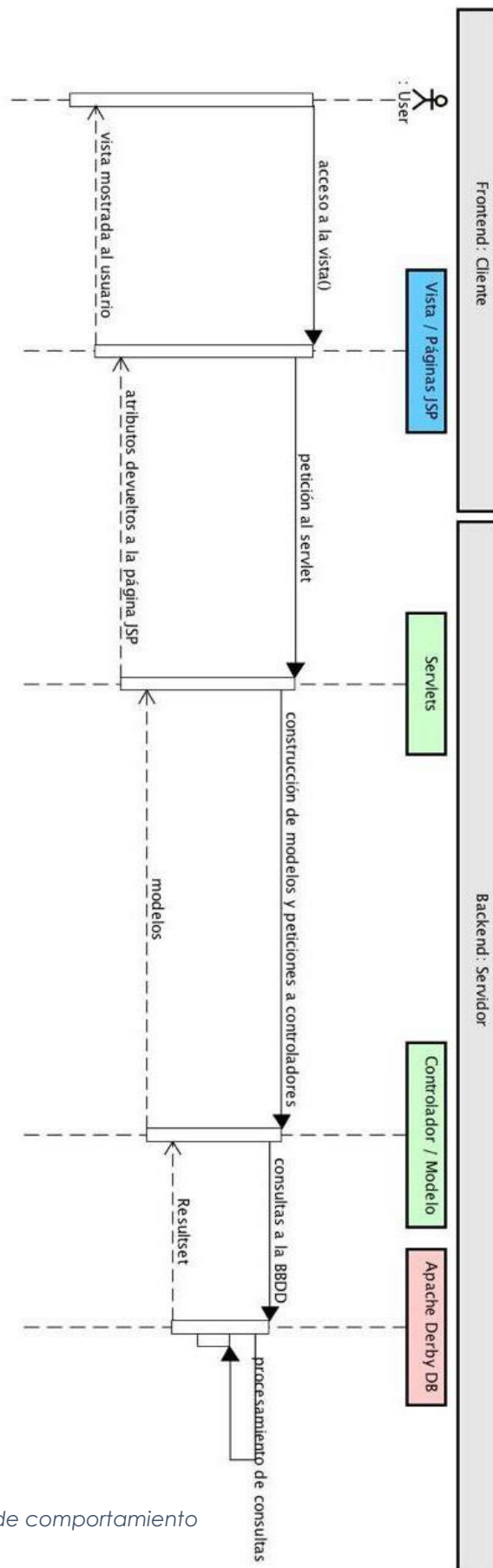


Figura 13 - Modelo de comportamiento

5.4. DIAGRAMA DE DESPLIEGUE

En la figura siguiente se muestra el modelo de despliegue de la aplicación a lo largo de tres dispositivos: el cliente, el servidor web y el servidor de base de datos.

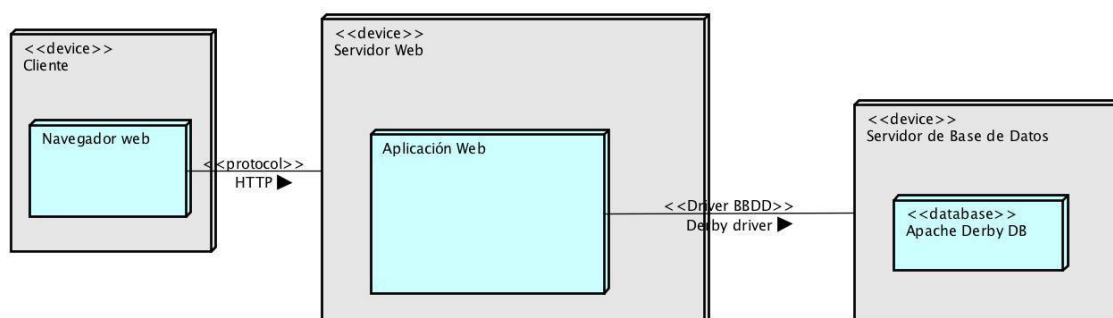


Figura 14 - Modelo de despliegue

5.5. DICCIONARIO DE DATOS

El diccionario de datos es un listado organizado de todos los datos que pertenecen a un sistema. El objetivo es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades.

La aplicación se centra en la tabla "PREGUNTA, ya que se trata de una aplicación para hacer exámenes online tipo test. Tiene una serie de campos que describiremos a continuación:

- **ID**, es el identificador único de una pregunta, no puede haber dos iguales. Es la llave primaria.
- **BLOQUE**, es el tipo de bloque al que pertenece cada pregunta. Los únicos valores posibles son MC de materias comunes, AO de ámbito organizativo, TB de tecnología básica, SE de seguridad, CR de comunicaciones y redes y EX de examen.
- **TEMA**, es el número de tema correspondientes al temario de la oposición. Puede estar vacío porque si el bloque es examen no tendrá número de tema.
- **NUMTEST**, es el número de Test dentro del tema ya que, aunque de momento solo se ha introducido un test de cada tema en la base de datos, con el tiempo ese número irá creciendo y habrá varios test para preparar cada tema.

- **ENUNCIADO**, es el enunciado de la pregunta.
- **RESPA**, es el texto correspondiente al apartado A de la pregunta.
- **RESPB**, es el texto correspondiente al apartado B de la pregunta.
- **RESPC**, es el texto correspondiente al apartado C de la pregunta.
- **RESPD**, es el texto correspondiente al apartado D de la pregunta.
- **CORRECTA**, indica cuál es la respuesta correcta. Obviamente los valores solo pueden estar entre A, B, C y D.

Tabla PREGUNTA				
Campo	Tipo	Tamaño	Restricciones	PK / FK
ID	Integer	6	Not Null, Autoincrement	PK
BLOQUE	Varchar	2	Not Null, Enum{'MC','AO','TB','SE','CR','EX'}	-
TEMA	Integer	3	-	-
NUMTEST	Integer	3	-	-
ENUNCIADO	Varchar	500	Not null	-
RESPA	Varchar	500	Not null	-
RESPB	Varchar	500	Not null	-
RESPC	Varchar	500	Not null	-
RESPD	Varchar	500	Not null	-
CORRECTA	Char	1	Not null, Enum {'A','B','C','D'}	-

Tabla 34 - Tabla Pregunta DB

5.6. DISEÑO CENTRADO EN EL USUARIO

De nada sirve hacer una aplicación si luego nadie la utiliza, así que se le ha prestado especial atención al usuario. Para el desarrollo del trabajo se han tenido en cuenta tres **aspectos fundamentales** del Diseño Centrado en el Usuario (DCU):

- a. Primero, el foco temprano en usuarios y tareas (análisis de usuario, de tareas, dominio, contexto).
- b. Segundo, utilizar los análisis para dibujar un boceto y realizar el primer prototipo.
- c. Tercero, evaluar constantemente a lo largo del proceso utilizando para ello el test de usabilidad.

También se han tenido en cuenta algunos **principios y guías de diseño**, empleados en la aplicación web:

- Lo primero intentar hacer una UI atractiva para el usuario, pues la página tiene que atraer al usuario. Se han usado colores vivos, una página principal que llame la atención, iconos modernos, etc.
- Hemos intentado hacerlo muy simple (decir no a añadir más opciones) e intuitivo sobre todo para las tareas que más se utilizan (rutas rojas o rutas más importantes). Esto va ligado a la eficiencia y la satisfacción del usuario y llevaría consigo el uso diario de la aplicación que es el objetivo final.
- Fácil de entender por medio de colores y mantener la misma estructura (consistencia) en las distintas partes de la aplicación.
- Más imágenes y menos texto para ir al grano y no distraer a los usuarios.
- Esquema de color monocromático (la mayoría de los menús, fondos, test, y diferentes partes son blancos, rojos y negros).
- Misma estructura apareciendo juntas las características que están relacionadas para mejorar la eficiencia pues recorres menos distancia entre unas y otras.
- Tamaño de los botones y textos importantes un poco más grandes (Ley de Fitts). No sólo lo que ocupe el texto. Metiendo padding en los botones de los test, por ejemplo. O pudiendo marcar el radiobutton seleccionando en cualquier parte del apartado de la pregunta.
- Color de los botones como Resultados o Fallos con colores vivos que llamen la atención.

- Tratamiento de errores intentando siempre implementar si se puede la opción atrás o deshacer. Dejando la barra de navegación fija o incluyendo un índice o migas de pan.
- Uso de las esquinas de la aplicación (menús, atrás).
- Uso correcto del lenguaje, palabras positivas que hagan entender al usuario lo que le quieres decir y no otra cosa. Por ejemplo, en la página de error.

Con todos estos principios se realizó un boceto inicial y más adelante un boceto para cada pantalla y cada posible interacción del usuario. Una vez se daba el visto bueno a cada boceto, que hacía a su vez como prototipo, se empezaba su implementación.

5.7. BOCETOS DE LA INTERFAZ DE USUARIO

Para este proyecto se empezó dibujando en la primera reunión con la tutora un boceto de cómo podría ser esta aplicación web. Después ese boceto se digitalizó por medio del programa Adobe Photoshop y se fue evolucionando hasta conseguir un prototipo de valor.

Se han cambiado unos cuantos detalles tras los test de usabilidad y las diferentes observaciones.

El más importante es la eliminación de menús y páginas adicionales que era la idea inicial (¡algo arcaica!), ya que es más intuitivo tener una única página con todo el contenido en vertical. Es una estructura más sencilla, si bien con el aumento de futuras funcionalidades podría volver a incluirse un menú.

Además, se cambian palabras por otras más descriptivas, se fija la barra de navegación a la parte superior y se añaden anclas para navegar entre las distintas secciones de forma directa y mejorar la productividad.

En las imágenes inferiores podemos ver la evolución de los diferentes bocetos.

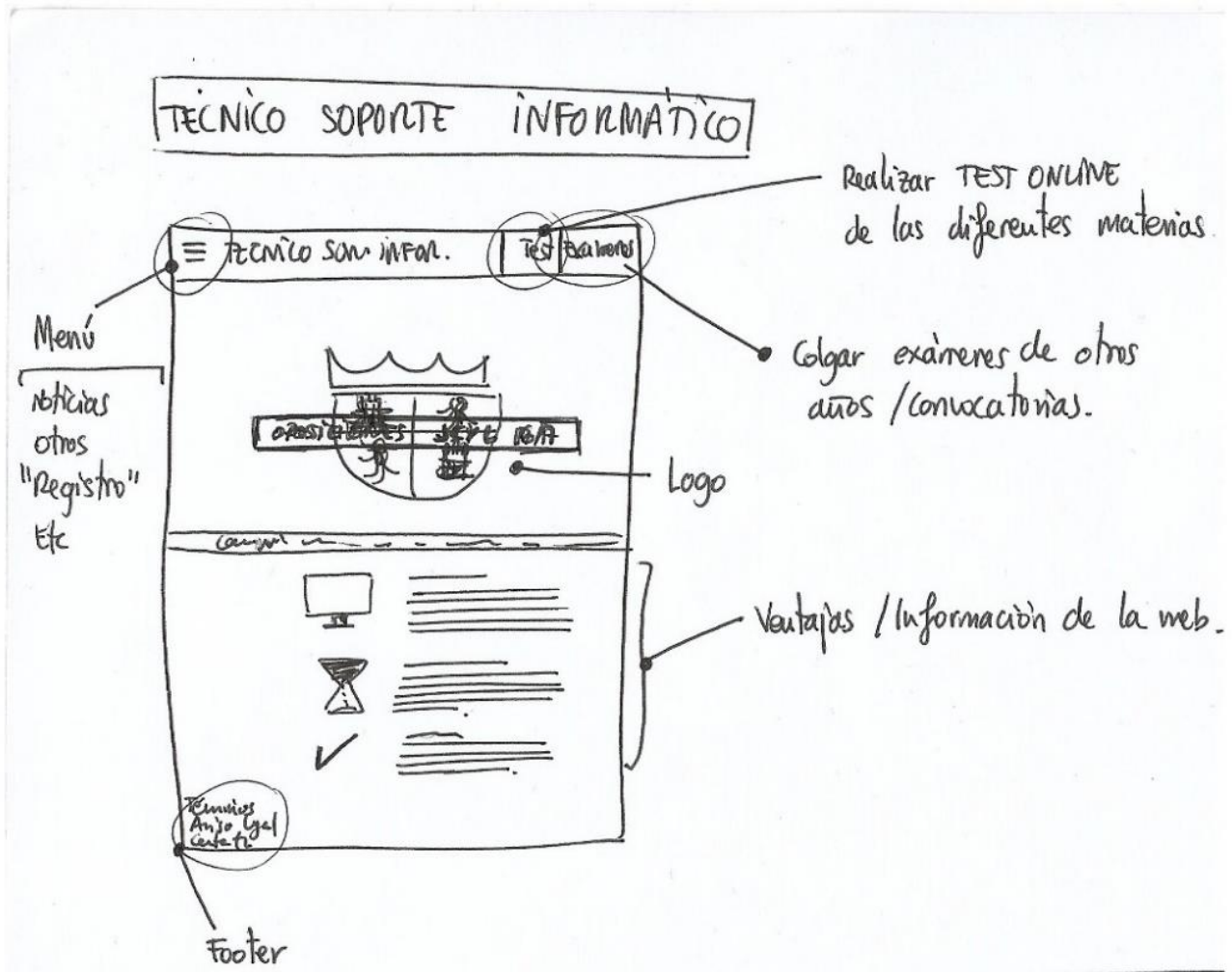


Figura 15 - Boceto pantalla principal



Figura 16 - Boceto color página principal con Photoshop

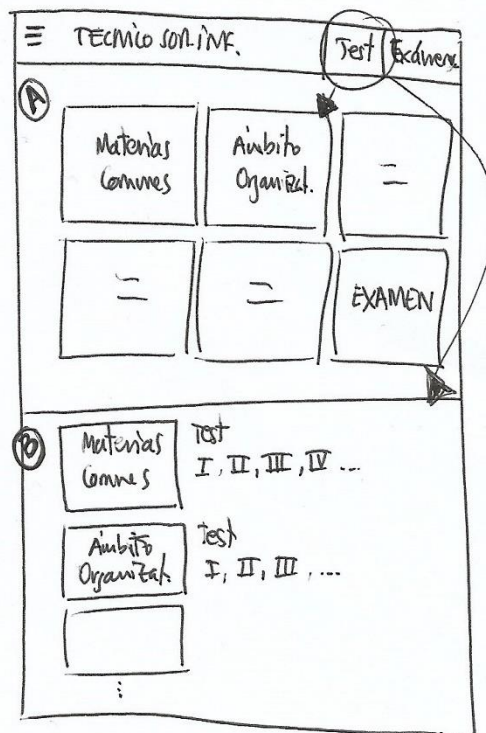


Figura 17 - Boceto A y boceto B del menú temario



Figura 18 - Boceto a color segunda versión página principal con Photoshop

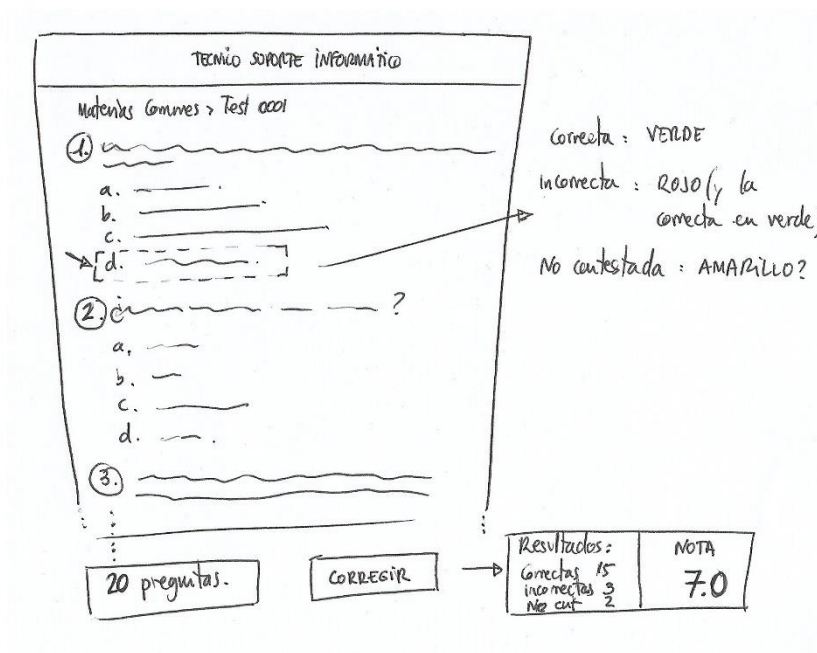


Figura 19 - Boceto inicial test de ejemplo

6 . IMPLEMENTACIÓN

Como se ha estado explicando anteriormente, Scrum es la metodología que se ha elegido para el desarrollo de este TFG. En base a esta metodología, se ha decidido hacer Sprints de una duración de unas cuatro semanas con sus consiguientes reuniones.

No se ha seguido un horario en concreto ya que la situación ha sido cambiante, ha habido días de muchas horas, otros en los que no se ha podido trabajar por falta de disponibilidad u otras obligaciones.

Para el desarrollo completo del TFG, se han realizado seis iteraciones, seis sprints, cada uno con un número de horas estimadas en la tabla inferior y un nombre que describe la tarea más importante dentro de cada uno de ellos:

Sprints	Nombre del Sprint	Horas requeridas	Producto final
Sprint 1	Inicio del TFG	50h	Boceto
Sprint 2	Planificación	50h	Boceto
Sprint 3	Desarrollo I	95h	✓
Sprint 4	Desarrollo II	100h	✓
Sprint 5	Pruebas	35h	✓
Sprint 6	Final	30h	✓
	TOTAL	360h	✓

Tabla 35 - Estimación horas por Iteración

Las dos primeras iteraciones son las fases iniciales de estudio del problema y tratan en mayor medida las decisiones iniciales, fase de análisis, formación en aspectos que se desconocen, planificación del

trabajo, diseño y bocetos a modo de prototipo representados en el apartado 5.3.

Mientras que las cuatro últimas iteraciones son las relacionadas con el desarrollo de la aplicación web en sí y sobre las que van a empezar a tomar más sentido los Sprints ya que será aquí donde la tutora vea los resultados obtenidos.

Antes de empezar con la descripción de cada iteración de una forma detallada, se muestra el reparto de las horas necesarias en cada una de ellas.

Podemos observar como las dos primeras iteraciones constan de casi la mitad de las horas que el Sprint 3 y 4. La explicación no es otra que la compaginación del presente proyecto con las prácticas de empresa.

SPRINTS	ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	TESTER	DOCUMENTACIÓN	TOTAL
1. Inicio del TFG	10h	10h	0h	0h	30h	50h
2. Planificación	30h	10h	0h	0h	10h	50h
3. Desarrollo I	5h	5h	80h	0h	5h	95h
4. Desarrollo II	5h	5h	80h	5h	5h	100h
5. Pruebas	0h	0h	10h	20h	5h	35h
6. Final	0h	0h	10h	5h	15h	30h

Tabla 36 - Desglose horas por cada Rol

6.1. ITERACIÓN 1. INICIO DEL TFG

En esta primera iteración o sprint el objetivo primordial consiste en conocer el alcance del proyecto, para lo cual se han de alcanzar los siguientes objetivos:

- Entender qué hay que construir.
- Investigar las tecnologías que van a permitir el desarrollo.

- Identificar las funcionalidades clave del sistema.
- Configurar el entorno de desarrollo.

En la primera reunión se habla de las ideas y objetivos principales, la tecnología requerida en su desarrollo, la viabilidad del proyecto, las funcionalidades principales y las futuras ampliaciones.

Los objetivos principales eran convertir dicha aplicación web en una ayuda al estudio, permitir realizar test de forma online y ver todas las novedades relativas con la oposición todo de una forma clara y agradable.

Se decide la mejor tecnología para su desarrollo, decantándose por una aplicación web implementada en html y css, frente a una aplicación móvil debido al tamaño de los dispositivos y de la finalidad de esta. Además de cubrir ese problema con el diseño responsivo para aquellos usuarios que accedan desde un dispositivo diferente.

Una vez decididas las funcionalidades principales y hecho un análisis inicial, se habla de funcionalidades futuras como ampliar la oferta de oposiciones, ampliar temario o traducción en otros idiomas y de la viabilidad del proyecto, dando luz verde al mismo.

Decidido el tema y los objetivos, se desarrolla un prototipo / boceto de la página principal con Adobe Photoshop para probar si la futura interfaz era atractiva visualmente e intuitiva. Se desarrolla hasta conseguir una respuesta positiva.

Se prosiguió con la lectura de las Guías de estilo de los Trabajos de Fin de Grado para iniciar la tarea de documentación. Completando portada, formatos, resumen, motivación, objetivos, estado del arte, metodología, herramientas y estructura general de la memoria.

6.2. ITERACIÓN 2. PLANIFICACIÓN

En esta segunda iteración los objetivos principales consisten en aprender todo lo necesario para afrontar el proyecto y establecer un plan a seguir para lograr exitosamente los objetivos, para lo cual se ha de conseguir:

- Adquirir los conocimientos necesarios para el desarrollo del TFG.
- Análisis básico y general del proyecto.
- Planificación del proyecto de una forma óptima y precisa.
- Plan de riesgos.

Se investiga acerca de los conocimientos necesarios para la iteración 3 llamada Desarrollo I. Especialmente lo relativo a diseños responsivos y a la forma de realizar los test, con tecnologías como Javascript, Html5, Servlets y las conexiones con las BBDD.

Se realiza el análisis de la aplicación web de una forma más detallada que en el apartado anterior repasando las historias de usuario y organizando el proyecto en pequeñas tareas a realizar dentro de cada sprint.

En cuanto al apartado del diseño se mejora el prototipo de la iteración 1, surgiendo del mismo dos posibilidades en cuanto a la corrección de los Test Online. La primera sería mostrar los resultados una vez finalizadas todas las preguntas del test y la segunda mostrarlos según se van contestando preguntas y tener así presente nuestro razonamiento. Finalmente se descartó esta segunda opción porque difería más del modo de examen usado en la realidad.

Dentro de esta iteración se estiman los costes que serán necesarios para concluir este proyecto de una forma satisfactoria, para más tarde una vez desarrollado, concluir si ha sido o no acertado. También se realiza un análisis de los posibles riesgos que pudieran surgir del proyecto y mecanismos para tratarlos.

Se establecen los plazos estimados de entrega (podrían variar levemente en función de la disponibilidad de la tutora) de cada iteración o sprint, cada 4 semanas aproximadamente, serían los siguientes:

- ✓ **Sprint 1** – 16 de octubre de 2017
- ✓ **Sprint 2** – 16 de noviembre de 2017
- ✓ **Sprint 3** – 18 de diciembre de 2017
- ✓ **Sprint 4** – 25 de enero de 2018
- ✓ **Sprint 5** – 22 de febrero de 2018
- ✓ **Sprint 6** – 15 de marzo de 2018

Se realiza la planificación general del Trabajo Fin de Grado, con una estimación de las horas de cada una de las fases y se decide que es lo que se va a realizar en cada una de ellas y los plazos.

En la siguiente tabla se observa el desglose semanal estimado desde el inicio del proyecto el 18 de septiembre con su Sprint o iteración correspondiente:

Semana	Fechas	Sprint
1	18 Sept. – 24 Sept.	1
2	25 Sept. – 1 Oct.	
3	2 Oct. – 8 Oct.	
4	9 Oct. – 15 Oct.	
5	16 Oct. – 22 Oct.	2
6	23 Oct. – 29 Oct.	
7	30 Oct. – 5 Nov.	
8	6 Nov. – 12 Nov.	
9	13 Nov. – 19 Nov.	3
10	20 Nov. – 26 Nov.	
11	27 Nov. – 3 Dic.	
12	4 Dic. – 10 Dic.	
13	11 Dic. – 17 Dic.	4
14	18 Dic. – 24 Dic.	
15	25 Dic. – 31 Dic.	
16	1 Ene. – 7 Ene.	
17	8 Ene. – 14 Ene.	5
18	15 Ene. – 21 Ene.	
19	22 Ene. – 28 Ene.	
20	29 Ene. – 4 Feb.	
21	5 Feb. – 11 Feb.	6
22	12 Feb. – 18 Feb.	
23	19 Feb. – 25 Feb.	
24	26 Feb. – 4 Mar.	
25	5 Mar. – 11 Mar.	Entrega
26	12 Mar. – 18 Mar.	

Tabla 37 - Calendarización

Y, por último, se documenta la memoria todo lo posible antes de iniciar la implementación o desarrollo del código en los sprints 3 y 4.

6.3. ITERACIÓN 3. DESARROLLO I

La tercera iteración comienza una vez finalizadas las prácticas de empresa y se centra básicamente en la implementación de lo expuesto en los dos Sprints anteriores.

Teniendo el análisis casi completo, la memoria bastante avanzada y los diseños o bocetos, el objetivo es convertir los prototipos en papel a la realidad.

Se empieza con la página index.html que es la página principal. Una de las ventajas de hacer el diseño antes, es que ya tienes claro cómo es la estructura de la página, cómo implementar la barra de navegación, si la UI ¹² es atractiva, etc.

Dentro de la página index.html se implementan de arriba abajo: la barra de navegación, la imagen de la web, ventajas y un apartado con información relevante, el temario, exámenes de otros años, un timeline con novedades y un footer (pie de página) con los datos de contacto, términos y demás.

La barra de navegación tiene dos formatos, para pantallas grandes en el que aparecen los tres apartados hacia donde podemos desplazarnos (test online, exámenes y novedades) y el de pantallas más pequeñas con el típico icono de menú desplegable.

Lo interesante de la barra de navegación es que los tres apartados comentados anteriormente se desplazan en vertical a través de la página principal index.html en función de donde estemos situados. No se abren otras pestañas ni pierdes de vista la información. Es un diseño más enfocado a dispositivos móviles en los que se tiende a hacer scroll, y es más cómodo e intuitivo.

La imagen de la web consta de la bandera de Castilla y León junto con las oposiciones a las que va destinada. Todas las imágenes que aparecen en la aplicación son de diseño propio.

En el apartado temario se puede acceder directamente a los test de los temas de las secciones que estés preparando, o se puede acceder a una página intermedia donde pueden observarse cada una de las secciones y los temas.

¹² UI o también llamada User Interface, se refiere a la creación de la interfaz, que puede ser gráfica o desarrollada por tecnologías como HTML, CSS, YUI, jQuery, etc.

En el apartado exámenes se pueden descargar exámenes de otros años de una forma sencilla y estando todos a la vista ordenados por año descendente.

Novedades es un apartado implementado con el objetivo de mostrar las últimas noticias relacionadas con la oposición. Había varias formas de desarrollo para este apartado, pero se eligió un Timeline con un diseño moderno e intuitivo con los ítems ordenados por fecha.

Por último, en el footer, un correo electrónico de contacto, enlaces a redes sociales, mapa del sitio web, preguntas frecuentes, términos y privacidad, aunque todos éstos están sin implementar.

Todo ello desde la base de diseño responsivo (Responsive Design), ya que actualmente hay multitud de dispositivos desde donde se consume contenido, cada uno con un tamaño y resolución.

Se empieza con el desarrollo de un test de prueba interactivo sin estilos, solo para conseguir mostrar preguntas de forma dinámica y con la conexión con la BBDD.

Continuamos con la labor de documentación, en el apartado que nos ocupa y complementando el resto de la memoria.

6.4. ITERACIÓN 4. DESARROLLO II

En este cuarto Sprint continuamos con las labores de implementación de la iteración anterior, mejorando la página principal, desarrollando la página o plantilla para la realización de test online, introduciendo información en la base de datos (BBDD) y desarrollando servlets para los test tipo examen.

Se crea la página que sirve de plantilla para mostrar los test. Se incluye un índice en la parte superior con el objetivo de ubicar al usuario en el sitio web y facilitar la navegación a diferentes partes de la aplicación. Dicha página cargará las preguntas de forma dinámica. Se explica el funcionamiento en el apartado de Apéndices.

Teniendo en cuenta la opinión de la tutora en la tercera reunión, se establece que, a la hora de cargar test de forma dinámica, el sistema lo hará de forma aleatoria. Esto nos ha hecho replantearnos un poco el sistema modificando la historia de usuario US-009 ("Cargar datos dinámicamente de una BBDD") ya que al principio se pensó en que toda la

información iba a ser estática realizando test que siempre iban a tener las mismas preguntas.

Por ejemplo, los test que simulan un examen real de 55 preguntas usarán 50 preguntas aleatorias de la base de datos de preguntas de examen y 5 preguntas ya aparecidas en los test por temas. Aunque estos números podrían variar en función del número total de preguntas introducidas en la base de datos. Cada vez que entremos en una simulación de examen, las preguntas cambiarán entre todo el banco de preguntas disponible.

También se cumple que las 10 primeras preguntas de las 55 son de materias comunes, con el fin de simular lo mejor posible el examen oficial.

El núcleo de esta iteración ha sido conseguir la forma de mostrar las preguntas de forma dinámica por medio de los servlets, mostrar los resultados una vez finalizado el test añadiendo cada vez más detalles como un contador que muestra la duración o un botón que nos lleva al siguiente test en caso de existir para ese tema.

También se han preparado los enlaces a los test de cada uno de los temas, e introducir información para los test.

En el apartado anterior "Desarrollo I", se crean los servlets haciendo pruebas sin datos reales ni hojas de estilo hasta conseguir que funcionase. En esta fase del proyecto se fueron añadiendo los estilos y por último los datos.

Se pasa todo a JSP ya que hasta el momento habíamos trabajado con html y se trabaja sobre todo con los servlets para ver las preguntas y para evaluar el test.

Se desarrolla también una página intermedia o de índice entre la página principal y la página individual de cada test en HTML, donde aparecen el temario y 5 apartados con los que elegir de qué grupo de temas o número de test queremos realizar test.

Por último, se completan las páginas que había vacías en el footer tanto los enlaces a redes sociales, como el About, Sitemap (mapa del sitio web), FAQ (preguntas frecuentes), Terms (términos y condiciones), Privacy (privacidad) y el enlace a la web del autor.

Todo ello desde la base de diseño responsivo (Responsive Design), ya que actualmente hay multitud de dispositivos desde donde se consume contenido, cada uno con un tamaño y resolución.

Continuamos con la labor de documentación, en el apartado que nos ocupa y complementando el resto de la memoria.

6.5. ITERACIÓN 5. PRUEBAS

Una vez finalizada la labor de desarrollo, se empieza la fase de pruebas y verificación del software con el objetivo de demostrar que el sistema es conforme a la especificación y que resuelve los requisitos planteados inicialmente.

Se prueban todas las funcionalidades de la aplicación web una a una con la intención de encontrar posibles fallos o futuras mejoras. Una de las formas de realizar dichas pruebas es por medio de un test de usabilidad que veremos en el apartado 6.5.1.

Algunos de los problemas subsanados eran que el resultado podía ser menor que cero si respondías varias preguntas mal (cada pregunta mal respondida resta 1/3), un error convirtiendo datos de un tipo a otro y al introducir ciertas combinaciones de respuestas correctas e incorrectas se caía la aplicación, incompatibilidades con las hojas de estilo y un problema para desmarcar los radio button.

Tampoco se entendía bien el apartado de "Temario" o el menú temario, así que se añaden descripciones para ambos apartados.

Continuamos con la labor de documentación, en el apartado que nos ocupa y complementando el resto de la memoria.

6.5.1. Test de usabilidad

Es necesario conocer y saber medir la usabilidad de la interfaz ya que, aunque a los creadores nos parezca sencilla e intuitiva no lo es siempre para el usuario, por ello es conveniente analizar su comportamiento frente a la interfaz a la hora realizar las tareas.

Así podemos detectar y corregir las debilidades de nuestra interfaz y llegar incluso a replanteárnosla en su conjunto. La usabilidad ha ido ganando importancia en los últimos años llegándose a considerar incluso ingeniería.

Para medir la usabilidad de la aplicación web se ha pedido a familiares y amigos realizar una determinada tarea. Se han recogido esas

observaciones y el comportamiento que han tenido para ir mejorando posibles errores de diseño, ortografía, etc.

Para realizar el test se han seguido una serie de pasos. El primero es elaborar una lista con los objetivos a analizar:

- Grado de interactividad de la aplicación.
- Eficiencia. Tiempo necesitado en realizar cada acción, tanto el tiempo cognitivo (el usuario piensa cual es el elemento que le permite hacer la acción) como el tiempo en mover el ratón y desplazarse hasta el correspondiente elemento de la interfaz.
- Efectividad. El usuario llega a realizar la tarea encomendada.
- Recoger comentarios sobre colores, fuentes, iconos.

El segundo paso sería asignar roles al equipo, al estar formado por una única persona, el autor del presente TFG realizará todos los roles (facilitador, ordenador y observador).

El tercer paso es desarrollar una plantilla para seguir los mismos pasos con cada uno de los usuarios. Se evalúan varios aspectos: Satisfacción, eficiencia, eficacia, observaciones y preguntas.

- **(SAT)** Reacción de usuario con el primer contacto con la interfaz.

Métrica: Mal / Aceptable / Bien / Muy bien

- **(EFC)** El usuario encuentra la acción "Ver novedades". (Si tarda menos de 5 segundos es válida).

Métrica: Mal / Aceptable / Bien / Muy bien

- **(EFZ)** La última novedad es "Nuevo temario". Queremos ver cómo es. (¿Abre el enlace?).

Métrica: Si / No

- **(EFC)** El usuario encuentra la acción "Abrir algún examen de años anteriores, cualquiera". (Si tarda menos de 5 segundos es válida).

Métrica: Mal / Aceptable / Bien / Muy bien

- **(EFZ)** El usuario encuentra la acción "Ver test de Materias Comunes".

Métrica: Si / No

- **(EFZ)** Ahora, el usuario encuentra la acción "Realizar test de simulación de examen".

Métrica: Si / No

- **(EFZ)** ¿Pasa por el menú temario o vuelve a index?
Métrica: Menú Temario / Vuelve a inicio
- **(OBS)** ¿El usuario ha necesitado ayuda del facilitador? ¿En qué?
- **(OBS)** El usuario parece perdido al realizar....
- **(PRE)** ¿Qué es lo que menos te ha gustado de la aplicación?
- **(PRE)** ¿Cambiaría algo? ¿Cómo?
- **(PRE)** ¿Qué es lo que más te ha gustado de la aplicación?
- **(SAT)** Si fueses estudiante de las oposiciones de TSI, ¿te sería útil?
Métrica: Sí / No

Esta es la plantilla seguida, los resultados se encuentran en el siguiente apartado 7.2 (resultados test de usabilidad).

6.6. ITERACIÓN 6. ENTREGA FINAL

Se trata de la última iteración, el sprint con menor duración de los seis que conforman el presente TFG. El objetivo es intentar pulir detalles visuales, mejoras de funcionamiento, realizar las últimas pruebas y añadir alguna funcionalidad para dar por concluido nuestro proyecto.

Una de las mejoras introducidas ha sido guardar en un fichero aparte la cabecera y el pie de página. Así evitamos repetir los mismos bloques de código en todas las páginas y hacemos que el código se vea mucho más claro.

Se ha mejorado el diseño responsivo para pantallas de menor tamaño, se han cambiado colores de corrección de los test hasta dar con unos acordes a la interfaz propuesta y se han implementado páginas por si salta un error mostrar información al usuario.

Se ha añadido una página llamada interés.jsp con información de interés relacionada con la oposición (fases del proceso de selección, sueldos, requisitos, etc).

Lo que más tiempo ha llevado ha sido introducir datos. Las fuentes han sido test de años anteriores, información de internet relacionada con cada una de las materias presentes en el temario de la oposición o algún test propio.

Se ha implementado como mínimo un test para cada uno de los temas, dos en algún caso. También en el apartado de exámenes se ha ingresado una cantidad importante de datos (ver el fichero adjunto para poblar la BBDD, son miles los apartados introducidos).

En esta fase se terminan las labores de documentación: bloque II (Implementación, resultados, conclusiones, etc), bloque III (Manual de instalación, manual de usuario y contenido del CD-ROM) y bloque IV (Bibliografía).

Por último, se ha preparado una presentación resumen del proyecto lo más amena y clara posible, con el objetivo de reflejar cada uno de los pasos seguidos del proceso de desarrollo. Presentación, introducción (con el objetivo del trabajo), nudo y desenlace.



7 . RESULTADOS

Con el proyecto finalizado, podemos comparar la planificación total con la estimación que se hizo en la planificación inicial.

Rol	Horas estimadas	Horas requeridas
Analista	50h	50h
Diseñador	30h	30h
Implementador	180h	200h
Tester	30h	25h
Documentador	70h	80h
TOTAL	360h	385h

Tabla 38 - Horas estimadas vs Horas reales

Como podemos comprobar, alguna tarea se alargó algo más de lo debido como la fase de implementación y la fase de documentación y en cambio la fase de pruebas o testeo tomó menos horas de las previstas.

La implementación del TFG aumentó en unas 25 horas a causa de cambios en las diversas iteraciones siendo el más significativo y de mayor complejidad cargar las preguntas desde la base de datos de una forma dinámica y aleatoria entre un banco de preguntas.

7.1. COSTES REALES

En el apartado 4.7 se detallaba la estimación inicial de los costes totales del proyecto. En este apartado vamos a comparar los costes estimados con los costes finales reales.

Como podemos observar en el apartado anterior las tareas de análisis y diseño no han aumentado. Sin embargo, las tareas de desarrollo y documentación provocan un ligero sobrecoste:

Desarrollo: 40€/hora x 20 horas = 800€

Documentación: 30€/hora x 10 horas = 300€

A esto habría que descontar 5 horas en tareas de testeo, que pasan de 30 a 25 horas.

Tester: 30€ x 5 horas = 150€

Los costes derivados de hardware y oficina se han mantenido tal y como se había previsto en los gastos iniciales. En cuanto a los servicios software, con los planes destinados a estudiantes de Microsoft Office y StarUML pudimos ahorrar otros 198€.

Así pues, si sumamos el **sobrecoste total de 782€** al coste inicial previsto del proyecto tenemos un coste final real de 17.610,40€.

Coste estimado proyecto	+16.858,40€
Desarrollo	+800,00€
Documentación	+300,00€
Ayudas	-198,00€
Tester	-150,00€
COSTE FINAL REAL DEL PROYECTO	17.610,40€

7.2. RESULTADOS TEST DE USABILIDAD

En el apartado 6 implementación, veíamos como en el quinto punto realizábamos pruebas a la aplicación. Al ser una web cuyo objetivo primordial es ser intuitiva y eficiente, se realizó un test de usabilidad a varios familiares y amigos. Estos son los resultados:

En principio la realización de las acciones principales se desarrolla de una forma correcta y la impresión inicial ha sido buena para todos.

Aun así, se introducen más enlaces a los mismos sitios desde otros puntos diferentes de la aplicación web para minimizar el tiempo de respuesta del usuario. Y se añaden títulos de los enlaces para que sirvan como guía durante la navegación.

TEST DE USABILIDAD

- **(SAT)** Reacción de usuario con el primer contacto con la interfaz.
 Métrica: Mal / Aceptable / Bien / Muy bien MB MB MB MB
Bechta Seguridad Lubric - Calle
- **(EFC)** El usuario encuentra la acción "Ver novedades". (Si tarda menos de 5 segundos es válida).
 Métrica: Mal / Aceptable / Bien / Muy bien MB MB MB MB
- **(EFZ)** La última novedad es "Nuevo temario". Queremos ver cómo es. (¿Abre el enlace?).
 Métrica: Si / No Si Si No Si
- **(EFC)** El usuario encuentra la acción "Abrir algún examen de años anteriores, cualquiera". (Si tarda menos de 5 segundos es válida).
 Métrica: Mal / Aceptable / Bien / Muy bien MB MB MB MB
- **(EFZ)** El usuario encuentra la acción "Ver test de Materias Comunes".
 Métrica: Si / No Si * Si Si Si
- **(EFZ)** Ahora, el usuario encuentra la acción "Realizar test de simulación de examen".
 Métrica: Si / No Si Si Si Si
- **(EFZ)** ¿Pasa por el menú temario o vuelve a index?
 Métrica: Menú Temario / Vuelve a inicio MENÚ TEMARIO (opuón ATRÁS del Navegador) MT (Más) MT (Medio) MT (Medio)
- **(OBS)** ¿El usuario ha necesitado ayuda del facilitador? ¿En qué? NO NO NO NO
- **(OBS)** El usuario parece perdido al realizar.... MENÚ TEMARIO MENÚ - -
- **(PRE)** ¿Qué es lo que menos te ha gustado de la aplicación? - - - -
- **(PRE)** ¿Cambiaría algo? ¿Cómo? - HAZER MÁS GRANDE LAS MÍGAS DE PAN - los colores igual, no se. - No se
- **(PRE)** ¿Qué es lo que más te ha gustado de la aplicación? los colores y los iconos - los test. - Todo - la utilidad que puedo darle.
- **(SAT)** Si fueses estudiante de las oposiciones de TSI, ¿te sería útil?
 Métrica: Sí / No sí Si Si Si

* Camino: Imagen temario > MC > Test. No entra directa. ← ACCIÓN: de añade INFO EXTRA en index.jsp

Figura 20 - Test de usabilidad

A la hora de evaluar los test y dar la puntuación final, Inicialmente se plantea rojo respuesta incorrecta, verde respuesta correcta y naranja respuesta sin contestar. Tras los resultados, la última se cambia por un gris claro para simplificar y se añade una leyenda con el significado de cada tipo de corrección ya que no resultaba del todo claro.

Viendo a los usuarios un poco perdidos con la forma de entrar en el menú de los test (todos lo consiguen, pero cada uno de una forma) se introduce una explicación en la página principal.

7.3. CAPTURAS DE LA APLICACIÓN

Con respecto al diseño final de la aplicación web podemos comprobar que no ha diferido mucho respecto al boceto inicial adjunto en el apartado 5.

Si bien es cierto que el diseño es bastante parecido, la estructura totalmente en vertical recorriendo su totalidad por medio de scroll vertical en un principio no se tuvo en cuenta, así como el timeline o el footer con un diseño final más moderno y atractivo.

Comparando los bocetos con la aplicación final podemos observar que se han añadido bastantes funcionalidades que al principio no se contemplaban y que han ido surgiendo según se ha ido desarrollando.

Por ejemplo, acceder a información relevante. Se empezó con la idea de añadir un test de prueba para que el usuario pudiese probar la aplicación de una manera rápida y simple, pero se ha acabado sustituyendo por información relevante, ya que se considera que va a tener una mayor utilidad.

TÉCNICO SOPORTE INFORMÁTICO TEST ONLINE EXÁMENES NOVEDADES

JCYL 17/18

OPOSICIONES

www.tecnicosoporteinformatico.com

VENTAJAS

- Ahorro de tiempo.**
Un opositor necesita tiempo para estudiar. Corrigiendo de forma automática e instantánea los test online o accediendo a nuestro apartado de novedades podrás aprovechar ese tiempo ganado en lo que tú quieras.
- Mejores resultados.**
Realizar los test de forma interactiva es una buena alternativa para conseguir hacer tu estudio más ameno y efectivo. Además tendrás todos los contenidos mejor organizados y disponibles en cualquier momento.
- Complemento al estudio.**
Dicha web no viene a sustituir los libros de teoría ni las intensas e interminables horas de estudio, pues es una oposición y es lo que conlleva. La finalidad es convertirse en una herramienta complementaria.

🔍 Ver datos de interés de la oposición

MENÚ TEMARIO

(CLICK EN LA IMAGEN PARA VER EL TEMARIO O ELEGIR LA SECCIÓN)

1	2	3	4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
MATERIAS COMUNES				ÁMBITO ORGANIZATIVO				TECNOLOGÍA BÁSICA										SEGURIDAD				COMUNICACIONES Y REDES										

TEST POR SECCIONES

(SI TIENES CLARO LA SECCIÓN A PREPARAR, ENTRA DIRECTAMENTE EN CADA UNA DE ELLAS)

 MATERIAS COMUNES	 ÁMBITO ORGANIZATIVO	 TECNOLOGÍA BÁSICA
 SEGURIDAD	 COMUNICACIONES Y REDES	

Figura 21 – Captura pantalla principal ventajas y temario

- 🔍 Examen 2016 - Ejercicio 1 turno libre.pdf
- 🔍 Solución Examen 2016 - Ejercicio 1 turno libre.pdf
- 🔍 Examen 2016 - Ejercicio 1 Promoción Interna.pdf
- 🔍 Solución 2016 - Ejercicio 1 Promoción Interna.pdf
- 🔍 Examen 2009 - Ejercicio 1 turno libre.pdf
- 🔍 Examen 2007 - Ejercicio 1 turno libre.pdf
- 🔍 Examen 2006 - Libro de ejercicios y soluciones. Recomendable.pdf
- 🔍 Examen 2006 - Ejercicio libre y soluciones.doc

NOVEDADES



CONTACTO

Email: tecnicosoporteinformatico@gmail.com

REDES SOCIALES



TFG

Escuela Universitaria de Informática (SEGOVIA).
Portal web para la preparación de las oposiciones de Técnico de Soporte Informático de la Junta de Castilla y León.

luisorozco.dx.am

Figura 22 – Captura exámenes timeline y footer

TÉCNICO SOPORTE INFORMÁTICO

Inicio > Temario



El objetivo es realizar test online para preparar lo que estemos estudiando en cada momento. Tenemos dos opciones:

A) Test para preparar un tema. Basta con elegir primero el apartado al que pertenezca y luego el número de tema. Constan de 20 preguntas fijas (no cambian). Fondo gris.

B) Test para preparar el examen. Constan de 55 preguntas (aleatorias). Fondo naranja.

La cantidad de test disponibles dentro de cada tema se irá ampliando con el paso del tiempo.



TEMARIO DE TÉCNICO DE SOPORTE INFORMÁTICO

MATERIAS COMUNES

Tema 1. La Constitución española de 1978: estructura y contenido.

Tema 2. El Estatuto de Autonomía de Castilla y León. La Ley del Gobierno y de la Administración de Castilla y León.

Tema 3. La Unión Europea. Las Instituciones comunitarias.

Tema 4. El personal al servicio de la Administración de la Comunidad Autónoma de Castilla y León. El Convenio Colectivo para el personal laboral de la Administración General de la Comunidad de Castilla y León y Organismos autónomos dependientes de ésta (Boletín Oficial de Castilla y León de 28 de octubre de 2013).

MATERIAS ESPECÍFICAS

Bloque I - Ámbito organizativo

Tema 1. Estructura y Organización de un Departamento de Sistemas de Información. Organización de la actividad informática en la Junta de Castilla y León. Catálogo de puestos de trabajo.

Tema 2. Gestión de la atención a clientes y usuarios: funciones y estructura. Técnicos de primer y segundo nivel. Gestión de incidencias y su ciclo de vida. ITIL v3: fundamentos básicos y estructura, funciones y roles en la Operación de Servicios.

Tema 3. La protección jurídica de los programas de ordenador. Propiedad intelectual. Software libre, propietario y licenciamiento. Código abierto.

BLOQUE II - Tecnología básica

Tema 4. Informática básica. Concepto de datos e información. Concepto de Sistema de Información. Elementos constitutivos de un sistema. Arquitectura de ordenadores: componentes y funciones.

Tema 5. Tecnologías actuales de ordenadores: desde los dispositivos móviles a los superordenadores y arquitecturas escalables.

Tema 6. Ficheros. Organización de la información. Métodos de inserción y acceso. Indexación. Formatos y características. Operaciones.

Tema 7. Dispositivos de almacenamiento: discos, discos sólidos, memorias flash, soportes magneto/ópticos, cintas? Tecnologías y velocidades. Arquitecturas de almacenamiento masivo NAS y SAN: componentes y funciones.

Tema 8. Periféricos y su conectividad. Periféricos de entrada, salida, almacenamiento y comunicación. Buses y sus tipos e interfaces. USB, Firewire, SCSI, Bluetooth. Interfaces gráficas clásicas y de alta definición.

Tema 9. Infraestructura física de un CPD: acondicionamiento y equipamiento.

Tema 10. Sistemas operativos. Clasificación. Arquitectura, componentes y funciones. Administración de memoria. Procesos y subprocesos. Entrada/salida. Sistemas de archivo. Planificadores y gestión multitarea.

Tema 11. Arquitectura de sistemas Cliente Servidor y multicapas: componentes y funcionalidades. Arquitectura de servicios web.

Tema 12. Distribución y actualización de software en un entorno distribuido. Conceptos.

Tema 13. La familia de sistemas operativos de usuario Windows (XP y posteriores). Características técnicas y funcionalidades. Servicios y recursos. Instalación, administración, mantenimiento y recuperación. Gestión del software base e instalación y desinstalación de productos.

Tema 14. El sistema operativo UNIX, LINUX. Características técnicas y funcionalidades. Servicios, operaciones y comandos básicos. Supervisión de logs y mensajes de consola.

Tema 15. Servidores Windows. Mantenimiento diario. Administración de usuarios y grupos de trabajo. Fundamentos de Directorio Activo. Supervisión de visor de eventos.

Tema 16. Herramientas y paquetes ofimáticos. Microsoft Office 2013 y sus componentes. Macros.

Tema 17. Correo electrónico. Arquitectura y componentes. Clientes de correo. El cliente de Microsoft Outlook 2013: configuración y funcionalidades. Acceso al correo mediante interfaz web y dispositivo móvil.

Tema 18. Introducción a las bases de datos. Tipos. Modelo conceptual de datos. Entidades, atributos y relaciones. El modelo lógico relacional. Estándares de conectividad.

Tema 19. Lenguajes de marca o etiquetas. Características y funcionalidades: HTML, SGML, XML y sus derivaciones. Lenguajes de script. Tecnologías de programación en Web.

Tema 20. Diseño de interfaz de usuario. Principios de diseño de interfaces. Accesibilidad y usabilidad.

BLOQUE III - Seguridad

Tema 21. La protección de datos de carácter personal. Ley Orgánica y Normativa de desarrollo. La Agencia de Protección de datos: competencias y funciones.

Tema 22. Conceptos de seguridad de los sistemas de información. Seguridad física. Seguridad Lógica. Amenazas y vulnerabilidades. Protección de activos. Políticas de salvaguarda de la información.

Tema 23. Técnicas Criptográficas y de cifrado, algoritmos y mecanismos de firma digital. Certificados digitales y DNI electrónico. PKI.

Tema 24. Virus. Tipos (Malware, Spyware, Ransomware, etc). Medios preventivos y reactivos. Programas antivirus: instalación, actualización y mantenimiento. Los paquetes de seguridad integral del puesto de trabajo.

BLOQUE IV - Comunicaciones y redes

Tema 25. Modelo de referencia OSI de ISO. El modelo TCP/IP: arquitectura capas, protocolos, direccionamiento y encaminamiento.

Tema 26. Redes LAN, MAN y WAN. Tipología. Métodos de acceso y protocolos. Medios de transmisión y equipos de interconexión. Cableado estructurado y normativas (EN50173, EN 50174). Administración de redes locales y equipamiento. Seguridad: contraseñas, protocolos seguros (IPSec, SSL) y redes privadas

Figura 23 – Captura menú temario

TÉCNICO SOPORTE INFORMÁTICO

Temario ▶ Materias Comunes



MATERIAS COMUNES

Tema 1.- La Constitución española de 1978: estructura y contenido.
TEST 1.1

Tema 2.- El Estatuto de Autonomía de Castilla y León. La Ley del Gobierno y de la Administración de Castilla y León.
TEST 2.1

Tema 3.- La Unión Europea. Las Instituciones comunitarias.
TEST 3.1

Tema 4.- El personal al servicio de la Administración de la Comunidad Autónoma de Castilla y León. El Convenio Colectivo para el personal laboral de la Administración General de la Comunidad de Castilla y León y Organismos autónomos dependientes de ésta.
TEST 4.1

CONTACTO

Email: tecnicosoporteinformaticojcy@gmail.com

REDES SOCIALES

[!\[\]\(d65d67af0d48c0b8df8416565067ed4b_img.jpg\)](#) [!\[\]\(903130ed6058e2c828f13cd7443f1600_img.jpg\)](#) [!\[\]\(b25355660c87d4489d7c3e6a9e50e78c_img.jpg\)](#)

TFG

Escuela Universitaria de Informática (SEGOVIA). Aplicación web para la preparación de las oposiciones de Técnico de Soporte Informático de la Junta de Castilla y León.

luisorozco.dx.am

[About](#) [Sitemap](#) [FAQ](#) [Terms](#) [Privacy](#)
Copyright © www.tecnicosoporteinformatico.com 2017

Figura 24 - Captura menú materias comunes

TÉCNICO SOPORTE INFORMÁTICO

Temario > Materias Comunes > Test

TEST 01

1. Según la Constitución Española, no es un derecho fundamental:

- a) El derecho a la propiedad privada.
- b) El derecho a la vida.
- c) La libertad ideológica, religiosa y de culto.
- d) El derecho de asociación.

2. La Constitución Española establece que es competencia exclusiva del Estado:

- a) La agricultura y ganadería, de acuerdo con la ordenación general de la economía.
- b) La asistencia social.
- c) La legislación sobre propiedad intelectual e industrial.
- d) La ordenación del territorio, urbanismo y vivienda.

3. Según la Constitución Española, los miembros del Tribunal Constitucional son nombrados por:

- a) El Rey.
- b) El Congreso.
- c) El Senado.
- d) El Consejo General del Poder Judicial.

4. Las Instituciones básicas de la Comunidad de Castilla y León son:

- a) Las Cortes de Castilla y León, El Procurador del Común y El Consejo de Cuentas.
- b) Las Cortes de Castilla y León, El Presidente de la Junta de Castilla y León y La Junta de Castilla y León.
- c) Las Cortes de Castilla y León, El Procurador del Común, El Consejo de Cuentas, El Consejo Económico y Social, y el Consejo Consultivo.
- d) Las Cortes de Castilla y León, La Junta de Castilla y León y El Consejo de Cuentas.

5. La Comunidad Autónoma de Castilla y León, según su Estatuto de Autonomía, tiene competencia exclusiva en:

- a) La estructura y organización de la Administración de la Comunidad.
- b) Las obras públicas de interés para la Comunidad Autónoma dentro o fuera de su propio territorio que no tengan la calificación legal de interés general del Estado.
- c) Los ferrocarriles, carreteras y caminos que transcurran total o parcialmente por el territorio de la Comunidad Autónoma y, en los mismos términos, los transportes terrestres, fluviales, por cable o tubería.
- d) Todas las respuestas anteriores son correctas.

6. La estructura orgánica de la Administración de la Junta de Castilla y León, se aprueba por:

- a) El Presidente, a propuesta de cada Consejero.
- b) La Junta de Castilla y León.
- c) Las Cortes, por mayoría simple, a propuesta del Presidente de la Junta.
- d) El Consejero competente en la materia, con la conformidad del Presidente de la Junta.

7. El acto jurídico de la Unión Europea que es obligatorio en todos sus elementos y directamente aplicable en cada Estado miembro, se denomina:

- a) Reglamento.
- b) Decisión.
- c) Directiva.

...

CORREGIR TEST

Figura 25 - Captura de Test

TÉCNICO SOPORTE INFORMÁTICO

Temario > Materias Comunes > Test

REVISIÓN TEST 01

PUNTUACIÓN 5,33

Correctas: 6 | Incorrectas: 2 | Sin contestar: 2

Color Verde = Respuesta correcta

Color Rojo y verde = Respuesta corregida (Rojo incorrecta marcada por nosotros, Verde correcta)

Color Gris = Respuesta sin contestar

1. Según la Constitución Española, no es un derecho fundamental:

- El derecho a la propiedad privada.
- El derecho a la vida.
- La libertad ideológica, religiosa y de culto.
- El derecho de asociación.

2. La Constitución Española establece que es competencia exclusiva del Estado:

- La agricultura y ganadería, de acuerdo con la ordenación general de la economía.
- La asistencia social.
- La legislación sobre propiedad intelectual e industrial.
- La ordenación del territorio, urbanismo y vivienda.

3. Según la Constitución Española, los miembros del Tribunal Constitucional son nombrados por:

- El Rey.
- El Congreso.
- El Senado.
- El Consejo General del Poder Judicial.

4. Las Instituciones básicas de la Comunidad de Castilla y León son:

- Las Cortes de Castilla y León, El Procurador del Común y El Consejo de Cuentas.
- Las Cortes de Castilla y León, El Presidente de la Junta de Castilla y León y La Junta de Castilla y León.
- Las Cortes de Castilla y León, El Procurador del Común, El Consejo de Cuentas, El Consejo Económico y Social, y el Consejo Consultivo.
- Las Cortes de Castilla y León, La Junta de Castilla y León y El Consejo de Cuentas.

10. Con subordinación a las necesidades del servicio, un trabajador podrá solicitar licencias por asuntos propios, cuya duración acumulada será:

- No podrá exceder de tres meses en un año, ni ser inferior a dos días.
- No podrá exceder nunca de tres meses cada dos años ni podrá ser inferior a cinco días.
- No podrá exceder nunca de seis meses cada dos años, ni ser inferior a quince días.
- Tres días por cada año completo de servicios prestados, con un máximo de tres meses cada dos años.

SIGUIENTE TEST

Figura 26 – Captura resultado corregido

8 . CONCLUSIONES

La realización de este TFG en lo personal me ha servido para afianzar conocimientos de HTML y CSS3, y ampliarlos en JavaScript, servlets y otras tecnologías web. Con esfuerzo y dedicación se ha conseguido un diseño atractivo, funcional y eficiente, que era uno de los objetivos iniciales.

Ha sido útil repasar algunos conceptos ya olvidados de cara a la búsqueda de un empleo, aprender a organizarme mejor, seguir unos patrones, unos objetivos y unos plazos que cumplir.

Siempre he estado muy ligado al diseño, siempre me llamó la atención, y este Trabajo Fin de Grado me ha hecho darme cuenta de que dentro de los muchos caminos que puede seguir un informático, quiero que mi camino sea el del diseño web, me siento más Front-End que Back-End.

Por último, destacar la utilidad de la web al menos a título personal pues me encuentro preparando las oposiciones de Técnico de Soporte Informático, tema central de dicho proyecto. Esperemos que de verdad cumpla su cometido, me ayude a estudiar, y con suerte a aprobar.

9

. TRABAJOS FUTUROS

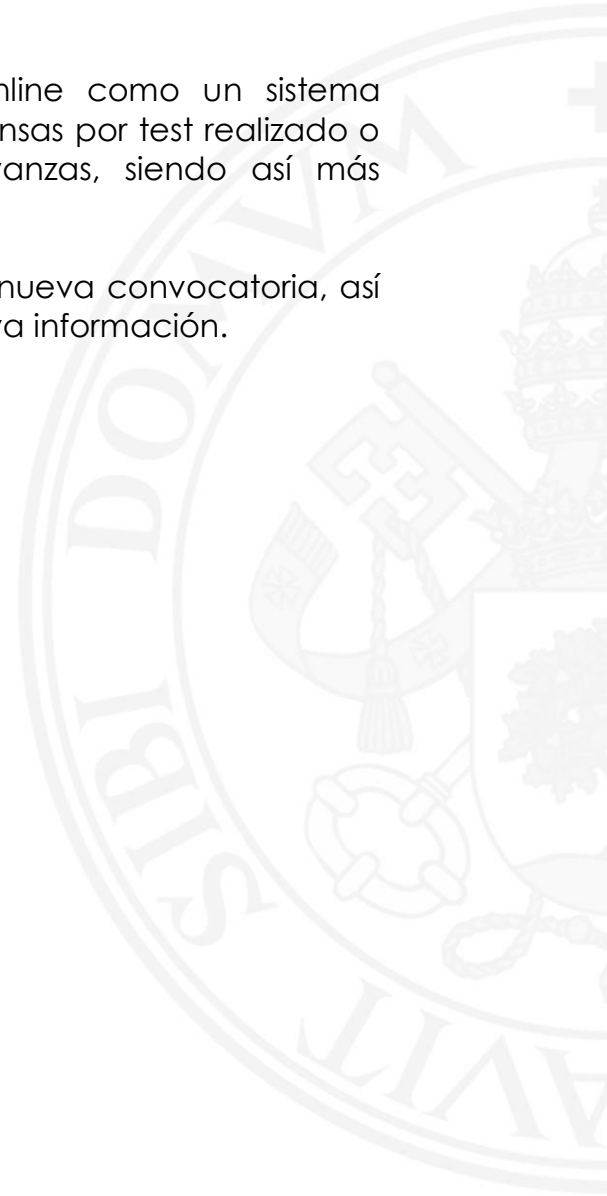
Son muchas las ampliaciones posibles para este Trabajo Fin de Grado, el cual tiene una duración determinada o estimada en unas 300 horas y limita las funcionalidades a implementar.

Aunque entiendo que lo que se pretende no es hacerlo completo al 100% sino aprender la metodología de trabajo, fases de este o saber aplicar los conceptos teóricos aprendidos en los años de carrera.

La siguiente lista enumera cada una de las posibles ampliaciones que se me han ido ocurriendo (posiblemente pudiera haber más):

- Añadir más oposiciones, de informática o de otras ramas, siguiendo la misma estructura.
- Posicionamiento SEO y SEM de la web.
- Mejoras de rendimiento de la web, mejorando el código, optimizando o reduciendo los tiempos de carga, de base de datos, etc.
- Ampliar el número de tests o preguntas disponibles para cada bloque.
- Eliminar la necesidad de una conexión a internet, por ejemplo, ofreciendo la posibilidad de descargar los tests.
- Funcionalidades nuevas como encuestas y comentarios de los usuarios.
- Ampliar sección de información interesante como funciones de los técnicos, condiciones, destinos, posibilidades de movilidad y promoción interna, compatibilidad.
- Blog con historias reales de los usuarios.
- Descarga o visualización de la teoría del temario.

- Acuerdos con academias que preparen dichas oposiciones para mejorar la página con todo lo aquí descrito y que permita a los usuarios acceder a contenidos exclusivos previo pago de un dinero.
- Posibilidad de registrarse, guardar resultados de los test online y gráficos que muestren la evolución.
- Replantear la realización de test online como un sistema parecido a un juego, dando recompensas por test realizado o subiendo niveles a medida que avanzas, siendo así más entretenido.
- Actualizar el temario cuando salga la nueva convocatoria, así como reubicar los test e introducir nueva información.



BLOQUE 3

APÉNDICES

III .APÉNDICES

El proyecto es compatible con los sistemas operativos Windows, Linux, Mac, para ejecutarlo solo será necesario un navegador web. Para que funcione correctamente se aconseja cumplir una serie de requisitos mínimos:

- Actualizar el navegador a la última versión.
- JavaScript y cookies activadas.
- Active X¹³ activo y desbloqueado para Microsoft Internet Explorer, sino alguna funcionalidad podría fallar.

La aplicación ha sido probada en los siguientes navegadores:

- Internet Explorer 11.0.9600.18861
- Google Chrome Versión 64.0.3282.119 (Build oficial) (64 bits)
- Mozilla Firefox 56.0

Por defecto hemos desarrollado todo el TFG usando Google Chrome por ser el navegador más utilizado con casi un 60% del total (según un reciente estudio de StatCounter). Como resolución se han ido probando muchas desde 2560x1080, hasta los 375x667 que es la proporción del iPhone 7.

Podemos acceder a la aplicación web de dos formas. La primera sería de forma local y la segunda a través de Internet, siendo el funcionamiento semejante en ambos casos. A lo largo del desarrollo se ha trabajado de forma local por comodidad y rapidez.

Me parece interesante repasar de una forma simple (sin tener en cuenta las acciones internas del servidor) lo que sucede cuando el cliente introduce la dirección URL de nuestro proyecto en un navegador.

¹³ **ActiveX** fue presentado en 1996 por Microsoft como una evolución de sus tecnologías Component Object Model (COM) y Object Linking and Embedding (OLE).

Lo primero, abrimos el navegador e introducimos la URL¹⁴ deseada (<http://localhost:8080/www.tecnicosoporteinformatico.com>). El navegador la descompone en los siguientes apartados:

- **http://** es el protocolo utilizado para obtener el recurso. (Por defecto).
- **localhost** es el nombre del servidor web que contiene el recurso o hostname, que luego es traducida por el servidor de nombres de dominio (DNS) en una dirección IP que identifica el servidor en internet. En nuestro caso es la 127.0.0.1, es decir, nuestro ordenador. (En caso de subir la página al dominio teórico del trabajo, sería www.tecnicosoporteinformatico.com que es justo lo que hemos puesto de nombre de recurso para que se asemeje algo).
- **:8080** es el puerto. (Por defecto en Tomcat).
- **/www.tecnicosoporteinformatico.com/** es el recurso buscado (Opcional).

Al ser el protocolo por defecto, podríamos sustituir la dirección anterior por "127.0.0.1:8080/www.tecnicosoporteinformatico.com/" y nos lleva igualmente a nuestro sitio.

El navegador mediante el protocolo TCP/IP (base de la comunicación en internet) establece una **conexión** con el servidor (ubicado en cualquier lugar del planeta) y envía una petición HTTP al servidor (**request**).

Una vez la recibe el servidor, se intercambian **cabeceras** (con información adicional de los datos a enviar, por ejemplo, el estándar MIME especifica los formatos de los datos) y el servidor procesa dicha solicitud utilizando la aplicación escrita en JSP.

Durante ese proceso se conectará si es necesario a la base de datos (Apache Derby) y procesará una respuesta (**response**), con el contenido solicitado `index.html`.

El navegador recibe los datos y crea la representación interna conocida como DOM, ejecuta el Javascript y a partir de ahí, junto con el CSS enviado el navegador nos muestra la página web.

¹⁴ **URL**: Uniform Resource Locator, se trata de la secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados.



Figura 27 - Esquema de la aplicación

APÉNDICE 1: MANUAL DE INSTALACIÓN

Para poder ejecutar y desarrollar la aplicación debemos tener instalados y configurados correctamente 4 programas:

1. JDK.
2. NetBeans.
3. Apache Tomcat.
4. Apache Derby.

JDK es el Java Development Kit o, en español, Herramientas de Desarrollo de Java. Sirve para construir programas usando el lenguaje de programación Java. Una instalación de JDK ya contiene un JRE dentro de las carpetas. JRE es el Java Runtime Environment o, en español, el Entorno de Ejecución de Java. JRE no posee compiladores ni herramientas para desarrollar las aplicaciones Java, solo posee las herramientas para ejecutarlas.

The screenshot shows the Oracle Java SE Development Kit 8 Downloads page. The page is divided into several sections:

- Navigation:** Oracle logo, Menu, Search, Sign In, Country/Region, Call.
- Breadcrumbs:** Oracle Technology Network / Java / Java SE / Downloads
- Left Sidebar:** List of Java products: Java SE, Java EE, Java ME, Java SE Advanced & Suite, Java Embedded, Java DB, Web Tier, Java Card, Java TV, New to Java, Community, Java Magazine.
- Main Content:**
 - Overview, Downloads, Documentation, Community, Technologies, Training** tabs.
 - Java SE Development Kit 8 Downloads** section: Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language. The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.
 - See also:**
 - Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
 - Java Developer Day hands-on workshops (free) and other events
 - Java Magazine
 - JDK 8u161 checksum
JDK 8u162 checksum
 - Java SE Development Kit 8u161** section: You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
 - Accept License Agreement Decline License Agreement
 - Table of download links for various operating systems:
 - Java SE Development Kit 8u162** section: You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
 - Accept License Agreement Decline License Agreement
- Right Sidebar:**
 - Java SDKs and Tools:** Java SE, Java EE and Glassfish, Java ME, Java Card, NetBeans IDE, Java Mission Control.
 - Java Resources:** Java APIs, Technical Articles, Demos and Videos, Forums, Java Magazine, Developer Training, Tutorials, Java.com

Figura 28 - Instrucciones de descarga del JDK

Como nuestro objetivo no solo era ejecutar una aplicación Java sino desarrollarla, nos tenemos que instalar el JDK. Se puede descargar de forma totalmente gratuita desde la siguiente dirección web “http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html”, eligiendo nuestro sistema operativo y aceptando las condiciones como puede verse en la anterior imagen.

Una vez que tenemos el JDK procedemos a la instalación de **NetBeans**, un IDE (entorno de desarrollo integrado) que simplifica el desarrollo de aplicaciones web. La instalación podemos hacerla directamente desde “https://netbeans.org/downloads/index.html”, descargando la versión completa preferentemente como podemos ver en la imagen inferior en donde pone “download”, ya que incluye el Apache Tomcat y nos evitará tener que realizar otra descarga e instalación a mayores.

NetBeans IDE 8.1 Download

8.0.2 | 8.1 | 8.2 | Development | JDK9 Branch | Archive

Email address (optional):

Subscribe to newsletters: Monthly Weekly

NetBeans can contact me at this address

IDE Language: Platform:

Note: Greyed out technologies are not supported for this platform.

NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•	•				•
Java FX	•	•				•
Java EE		•				•
Java ME						•
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3 Connected						•
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

Download buttons: Download, Download, Download x86, Download x86, Download x86, Download, Download x64, Download x64, Download x64, Download

Free, 95 MB Free, 192 MB Free, 104 - 107 MB Free, 104 - 107 MB Free, 106 - 110 MB Free, 215 MB

* You can add or remove packs later using the IDE's Plugin Manager (Tools | Plugins).

HTML/JS, PHP and C/C++ NetBeans bundles include Java Runtime Environment and do not require a separate Java installation.

JDK 7 and later versions are required for installing and running the Java SE, Java EE and All NetBeans Bundles. You can download standalone JDK or download the latest JDK with NetBeans IDE Java SE bundle.

You can start developing applications based on the NetBeans Platform using the NetBeans IDE for Java SE. Learn more about the NetBeans Platform. NetBeans source code and binary builds without bundled runtimes are also available in zip file format. See also instructions on how to build the IDE from sources or installation

Important Legal Information:
NetBeans Community Distributions are available under a Dual License consisting of the Common Development and Distribution License (CDDL) v1.0 and GNU General Public License (GPL) v2. Such distributions include additional components under separate licenses identified in the License file. See the Third Party License file for external components included in NetBeans and their associated licenses.

WARNING: These builds are still under development and could contain defects that lead to data loss. They are meant for evaluation of new features. As always, be sure to back

Figura 29 - Instrucciones de descarga de NetBeans

En la misma página de NetBeans, podemos ver un enlace (señalado en verde en la anterior imagen) con un paquete todo en uno o bundle que contiene NetBeans y el JDK en un único archivo. Por simplicidad es recomendable esta segunda opción, aunque cualquiera de las dos es válida, se instala fácilmente y en tan solo unos minutos.

Actualmente (marzo de 2018) está disponible la versión 8.2 de NetBeans y la versión JDK 8u161, aunque en la página de Oracle podremos descargar otras versiones anteriores. En este proyecto se ha trabajado con NetBeans 8.1 y JDK 1.8.0_77.

The screenshot shows the Oracle Technology Network website for downloading JDK 7u80 with NetBeans 8.0.2. The page includes a navigation menu, a search bar, and a sidebar with various Java SDKs and Tools. The main content area features a table of download links for different operating systems, with the Windows x64 link highlighted in green.

Product / File Description	File Size	Download
Linux x86	233.68 MB	jdk-7u80-nb-8_0_2-linux-i586.sh
Linux x64	229.83 MB	jdk-7u80-nb-8_0_2-linux-x64.sh
Mac OS X x64	301.5 MB	jdk-7u80-nb-8_0_2-macosx-x64.dmg
Windows x86	243.83 MB	jdk-7u80-nb-8_0_2-windows-i586.exe
Windows x64	246.65 MB	jdk-7u80-nb-8_0_2-windows-x64.exe

Figura 30 - Instrucciones instalación bundle

Apache Tomcat es un contenedor web con soporte de servlets y JSPs, no es un servidor de aplicaciones, aunque implementa las especificaciones necesarias para el desarrollo del proyecto.

Podemos descargarlo de la web "http://tomcat.apache.org/ ", en nuestro caso al ser Windows la versión .zip, aunque como se comentó anteriormente la versión completa de NetBeans lo trae incluido. La versión con la que hemos trabajado es la 8.0.41.

Si se quiere añadir Tomcat a Netbeans debemos ir a Herramientas, servidores, añadir servidor, seleccionamos Apache Tomcat y damos a siguiente. Se crea un manager role, poniendo usuario y contraseña y se finaliza.

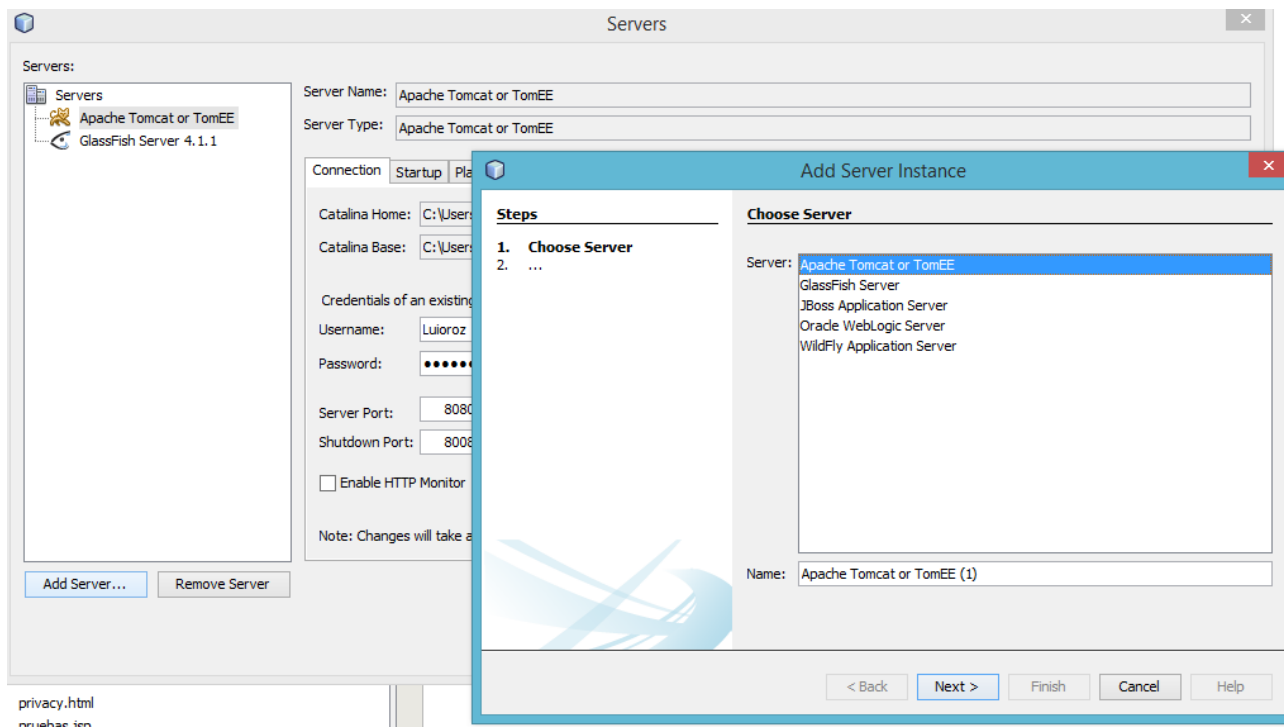
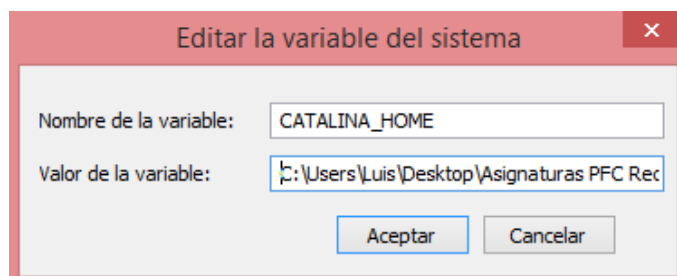


Figura 31 - Instrucciones instalación Tomcat en Netbeans

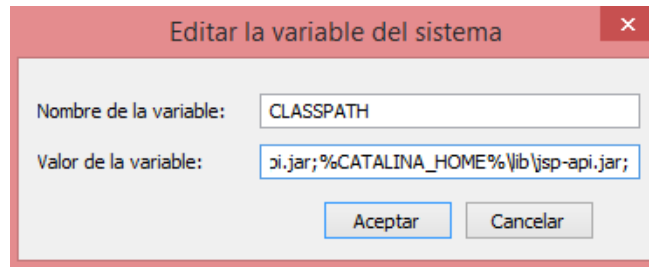
Imprescindible para que Tomcat funcione correctamente es necesario tener instalado el JDK y Java, redefiniendo y creando las siguientes variables de entorno:

1º. Se crea una nueva variable con nombre CATALINA_HOME y valor de la variable la ruta a la carpeta donde tengamos la carpeta con Tomcat descargado.

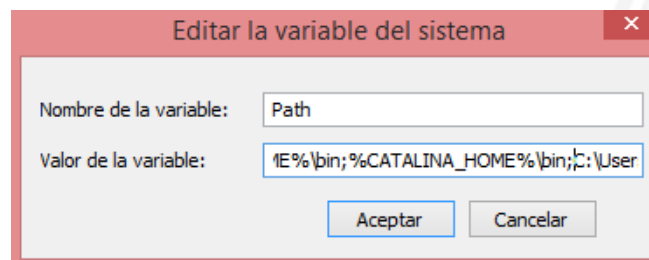


2º. Se edita la variable CLASSPATH, nos ubicamos en el campo Valor de la variable y nos desplazamos hasta el final, colocamos un punto

y coma e ingresamos el texto `%CATALINA_HOME%\lib\servlet-api.jar;` y sin cerrar la ventana editar hacemos lo mismo para añadir `%CATALINA_HOME%\lib\jsp-api.jar;`



3º. Editamos el valor de la variable PATH, nos ubicaremos sobre el campo Valor de la variable, nos desplazaremos hasta el final del valor y colocaremos un punto y coma he ingresaremos el texto `%CATALINA_HOME%\bin`



Una vez instalado y ejecutado, tenemos dos maneras de desplegar aplicaciones. Se puede hacer desde Netbeans y se puede hacer desde `http://localhost:8080` la cual nos muestra la pantalla de bienvenida de Apache Tomcat, esto quiere decir que todo se ha configurado exitosamente. Normalmente se trabajará directamente desde el IDE.

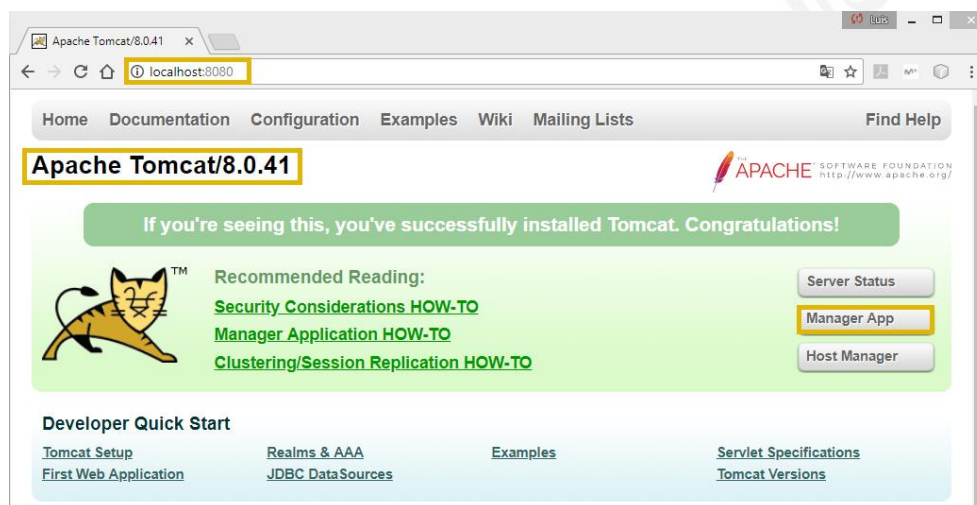


Figura 32 - Tomcat pantalla de bienvenida

Por último, conectar Tomcat con la base de datos **Apache Derby**. Para utilizar una base de datos en la aplicación hay dos alternativas.

1. Usar el API JDBC¹⁵ directamente, driver incrustado o embebido, no recomendado ya que cuando el número de usuarios es grande sobrecarga el servidor y puede significar la caída de la aplicación.

2. Configurar un “database connection pool”. La base de datos se ejecuta como un proceso aparte y la aplicación se conecta con ella mediante un socket, siendo preciso arrancarla y pararla.

Esta segunda opción es más correcta y es la que se ha utilizado puesto que mejora el rendimiento al crearse varios objetos de conexión con la base de datos que se reutilizan evitando crear un objeto conexión para cada acceso a la base de datos.

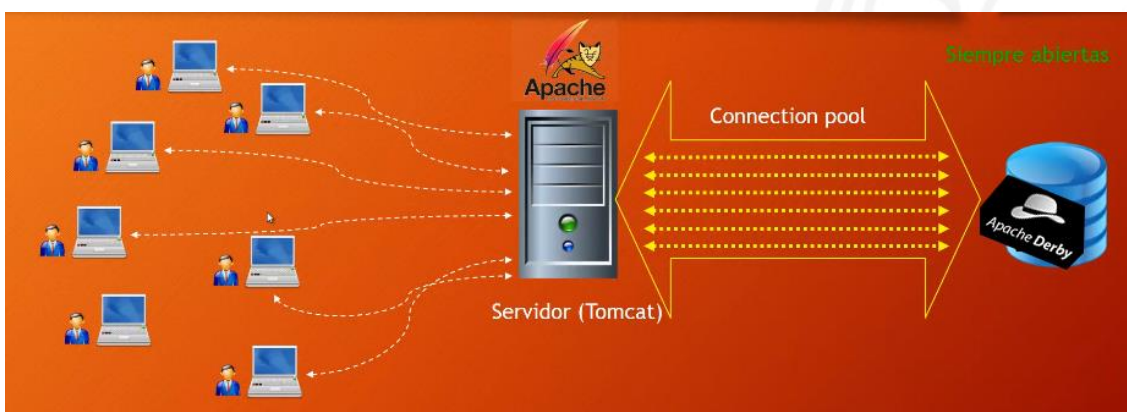


Figura 33 - Pool de conexiones

Para crear el pool de conexiones se necesita descargar el driver JDBC para Apache Derby llamado Derby.jar, colocar el driver en el apartado de librerías, definir el pool de conexiones en “META-INF/context.xml” y crear una referencia al pool de conexiones desde el código Java.

El driver JDBC permite conectar aplicaciones Java con la base de datos, como un puente. Establece la conexión y manipula la BBDD utilizando SQL. En nuestro caso se han copiado los drivers Derby.jar y DerbyClient.jar.

El siguiente paso es inicializar el servicio de Derby dando a iniciar servidor. Una vez iniciado podemos crear la base de datos, denominada

¹⁵ Java Database Connectivity es una API que permite la ejecución de operaciones sobre bases de datos desde Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

“TestOnline” en el presente TFG. Por simplicidad se ha establecido como nombre de usuario “root” y como contraseña “root”.

Por último, lo único que tenemos que hacer es conectar el servicio en la pestaña Services, dando botón derecho y *Connect* (conectar) en la base de datos TestOnline.

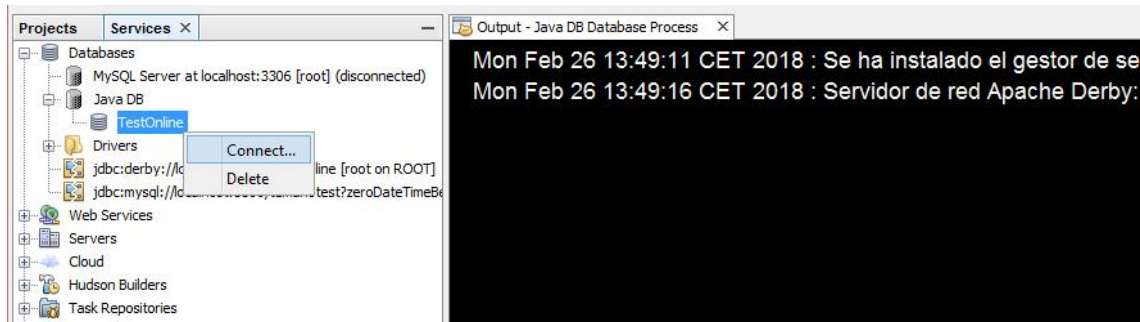


Figura 34 - Conexión Apache Derby

Una vez conectado tendremos que crear las tablas, para ello copiamos el texto de “*tsiDB.sql*” (se encuentra en la carpeta BD del CD adjunto), hacemos click derecho en “Tables” (tablas) y “*Execute Command...*”, pegamos la sentencia SQL y se ejecuta con el botón “*Run SQL*”.

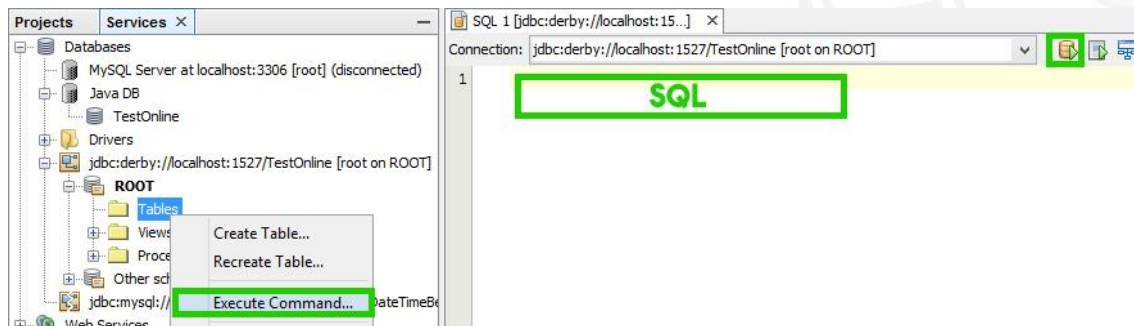


Figura 35 - Instrucciones crear tabla a partir de sentencia SQL

Una vez ejecutado observamos la tabla PREGUNTA, dentro del esquema ROOT, con cada uno de los campos que contienen toda la información de la aplicación. Para poblar la tabla con los datos seguimos los mismos pasos que en la figura anterior, pero con el script “*tsiDatos.sql*”.

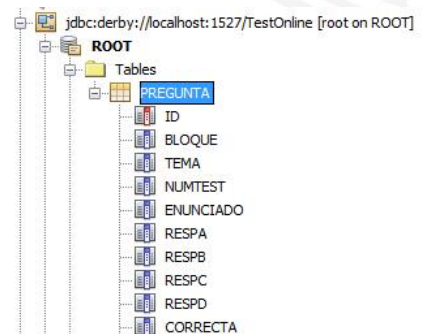


Figura 36 - Esquema tabla pregunta

APÉNDICE 2: MANUAL DE USUARIO

Una vez que sabemos cómo instalar la aplicación y hacerla funcionar, en esta sección se darán todas las indicaciones necesarias sobre su funcionamiento.

Entramos en la página principal:
<http://localhost:8080/www.tecnicosoporteinformatico.com/>

Lo primero que nos encontramos es la barra de navegación, la cual está fijada a la parte superior quedando siempre a la vista.



En la barra tenemos cuatro enlaces en la versión de escritorio o uno y el menú si es desde un dispositivo con menos resolución.

- **Técnico soporte informático** es un ancla que te sube o devuelve al inicio.
- **Test Online** te desplaza hacia el apartado del temario, las diferentes secciones y es el sitio para realizar los test.
- **Exámenes** te desplaza hacia el apartado de exámenes y soluciones de años anteriores.
- **Novedades** te desplaza hacia un timeline o línea del tiempo con las noticias más relevantes ordenadas cronológicamente.

Debajo nos encontramos con el logo con el escudo de la Junta de Castilla y León delante de un fondo donde aparece un examen de oposición. Podemos leer las ventajas que obtenemos con el uso de la web y llegamos al temario.

Tenemos **dos opciones de realizar test**. La primera opción es visitar el temario con sus diferentes secciones organizadas por colores. Toda la imagen que aparece con los temas de cada apartado es un enlace al menú de las secciones. Sirve como guía de los temas que contiene cada sección y del número de temas que tiene la oposición.



Si entramos en el enlace nos aparece la pantalla que podemos ver en la imagen inferior, con las secciones y la información referente a los tipos de test que existen.

Los normales de 20 preguntas fijas, que siempre serán las mismas, tienen el fondo gris con una raya inferior con el color de cada sección. De la base de datos se cargarán con su sección, tema y test correspondiente.

Los test de simulación de examen constan de 55 preguntas y tienen el fondo naranja. De esas 55 preguntas, cargaremos 10 preguntas de materias comunes aleatorias del banco de preguntas ('Sección = MC' y 'Tema=0'), luego 40 preguntas aleatorias de tipo examen ('Sección = EX') y por último 5 preguntas repetidas de los test normales que no sean de materias comunes.

El objetivo es realizar test online por temas, para preparar lo que estemos estudiando en ese momento. El procedimiento es muy sencillo, basta con elegir primero el apartado al que pertenezca el tema y luego el número de tema. La cantidad de test disponibles dentro de cada tema se irá ampliando con el paso del tiempo.

Test normal: 20 preguntas fijas (no cambian).

Test de simulación de examen: 55 preguntas (aleatorias).



Se han diferenciado para hacer los test de preparación más amenos y los test de examen para acostumbrarse a los tiempos y ser lo más parecidos posibles al examen real. Por esa misma razón se incluyen en los test de simulación de examen 10 preguntas de materias comunes y un temporizador a la hora de iniciar cada test.

Si bien la primera opción era visitar el temario y elegir la sección, la segunda opción es para usuarios que están preparando alguna sección en particular, ir directamente a ella y ahorrar tiempo.



El nombre de cada sección está escrito en su imagen. Al situarse encima con el ratón se desplaza la imagen hacia arriba apareciendo un enlace con la opción de realizar test online (letras amarillas) de la sección correspondiente.



Una vez dentro del apartado de materias comunes aparecen los temas correspondientes y el número de test disponibles para cada tema. En el caso de materias comunes 4 temas con su título y un test para cada tema.

La forma de numerar el número de test dentro de cada tema sigue el siguiente esquema:

$$[N^{\circ}\text{tema} . N^{\circ} \text{ de Test}]$$

En la imagen inferior se observa la sección de materias comunes, con sus temas y test correspondientes. Dicho número de test en las futuras evoluciones o desarrollos del trabajo se espera que aumente considerablemente. Actualmente como se observa en la figura inferior, en materias comunes tenemos dos test para los temas 1 y 4.

Temario > Materias Comunes



MATERIAS COMUNES

Tema 1.- La Constitución española de 1978: estructura y contenido.

TEST 1.1

TEST 1.2

Tema 2.- El Estatuto de Autonomía de Castilla y León. La Ley del Gobierno y de la Administración de Castilla y León.

TEST 2.1

Tema 3.- La Unión Europea. Las Instituciones comunitarias.

TEST 3.1

Tema 4.- El personal al servicio de la Administración de la Comunidad Autónoma de Castilla y León. El Convenio Colectivo para el personal laboral de la Administración General de la Comunidad de Castilla y León y Organismos autónomos dependientes de ésta.

TEST 4.1

TEST 4.2

A la hora de realizar Test Online para marcar o desmarcar preguntas simplemente hay que hacer clic en el apartado deseado (a lo largo de toda la línea) y el radiobutton con la respuesta quedará iluminado en color naranja. Con un segundo clic quedaría desmarcada.

1. Según la Constitución Española, no es un derecho fundamental:

- a) El derecho a la propiedad privada.
- b) El derecho a la vida.
- c) La libertad ideológica, religiosa y de culto.
- d) El derecho de asociación.

Una vez marcadas las respuestas se selecciona el botón "Corregir test". Dentro del marco principal que aparece en la evaluación tenemos la puntuación con dos decimales redondeados, el número de respuestas correctas en color verde, el número de respuestas incorrectas en color rojo, el número de respuestas sin contestar en color gris y el tiempo utilizado.

PUNTUACIÓN 0,00

Correctas: 1 | Incorrectas: 16 | Sin contestar: 3 | Tiempo utilizado: 02 min 53 sec

Color Verde = Respuesta correcta

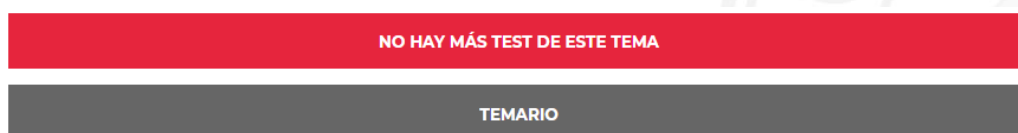
Color Rojo y verde = Respuesta corregida (Rojo incorrecta marcada por nosotros, Verde correcta)

Color Gris = Respuesta sin contestar

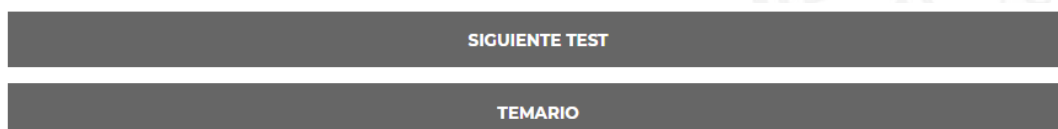
Se incluye una leyenda con los colores posibles en la corrección debajo del recuadro de la puntuación.

- **Color verde** hemos acertado la respuesta. **Suma un punto.** (En realidad es 0,5 para test de 20 preguntas y 0,18 en simulación de examen).
- **Color rojo** hemos fallado la respuesta, aparecerá en verde la respuesta corregida y en rojo la errada por nosotros. **Resta 1/3** de una respuesta correcta.
- **Color gris** no habíamos respondido así que **ni suma ni resta** y la solución aparece marcada en gris.

En la parte inferior de los test evaluados, aparecen varias opciones en función de si existen más test para el mismo tema o no. En el caso de ejemplo del tema 2 de materias comunes visto anteriormente, nos aparecería que no hay más test o que el que estamos haciendo es el último y la opción de regresar al temario.



Y en el caso de la evaluación del test 1 del tema 1, nos aparece la opción de realizar el siguiente test, hasta llegar al último test de cada uno de los temas. También la opción de regresar al temario.



En la página principal, el siguiente apartado después de los test online es el de los exámenes y soluciones de años anteriores. Si te quedas encima con el ratón cambia de color y al hacer clic, el examen correspondiente se abrirá en una nueva pestaña.

Está ordenado por nombre de oposición ya que se han colgado exámenes de otras oposiciones que pueden ser de interés, y por fecha descendente (lo más reciente arriba).


EXÁMENES

🔍 [TSI Examen 2016 - Ejercicio 1 turno libre.pdf](#)

🔍 [TSI Solución 2016 - Ejercicio 1 turno libre.pdf](#)

El último apartado es el **timeline** con las noticias más relevantes relacionadas con la oposición de técnico de soporte. También en orden descendente, consta de un título (Nuevo temario) que además es un enlace a la noticia completa en una nueva pestaña, la fecha y una descripción resumen.

NOVEDADES



Nuevo temario Febrero 2018

Temario provisional 2018. Temario provisional y orientativo, sujeto a posibles modificaciones. El temario definitivo será publicado en la propia convocatoria.

El pie de página consta de cuatro apartados muy sencillos. El primero un email de contacto por si tenemos cualquier duda o petición. El segundo un enlace a las redes sociales relacionadas con las oposiciones. Tercero una breve reseña del TFG y un enlace a la web personal del autor. Por último, enlaces a las páginas de sobre nosotros, mapa del sitio, preguntas frecuentes, términos y condiciones y política de privacidad.

CONTACTO

Email: tecnicosoporteinformaticojcy@gmail.com

REDES SOCIALES



TFG

Escuela Universitaria de Informática (SEGOVIA).
Portal web para la preparación de las oposiciones de
Técnico de Soporte Informático de la Junta de
Castilla y León.

luisorozco.dx.am

APÉNDICE 3: CONTENIDO DEL CD-ROM

Acompañando a esta memoria se adjunta un CD-ROM que contiene los instaladores de la aplicación, el código fuente y documentación.

Los directorios del CD son los siguientes:

- Carpeta **Código**: Contiene el código fuente completo del proyecto, exportado en un .zip directamente desde NetBeans.
- Carpeta **BD**: Contiene el archivo para crear la tabla pregunta de la base de datos llamado "tsiDB.sql" y el archivo para poblar la base de datos llamado "tsiDatos.sql".
- Carpeta **Programas**: Contiene los instaladores de los programas que se han usado para el desarrollo del TFG. (*Apache Tomcat 8.0.41, Sublime Text 3, Drivers JDBC de Apache Derby llamados Derby.jar y DerbyClient.jar y un enlace para descargar el pack de NetBeans+Tomcat+JDK correspondiente al sistema operativo donde instalemos el proyecto*).
- Carpeta **Memoria**: Contiene la memoria del proyecto en un fichero llamado memoria_luioz.pdf. También se ha incluido la presentación presentación_luioz.pptx.

En la figura inferior podemos ver una captura de pantalla del contenido del CD.

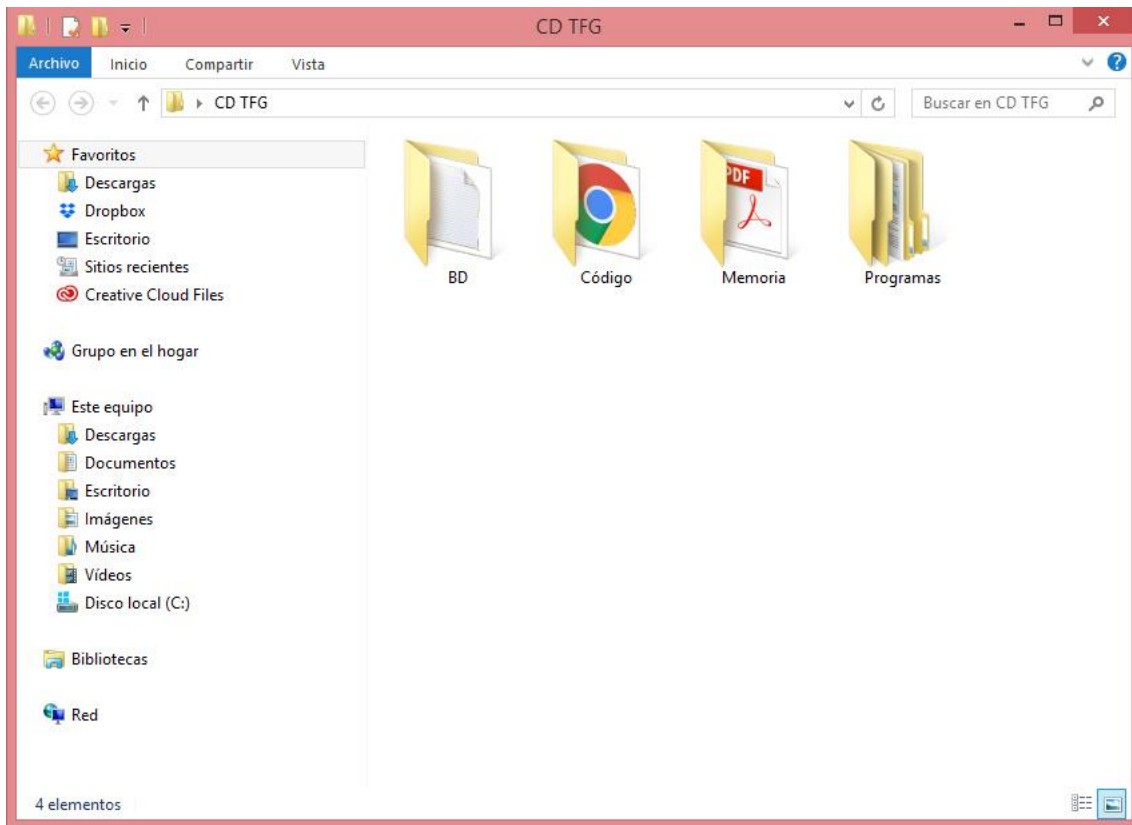


Figura 37 - Contenido CD-ROM

BLOQUE 4

BIBLIOGRAFÍA

IV

.BIBLIOGRAFÍA

- [1] Erich et Al.Gamma. "Patrones de diseño: Elementos de software orientado a objetos reutilizables". Addison Wesley, 2002. ISBN: 8478290591.
- [2] Durán Toro, Amador; Bernárdez Jiménez, Beatriz. "Metodología para el análisis de requisitos de sistemas Software". Universidad de Sevilla, 2001. [S.N.].
- [3] Larman, Craig. "UML y patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado". Pearson Alhambra, 2010. ISBN: 9788420534381.
- [4] Sommerville, Ian. "Ingeniería de software (sexta edición)". Prentice Hall, 2002. ISBN: 9789702602064.
- [5] Silberschatz, Abraham. "Fundamentos de bases de datos (quinta edición)". S.A. McGraw-Hill, 2006. ISBN: 9788448146443.
- [6] Álvarez García, Alonso. "Métodos ágiles y scrum". Anaya multimedia, 2012. ISBN: 9788441531048.

WEBGRAFÍA

- [1] M. Beedle, A; Van Bennekum, A; Cockburn, W; Cunningham, M; Fowler, A; Hunt, J. Kern; R. Jeffries. "Principles of the Agile Manifesto" [En línea, 2001]. Available:
<http://agilemanifesto.org/iso/es/manifesto.html>.
[Último acceso: 2017 noviembre].

- [2] P. Venkata Gayatri. "Agile User Stories". [En línea, 2013]. Available: <https://www.scrumalliance.org/community/articles/2013/september/agile-user-stories>. [Último acceso: 2017 noviembre].
- [3] Colaborativo." Scrum (Software Development)". [En línea, 2017]. Available: [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)). [Último acceso: 2017 octubre].
- [4] Softeng. "Metodología Scrum". [En línea, 2014]. Available: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>. [Último acceso: 2017 octubre].
- [5] OBS Busines School." Metodologías para la gestión de proyectos que más se utilizan". [En línea]. Available: <https://www.obs-edu.com/es/blog-project-management/administracion-de-proyectos/las-3-metodologias-para-la-gestion-de-proyectos-que-mas-se-utilizan> [Último acceso: 2017 octubre].
- [6] BBVA API_ Market. "Herramientas básicas para los desarrolladores HTML". [En línea, 2015]. Available: <https://bbvaopen4u.com/es/actualidad/herramientas-basicas-para-los-desarrolladores-en-html>. [Último acceso: 2017 septiembre].
- [7] Martínez, Sergio. "Metodología iterativa o incremental en la gestión de proyectos". [En línea, 2017]. Available: <http://mundoerp.com/blog/metodologia-iterativa-o-incremental-gestion-proyectos/>. [Último acceso: 2017 septiembre].
- [8] Pestana, José. "Matriz de probabilidad e impacto de riesgos". [En línea, 2017]. Available: <https://aulainteractiva.org/matiz-probabilidad-e-impacto-riesgos-4383/>. [Último acceso: 2018 enero].

- [9] Universidad EAFIT. "Medidas de tratamiento del riesgo". [En línea, 2017]. Available: <http://www.eafit.edu.co/escuelas/administracion/consultorio-contable/Documents/Nota%20de%20Clase%2010%20Medidas%20de%20Tratamiento%20del%20Riesgo.pdf>. [Último acceso: 2018 enero].
- [10] Abad, Paco. "Cómo escribir un Trabajo Fin de Grado". [En línea]. Available: <http://users.dsic.upv.es/~fjabad/pfc/comoEscribir.pdf> [Último acceso: 2017 septiembre].
- [11] Santiago Castro, Ariel. "Ejemplo de javascript corrigiendo un test de radio buttons". [En línea, 2014]. Available: <http://www.infosofia.es/ejemplo-de-javascript-corrigiendo-un-test-de-radio-buttons/>. [Último acceso: 2017 diciembre].
- [12] González, Ramón. "Scrum: para equipos extra pequeños de 1 a 3 personas". [En línea, 2015]. Available: <http://elcatalejoderami.blogspot.com.es/2015/02/scrum-para-equipos-extra-pequenos.html> [Último acceso: 2017 diciembre].
- [13] Berzal, Fernando. "El ciclo de vida de un sistema de información". [En línea]. Available: <http://flanagan.ugr.es/docencia/2005-2006/2/apuntes/ciclovida.pdf> [Último acceso: 2017 diciembre].
- [14] Laguna, Miguel Ángel. "Ingeniería del Software I". [En línea, 2010]. Available: <https://www.infor.uva.es/~mlaguna/is1/apuntes/1-intro.pdf> [Último acceso: 2017 octubre].
- [15] Figuerola, Norberto. "Riesgos: Plan de Mitigación vs Plan de Contingencia vs Fallback Plan". [En línea, 2015]. Available: <https://articulospm.files.wordpress.com/2015/06/riesgo-s-plan-mitigacion-vs-plan-contingencia-vs-fallback-plan.pdf>. [Último acceso: 2017 diciembre].

- [16] Goicochea, Aníbal. "Análisis de los riesgos de un proyecto". [En línea, 2012]. Available: <https://anibalgoicochea.com/2012/11/23/analisis-de-los-riesgos-de-un-proyecto/>. [Último acceso: 2017 diciembre].
- [17] De la Cruz, Christian." Firefox se convierte en el segundo navegador más usado". [En línea, 2018]. Available: <https://www.fayerwayer.com/2016/05/firefox-se-convierte-en-el-segundo-navegador-mas-usado/>. [Último acceso: 2017 diciembre].
- [18] Laguna, Miguel Ángel. "Ingeniería del Software I". [En línea, 2010]. Available: <https://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf> [Último acceso: 2018 enero].
- [19] Laguna, Miguel Ángel. "Ingeniería del Software I". [En línea, 2010]. Available: https://www.infor.uva.es/~mlaguna/is1/apuntes/3-casos_uso.pdf [Último acceso: 2018 enero].
- [20] Colaborativo. "Requisito (sistemas)". [En línea, 2018]. Available: [https://es.wikipedia.org/wiki/Requisito_\(sistemas\)#An%C3%A1lisis_de_requisitos](https://es.wikipedia.org/wiki/Requisito_(sistemas)#An%C3%A1lisis_de_requisitos) [Último acceso: 2018 enero].
- [21] Colaborativo. "Scrum Manager (v2.6)". [En línea, 2016]. Available: http://www.scrummanager.net/files/sm_proyecto.pdf [Último acceso: 2018 febrero].
- [22] Hernández, Raúl. "BDD Cucumber y Gherkin. Desarrollo dirigido por comportamiento." [En línea, 2016]. Available: <https://m.genbetadev.com/metodologias-de-programacion/bdd-cucumber-y-gherkin-desarrollo-dirigido-por-comportamiento> [Último acceso: 2018 febrero].

- [23] Llopis, Fernando. "Scrum. Historias de Usuario"
[En línea, 2016]. Available:
<http://fernandollopis.dlsi.ua.es/?p=39>
[Último acceso: 2018 febrero].
- [24] González, Guillermo. "¿Test de usabilidad?
Cuando tus visitas pueden dártela clave del diseño
de tu web". [En línea, 2015]. Available:
<https://es.jimdo.com/2015/02/03/tests-de-usabilidad-cuando-tus-visitas-pueden-darte-la-clave-del-dise%C3%B1o-de-tu-web/>
[Último acceso: 2018 febrero].
- [25] Walter.Lara.37 "Metodología Scrum: Cómo
funciona la metodología de trabajo Scrum". [En línea,
2015]. Available:
<https://platzi.com/blog/metodologia-scrum-fases/>
[Último acceso: 2018 febrero].