**ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

**GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN**

**ÁREA DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL**

**TRABAJO FIN DE GRADO Nº 17010059**

**MEJORA DEL RENDIMIENTO Y AMPLIACIÓN DE UNA WEB PARA LA GESTIÓN INTEGRAL DE CONGRESOS CIENTÍFICOS**

**D. VIGIL RODRÍGUEZ, Guillermo**

**TUTOR: D. RANILLA PASTOR, José**

**FECHA: Mayo 2018**

Índice de contenidos

[Índice de contenidos 3](#_Toc505426573)

[Índice de figuras 10](#_Toc505426574)

[Índice de tablas 13](#_Toc505426575)

[DOCUMENTO 1: MEMORIA 15](#_Toc505426576)

[Índice de contenidos 17](#_Toc505426577)

[1.1 Resumen 19](#_Toc505426578)

[1.2 Introducción 20](#_Toc505426579)

[1.3 Objetivos y alcance 21](#_Toc505426580)

[1.3.1 Objetivos de la aplicación 21](#_Toc505426581)

[1.3.2 Objetivos académicos y de aprendizaje 22](#_Toc505426582)

[1.4 Estudios y análisis previos 23](#_Toc505426583)

[1.4.1 Estudios de carácter teórico 23](#_Toc505426584)

[1.4.1.1 Aconf 24](#_Toc505426585)

[1.4.1.2 EasyChair 25](#_Toc505426586)

[1.4.1.3 Primoris 26](#_Toc505426587)

[1.4.1.4 Sistedes 27](#_Toc505426588)

[1.4.2 Estudios de carácter técnico 28](#_Toc505426589)

[1.4.2.1 Patrones de diseño 29](#_Toc505426590)

[1.4.2.1.1 Modelo 29](#_Toc505426591)

[1.4.2.1.2 Vista 29](#_Toc505426592)

[1.4.2.1.3 Controlador 29](#_Toc505426593)

[1.4.2.2 Lenguajes de Programación 31](#_Toc505426594)

[1.4.2.2.1 PHP 31](#_Toc505426595)

[1.4.2.2.2 Java 32](#_Toc505426596)

[1.4.2.2.3 Ruby 32](#_Toc505426597)

[1.4.2.2.4 Python 32](#_Toc505426598)

[1.4.2.2.5 Node.js 32](#_Toc505426599)

[1.4.2.3 Sistema de gestión de bases de datos 32](#_Toc505426600)

[1.5 Organización del documento 33](#_Toc505426601)

[1.5.1 Documento 1 – Memoria 33](#_Toc505426602)

[1.5.2 Documento 2 – Presupuesto y planificación 33](#_Toc505426603)

[1.5.3 Documento 3 – Requisitos de usuario y análisis de alternativas 34](#_Toc505426604)

[1.5.4 Documento 4 – Análisis de requisitos del sistema 34](#_Toc505426605)

[1.5.5 Documento 5 – Diseño del sistema 34](#_Toc505426606)

[1.5.6 Documento 6 – Pruebas 34](#_Toc505426607)

[1.5.7 Documento 7 – Seguridad 34](#_Toc505426608)

[1.5.8 Documento 8 – Manuales 34](#_Toc505426609)

[1.6 Conclusiones y ampliaciones 35](#_Toc505426610)

[1.6.1 Conclusiones 35](#_Toc505426611)

[1.6.2 Ampliaciones 35](#_Toc505426612)

[1.6.2.1 Mejora de la infraestructura 36](#_Toc505426613)

[1.6.2.2 Mantenimiento continuo de actualizaciones de *software* 36](#_Toc505426614)

[1.6.2.3 Diseño responsivo de la interfaz 36](#_Toc505426615)

[1.6.2.4 Internacionalización 36](#_Toc505426616)

[1.7 Bibliografía 37](#_Toc505426617)

[1.7.1 Libros y artículos 37](#_Toc505426618)

[1.7.2 Referencias en internet 37](#_Toc505426619)

[DOCUMENTO 2: PLANIFICACIÓN Y PRESUPUESTO 40](#_Toc505426620)

[Índice de contenidos 42](#_Toc505426621)

[2.1 Introducción 43](#_Toc505426622)

[2.2 Planificación temporal 43](#_Toc505426623)

[2.2.1 Fase de análisis 44](#_Toc505426624)

[2.2.2 Fase de diseño 44](#_Toc505426625)

[2.2.3 Fase de implementación 45](#_Toc505426626)

[2.2.4 Fase de pruebas 45](#_Toc505426627)

[2.3 Presupuesto 46](#_Toc505426628)

[2.3.1 Coste del material software/hardware 46](#_Toc505426629)

[2.3.2 Coste del personal 47](#_Toc505426630)

[2.3.3 Coste total 49](#_Toc505426631)

[DOCUMENTO 3: REQUISITOS DE USUARIO Y ANÁLISIS DE ALTERNATIVAS 50](#_Toc505426632)

[Índice de contenidos 52](#_Toc505426633)

[3.1 Introducción 54](#_Toc505426634)

[3.2 Identificación de usuarios participantes en el sistema 54](#_Toc505426635)

[3.2.1 Rol de estudiante 54](#_Toc505426636)

[3.2.2 Rol de empresa 54](#_Toc505426637)

[3.3 Requisitos de usuario 55](#_Toc505426638)

[3.3.1 Requisitos de usuario comunes 55](#_Toc505426639)

[3.3.2 Requisitos de usuario – rol de estudiante 55](#_Toc505426640)

[3.3.2 Requisitos de usuario – rol de empresa 56](#_Toc505426641)

[3.4 Análisis de alternativas 57](#_Toc505426642)

[3.4.1 Tecnologías back-end 57](#_Toc505426643)

[3.4.2 Tecnologías front-end 63](#_Toc505426644)

[3.4.2.1 HTML (HyperText Markup Language) 63](#_Toc505426645)

[3.4.2.2 XML (eXtensive Markup Language): 63](#_Toc505426646)

[3.4.2.3 Javascript 63](#_Toc505426647)

[3.4.2.4 AngularJS 63](#_Toc505426648)

[3.4.2.5 CSS (Cascading Style Sheets): 64](#_Toc505426649)

[3.4.2.6 Bootstrap 64](#_Toc505426650)

[3.4.3 Base de datos 64](#_Toc505426651)

[3.5. DESCRIPCIÓN DE LA ALTERNATIVA SELECCIONADA 66](#_Toc505426652)

[3.5.1 Tecnologías back-end 66](#_Toc505426653)

[3.5.2 Tecnologías front-end 68](#_Toc505426654)

[3.5.3 Base de datos y gestor de base de datos 69](#_Toc505426655)

[3.5.4 Sistema operativo 70](#_Toc505426656)

[DOCUMENTO 4: ANÁLISIS DE REQUISITOS DEL SISTEMA 71](#_Toc505426657)

[Índice de contenidos 73](#_Toc505426658)

[4.1 Introducción 75](#_Toc505426659)

[4.2 Subsistemas de análisis y casos de uso 75](#_Toc505426660)

[4.2.1 Acceso de usuarios 76](#_Toc505426661)

[4.2.1.1 Registro de usuario 76](#_Toc505426662)

[4.2.1.2 Login de usuario 77](#_Toc505426663)

[4.2.1.3 Recuperación de contraseña de usuario 78](#_Toc505426664)

[4.2.2 Gestión de la información de usuario 78](#_Toc505426665)

[4.2.2.1 Modificar información básica de cuenta de usuario 79](#_Toc505426666)

[4.2.2.2 Configurar perfil profesional de usuario 80](#_Toc505426667)

[4.2.2.3 Borrar cuenta de usuario 80](#_Toc505426668)

[4.2.3 Gestión de ofertas de prácticas 81](#_Toc505426669)

[4.2.3.1 Creación de ofertas de prácticas 82](#_Toc505426670)

[4.2.3.2. Inscripción en ofertas de prácticas 82](#_Toc505426671)

[4.2.3.3 Proceso de selección – Rol de estudiante 83](#_Toc505426672)

[4.2.3.4 Proceso de selección – Rol de empresa 84](#_Toc505426673)

[4.2.3.5 Búsqueda de estudiantes – Rol de empresa 85](#_Toc505426674)

[4.2.3.6 Búsqueda de empresas – Rol de estudiante 85](#_Toc505426675)

[4.2.4 Comunicación entre usuarios 86](#_Toc505426676)

[4.2.4.1 Envío de mensajes entre estudiantes y empresas 86](#_Toc505426677)

[4.2.4.2 Gestión de mensajes 86](#_Toc505426678)

[4.3 Requisitos no funcionales 87](#_Toc505426679)

[4.4 MODELO DE DATOS 88](#_Toc505426680)

[4.4.1 DESCRIPCIÓN DE LOS MODELOS 88](#_Toc505426681)

[4.4.1.1 Subsistema de sesión 88](#_Toc505426682)

[4.4.1.2 Subsistema de gestión de usuarios 89](#_Toc505426683)

[4.4.1.3 Subsistema de gestión de información de usuario 90](#_Toc505426684)

[4.4.1.4 Subsistema de gestión de ofertas de prácticas 92](#_Toc505426685)

[4.4.1.5 Subsistema de gestión de mensajes 93](#_Toc505426686)

[DOCUMENTO 5: DISEÑO DEL SISTEMA 96](#_Toc505426687)

[Índice de contenidos 98](#_Toc505426688)

[5.1 Introducción 100](#_Toc505426689)

[5.2 Diagrama de paquetes 100](#_Toc505426690)

[5.2.1 User 101](#_Toc505426691)

[5.2.2 Vistas 101](#_Toc505426692)

[5.2.3 Offers 101](#_Toc505426693)

[5.2.4 Offer\_inscriptions 101](#_Toc505426694)

[5.2.5 Messages 102](#_Toc505426695)

[5.2.6 Sessions 102](#_Toc505426696)

[5.3 Diagrama de clases 102](#_Toc505426697)

[5.4 Diagrama de interacción 103](#_Toc505426698)

[5.5 Diseño de la bases de datos 105](#_Toc505426699)

[5.5.1 Sistema Gestor de la Base de Datos 105](#_Toc505426700)

[5.5.2 Integración del SGBD en nuestro sistema 105](#_Toc505426701)

[5.6 Diseño de la interfaz 108](#_Toc505426702)

[5.6.1 Interfaces públicos 108](#_Toc505426703)

[5.6.1.1 Pantalla de bienvenida o home 109](#_Toc505426704)

[5.6.1.2 Pantalla de login 110](#_Toc505426705)

[5.6.1.3 Pantalla de recuperar contraseña 110](#_Toc505426706)

[5.6.1.4 Pantalla de registro 112](#_Toc505426707)

[5.6.2 Interfaces privados 114](#_Toc505426708)

[5.6.2.1 Interfaces del rol de estudiante 114](#_Toc505426709)

[5.6.2.1.1 Pantalla de inicio 114](#_Toc505426710)

[5.6.2.1.2 Pantalla de ofertas 115](#_Toc505426711)

[5.6.2.1.3 Pantalla de empresas y envío de mensajes 118](#_Toc505426712)

[5.6.2.1.4 Pantalla de perfil 121](#_Toc505426713)

[5.6.2.1.5 Configuración de cuenta 122](#_Toc505426714)

[5.6.2.1.6 Perfil académico 122](#_Toc505426715)

[5.6.2.1.7 Ofertas de usuario 125](#_Toc505426716)

[5.6.2.1.8 Mensajes 125](#_Toc505426717)

[5.6.2.2 Interfaces del rol de empresa 126](#_Toc505426718)

[5.6.2.2.1 Pantalla de inicio 126](#_Toc505426719)

[5.6.2.2.2 Pantalla de estudiantes y envío de mensajes 127](#_Toc505426720)

[5.6.2.2.3 Pantalla de ofertas 129](#_Toc505426721)

[5.6.2.2.4 Pantalla de perfil 129](#_Toc505426722)

[5.6.2.2.5 Pantalla de configuración de cuenta 130](#_Toc505426723)

[5.6.2.2.6 Pantalla de ofertas 131](#_Toc505426724)

[5.6.2.2.7 Pantalla de procesos de selección 132](#_Toc505426725)

[5.6.2.2.8 Pantalla de mensajes 135](#_Toc505426726)

[DOCUMENTO 6: PRUEBAS 137](#_Toc505426727)

[6.1 Introducción 139](#_Toc505426728)

[6.2 Pruebas unitarias 140](#_Toc505426729)

[6.3 Pruebas de integración y de sistema 142](#_Toc505426730)

[6.4 Pruebas de usabilidad y accesibilidad 148](#_Toc505426731)

[6.4.1 Criterios generales 149](#_Toc505426732)

[6.4.2 Criterios de identidad e información 150](#_Toc505426733)

[6.4.3 Criterios de lenguaje y redacción 150](#_Toc505426734)

[6.4.4 Criterios de rotulado 151](#_Toc505426735)

[6.4.5 Criterios de navegación 151](#_Toc505426736)

[6.4.6 Lay-out de la página 152](#_Toc505426737)

[6.4.7 Criterios de búsqueda 153](#_Toc505426738)

[6.4.8 Criterios de ayuda 154](#_Toc505426739)

[6.4.9 Accesibilidad 154](#_Toc505426740)

[6.4.10 Criterios de control y retroalimentación 155](#_Toc505426741)

[6.5 Pruebas de rendimiento 156](#_Toc505426742)

[DOCUMENTO 7: MANUALES 158](#_Toc505426743)

[Índice de contenidos 160](#_Toc505426744)

[7.1 Introducción 161](#_Toc505426745)

[7.2 Librerías 161](#_Toc505426746)

[7.2.1 Bootstrap 161](#_Toc505426747)

[7.2.2 Byebug 162](#_Toc505426748)

[7.2.3 jQuery 162](#_Toc505426749)

[7.2.4 Nested forms 162](#_Toc505426750)

[7.2.5 Paranoia 162](#_Toc505426751)

[7.2.6 PostgreSQL 162](#_Toc505426752)

[7.2.7 PostgreSQL Search 163](#_Toc505426753)

[7.2.8 RailRoady 163](#_Toc505426754)

[7.2.9 Rspec 163](#_Toc505426755)

[7.2.10 Will\_paginate 163](#_Toc505426756)

[7.3 Manual de instalación y ejecución 163](#_Toc505426757)

[7.3.1 Base de datos 163](#_Toc505426758)

[7.3.2 Ruby Version Manager 164](#_Toc505426759)

[7.3.3 Ejecución 165](#_Toc505426760)

Índice de figuras

[Figura 1 Página inicial de Aconf 28](#_Toc505427336)

[Figura 2 Página inical de EasyChair 29](#_Toc505427337)

[Figura 3 Página inicial de Primoris 30](#_Toc505427338)

[Figura 4 Página inicial de Sistedes 31](#_Toc505427339)

[Figura 6 Diagrama Gantt 53](#_Toc505427340)

[Figura 7 Diagrama Gantt - Fase de análisis 54](#_Toc505427341)

[Figura 8 Diagrama Gantt - Fase de diseño 54](#_Toc505427342)

[Figura 9 Diagrama Gantt - Fase de implementación 55](#_Toc505427343)

[Figura 10 Diagrama Gantt - Fase de pruebas 55](#_Toc505427344)

[Figura 11 Logo PHP 68](#_Toc505427345)

[Figura 12 Logo Ruby 68](#_Toc505427346)

[Figura 13 Logo Python 69](#_Toc505427347)

[Figura 14 Logo Java EE 70](#_Toc505427348)

[Figura 15 Gráfico benchmark de rendimiento. 71](#_Toc505427349)

[Figura 16 Gráfico de preguntas realizadas en Internet. 72](#_Toc505427350)

[Figura 17 Logo Bootstrap 74](#_Toc505427351)

[Figura 18 Gráfico comparativo de rendimiento. 75](#_Toc505427352)

[Figura 19 Logo del fraemework Ruby On Rails 78](#_Toc505427353)

[Figura 20 Logos HTML5, Javascript y CSS3 78](#_Toc505427354)

[Figura 21 Logo PostgreSQL 80](#_Toc505427355)

[Figura 22 Logo Ubuntu 14.04 LTS 80](#_Toc505427356)

[Figura 23 Subsistema acceso de usuarios 86](#_Toc505427357)

[Figura 24 Subsistema de gestión de la información de usuario 89](#_Toc505427358)

[Figura 25 Gestión de las ofertas de prácticas. 91](#_Toc505427359)

[Figura 26 Diagrama E/R de la base de datos 104](#_Toc505427360)

[Figura 27 Diagrama de paquetes del sistema 110](#_Toc505427361)

[Figura 28 Diagrama de clases de la aplicación. 113](#_Toc505427362)

[Figura 29 Diagrama de interacción sistema-usuario 114](#_Toc505427363)

[Figura 30 Captura de pantalla de la pantalla principal de la aplicación 119](#_Toc505427364)

[Figura 31 Captura de pantalla de la pantalla de login de la aplicación. 120](#_Toc505427365)

[Figura 32 Pantalla de recuperar contraseña 1/2. 121](#_Toc505427366)

[Figura 33Pantalla de recuperar contraseña 2/2. 121](#_Toc505427367)

[Figura 34 Captura de pantalla de la pantalla de registro principal de la aplicación 122](#_Toc505427368)

[Figura 35 Captura de pantalla de la pantalla de registro de estudiante. 123](#_Toc505427369)

[Figura 36 Captura de pantalla de la pantalla de registro de empresa. 123](#_Toc505427370)

[Figura 37 Captura de pantalla de la pantalla de inicio con el rol de estudiante. 125](#_Toc505427371)

[Figura 38 Captura de pantalla de la pantalla de oferta de empresa. 126](#_Toc505427372)

[Figura 39 Captura de pantalla de la pantalla de detalle de la oferta seleccionada. 127](#_Toc505427373)

[Figura 40 Captura de pantalla de la pantalla de detalle de la oferta seleccionada una vez hecha la inscripción. 128](#_Toc505427374)

[Figura 41 Captura de pantalla de la pantalla principal de empresas con el rol de estudiante. 129](#_Toc505427375)

[Figura 42 Captura de pantalla de la pantalla de detalle de la empresa seleccionada. 129](#_Toc505427376)

[Figura 43 Captura de pantalla de la ventana modal de envío de mensaje. 130](#_Toc505427377)

[Figura 44 Captura de pantalla de la pantalla de la empresa seleccionada una ve enviado el mensaje. 131](#_Toc505427378)

[Figura 45 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis Datos”. 132](#_Toc505427379)

[Figura 46 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mi CV”. 134](#_Toc505427380)

[Figura 47 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis Ofertas”. 135](#_Toc505427381)

[Figura 48 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis Mensajes”. 136](#_Toc505427382)

[Figura 49 Captura de pantalla de la pantalla de inicio con el rol de empresa. 137](#_Toc505427383)

[Figura 50 Captura de pantalla de la pantalla principal de estudiantes con el rol de empresa. 138](#_Toc505427384)

[Figura 51 Captura de pantalla de la pantalla de detalle del estudiante seleccionado. 139](#_Toc505427385)

[Figura 52 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis datos”. 140](#_Toc505427386)

[Figura 53 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis ofertas” con el formulario de crear nueva oferta abierto. 141](#_Toc505427387)

[Figura 54 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis procesos de selección” con el contenedor de la fase de preselección de una oferta abierto. 143](#_Toc505427388)

[Figura 55 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis procesos de selección” con el contenedor de la fase final de selección de una oferta abierto. 144](#_Toc505427389)

[Figura 56 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis procesos de selección” con la ventana modal de perfil del estudiante seleccionado abierta. 145](#_Toc505427390)

Índice de tablas

[TABLA 1. COMPARATIVA ENTRE MONGODB Y MYSQL 57](#_Toc486816833)

[TABLA 2. MODELO DE SESIÓN. 78](#_Toc486816834)

[TABL3. MODELO DE USUARIO 78](#_Toc486816835)

[TABLA 4. MODELO DE ROL. 79](#_Toc486816836)

[TABLA 5. MODELO DE ESTUDIOS 79](#_Toc486816837)

[TABLA 6. MODELO DE UNIVERSIDAD 80](#_Toc486816838)

[TABLA 7. MODELO DE PROVINCIA 80](#_Toc486816839)

[TABLA 8. MODELO DE CV\_SKILLS 81](#_Toc486816840)

[TABLA 9. MODELO DE OFERTA 81](#_Toc486816841)

[TABLA 10. MODELO DE INSCRIPCIÓN EN OFERTA (PROCESO DE SELECCIÓN) 82](#_Toc486816842)

[TABLA 11. MODELO DE MENSAJES 82](#_Toc486816843)

[TABLA 12. PRUEBAS DE INTEGRACIÓN Y SISTEMA 128](#_Toc486816844)

[TABLA 13. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS GENERALES 133](#_Toc486816845)

[TABLA 14. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE IDENTIFICACIÓN E INFORMACIÓN 135](#_Toc486816846)

[TABLA 15. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE LENGUAJE Y REDACCIÓN 135](#_Toc486816847)

[TABLA 16. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE ROTULADO 136](#_Toc486816848)

[TABLA 17. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE NAVEGACIÓN 136](#_Toc486816849)

[TABLA 18. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE LAYOUT DE LA PÁGINA 137](#_Toc486816850)

[TABLA 19. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE BÚSQUEDA 138](#_Toc486816851)

[TABLA 20. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE AYUDA 139](#_Toc486816852)

[TABLA 21. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE ACCESIBILIDAD 139](#_Toc486816853)

[TABLA 22. PRUEBAS DE USABILIDAD Y ACCESIBILIDAD - CRITERIOS DE CONTROL Y RETROALIMENTACIÓN 140](#_Toc486816854)

# DOCUMENTO 1: MEMORIA

**D. VIGIL RODRÍGUEZ, Guillermo**

**TUTOR: D. RANILLA PASTOR, José**

**FECHA: Mayo 2018**

Índice de contenidos

[DOCUMENTO 1: MEMORIA 15](#_Toc505427149)

[Índice de contenidos 17](#_Toc505427150)

[1.1 Resumen 24](#_Toc505427151)

[1.2 Introducción 25](#_Toc505427152)

[1.3 Objetivos y alcance 26](#_Toc505427153)

[1.3.1 Objetivos de la aplicación 26](#_Toc505427154)

[1.3.2 Objetivos académicos y de aprendizaje 27](#_Toc505427155)

[1.4 Estudios y análisis previos 28](#_Toc505427156)

[1.4.1 Estudios de carácter teórico 28](#_Toc505427157)

[1.4.1.1 Aconf 29](#_Toc505427158)

[1.4.1.2 EasyChair 30](#_Toc505427159)

[1.4.1.3 Primoris 31](#_Toc505427160)

[1.4.1.4 Sistedes 32](#_Toc505427161)

[1.4.2 Estudios de carácter técnico 33](#_Toc505427162)

[1.4.2.1 Patrones de diseño 34](#_Toc505427163)

[1.4.2.1.1 Modelo 34](#_Toc505427164)

[1.4.2.1.2 Vista 34](#_Toc505427165)

[1.4.2.1.3 Controlador 34](#_Toc505427166)

[1.4.2.2 Lenguajes de Programación 36](#_Toc505427167)

[1.4.2.2.1 PHP 36](#_Toc505427168)

[1.4.2.2.2 Java 37](#_Toc505427169)

[1.4.2.2.3 Ruby 37](#_Toc505427170)

[1.4.2.2.4 Python 37](#_Toc505427171)

[1.4.2.2.5 Node.js 37](#_Toc505427172)

[1.4.2.3 Sistema de gestión de bases de datos 37](#_Toc505427173)

[1.5 Organización del documento 38](#_Toc505427174)

[1.5.1 Documento 1 – Memoria 38](#_Toc505427175)

[1.5.2 Documento 2 – Presupuesto y planificación 38](#_Toc505427176)

[1.5.3 Documento 3 – Requisitos de usuario y análisis de alternativas 39](#_Toc505427177)

[1.5.4 Documento 4 – Análisis de requisitos del sistema 39](#_Toc505427178)

[1.5.5 Documento 5 – Diseño del sistema 39](#_Toc505427179)

[1.5.6 Documento 6 – Pruebas 39](#_Toc505427180)

[1.5.7 Documento 7 – Seguridad 39](#_Toc505427181)

[1.5.8 Documento 8 – Manuales 39](#_Toc505427182)

[1.6 Conclusiones y ampliaciones 40](#_Toc505427183)

[1.6.1 Conclusiones 40](#_Toc505427184)

[1.6.2 Ampliaciones 40](#_Toc505427185)

[1.6.2.1 Mejora de la infraestructura 41](#_Toc505427186)

[1.6.2.2 Mantenimiento continuo de actualizaciones de *software* 41](#_Toc505427187)

[1.6.2.3 Diseño responsivo de la interfaz 41](#_Toc505427188)

[1.6.2.4 Internacionalización 41](#_Toc505427189)

[1.7 Bibliografía 42](#_Toc505427190)

[1.7.1 Libros y artículos 42](#_Toc505427191)

[1.7.2 Referencias en internet 42](#_Toc505427192)

1.1 Resumen

El presente trabajo fin de grado consiste en la mejora del rendimiento y ampliación de una web ya existente y en producción, dedicada a la gestión integral de congresos científicos. En particular, la implementación realizada tiene como objeto la gestión del congreso *International Conference on Computational and Mathematical Methods in Science and Engineering* (de aquí en adelante CMMSE), que se celebra en julio de cada año en Rota (Cádiz).

Parte de la aplicación web ha sido desarrollada por un antiguo alumno de la escuela politécnica de ingeniería de Gijón. Dicha aplicación lleva en uso desde 2014, y ha sustituido a otra aplicación previa de objetivo similar. Cientos de profesionales científicos han usado la aplicación hasta día de hoy.

En primer lugar, se han actualizado las tecnologías existentes en el proyecto para optimizar la seguridad y el rendimiento tanto del servidor como de la aplicación. En segundo lugar, se ha mejorado la interfaz de usuario y la usabilidad de la aplicación, además de añadir diferentes nuevas funcionalidades a petición de la directiva del CMMSE.

La aplicación web funcionará principalmente como una herramienta de gestión integral de congresos científicos celebrados una vez al año. Los usuarios serán capaces de registrarse para acudir al congreso anual, además de poder subir archivos a la plataforma con contenido científico para presentarlos en dicho congreso. También podrán seguir el proceso de aceptación de archivos del congreso, y el alta y modificación de sus datos y archivos. Otra parte importante de la aplicación será la visión del usuario administrador de los datos relativos a los usuarios registrados y todo lo relacionado al congreso.

Las herramientas que se han escogido para el desarrollo de esta aplicación web han sido el lenguaje de programación PHP, lenguajes como HTML5, CSS3, Javascript o jQuery se han utilizado para satisfacer las necesidades del *front-end* del proyecto, garantizando la mejor experiencia de usuario posible. En último lugar, el sistema de gestión de bases de datos que se usará para almacenar y gestionar la información de la aplicación web es MySQL. Por parte del servidor, la aplicación ha sido desarrollada a medida para ejecutarse sobre un servidor Linux con Apache, teniendo en cuenta las particularidades del congreso a gestionar y del entorno *hardware* y *software* disponible.

1.2 Introducción

Este trabajo tiene como objetivo la ampliación de desarrollo y mejora de una aplicación web para la gestión de una conferencia científico-técnica sobre la aplicación de métodos matemáticos en problemas de ciencia e ingeniería que se celebra desde el año 2001 ininterrumpidamente. La aplicación desarrollada tiene como finalidad la gestión completa del congreso y su administración.

El proyecto surge con el objetivo de mejorar una aplicación web existente que la organización venía utilizando en ediciones anteriores del congreso. Para ello se ha realizado un estudio de las funcionalidades cubiertas por la aplicación original, además de considerar e implementar numerosas mejoras relativas a la usabilidad, a los tiempos de respuesta de la aplicación y al uso de componentes *software* actuales, utilizando librerías de código abierto *(open source)* ampliamente testadas por diferentes comunidades de desarrolladores de aplicaciones.

La aplicación desarrollada cubre todos los aspectos principales de la gestión de la conferencia, desde el registro de los asistentes y el envío de los artículos a presentar, hasta la generación de facturas, justificantes de pago, asistencia y presentación de ponencias, además del envío de notificaciones por correo electrónico tanto a los asistentes como a los gestores de la misma. También cubre otros aspectos administrativos, como la administración de documentos oficiales del congreso, la generación de estadísticas y listados, la verificación de pagos y envíos o la gestión de aspectos logísticos, como la preparación de etiquetas identificativas de distintos tipos para los asistentes y el personal.

En este primer capítulo correspondiente a la memoria del proyecto se plantean en primer lugar los objetivos que se pretenden conseguir con su realización y su alcance. A continuación, se expondrán los estudios y análisis realizados, así como la organización general de documentos que sigue este trabajo fin de grado. Acto seguido se detallarán las conclusiones y ampliaciones tras haber realizado el trabajo de desarrollo del sistema y finalmente se expondrá la bibliografía consultada.

1.3 Objetivos y alcance

A continuación, se expondrán los principales objetivos que se pretenden alcanzar con el desarrollo de este trabajo fin de grado, así como el alcance de este.

Se dividirá este apartado en dos, mostrando en primer lugar los objetivos principales de la aplicación que se va a desarrollar, los cuales detallarán el servicio que se pretende proporcionar a aquellos usuarios que utilicen la aplicación. En segundo lugar, se mostrarán aquellos objetivos de ámbito académico que se pretenden conseguir durante el desarrollo de esta.

1.3.1 Objetivos de la aplicación

El principal objetivo de este trabajo fin de grado es desarrollar una ampliación de una plataforma web dirigida a las personas que asistan al CMMSE y a todas las personas que la dirigen.

De esta forma los usuarios finales podrán usar la plataforma online con todos los requisitos de seguridad necesarios y con las últimas actualizaciones de todas las tecnologías que se usan en la actualidad. Su interfaz será mucho más intuitiva permitiendo una navegación más cómoda para todos los usuarios. El aspecto contará con una apariencia moderna respetando las peticiones del CMMSE.

La usabilidad de la plataforma será sencilla, así como práctica, ofreciendo al conferenciante la posibilidad de inscribirse al congreso y administrar todos sus datos y archivos de manera online, sin necesidad de realizar ninguna gestión presencial.

Por otro lado, la dirección del CMMSE utilizará la aplicación para gestionar todo el congreso de forma online. Desde ella podrá consultar rápidamente cualquier dato de los usuarios, de los artículos que se van a presentar o de las facturas generadas, entre otras muchas cosas.

La aplicación web será accesible a través de Internet, tanto para conferenciantes como para el personal del CMMSE, accediendo por medio de un navegador web a la siguiente dirección: <http://pirweb.edv.uniovi.es/webcmmse/index.php>.

1.3.2 Objetivos académicos y de aprendizaje

Otro de los objetivos principales de este trabajo fin de grado es desarrollar las nuevas funcionalidades de la aplicación con los lenguajes de programación actuales destinados al desarrollo web.

Debido al crecimiento de este tipo de aplicaciones estos últimos años, el conocimiento y uso de este tipo de lenguajes se ha convertido en una de las competencias profesionales más demandadas e importantes en el sector de la programación web hoy en día.

Además, se pretende cumplir con unos objetivos mínimos de seguridad en el servidor de producción y en las tecnologías usadas para el desarrollo de la aplicación.

Algunos de los objetivos académicos y de aprendizaje que se pretenden conseguir con el desarrollo de este trabajo fin de grado son:

* Desarrollar una ampliación de funcionalidades de una aplicación web ya existente, con los lenguajes y tecnologías que se utilizan en la aplicación en producción.
* Utilizar el patrón de diseño arquitectónico modelo-vista-controlador (MVC) durante el desarrollo de las nuevas funcionalidades de la aplicación.
* Utilizar un sistema de gestión de bases de datos más usados en el ámbito del desarrollo web.
* Utilizar y ampliar conocimiento sobre técnicas o herramientas de desarrollo web de gran importancia en la actualidad, así como de lenguajes de estilos para la interfaz de la aplicación.
* Realizar un estudio de seguridad de la aplicación, de su infraestructura y de las tecnologías y lenguajes usados.
* Aplicar las técnicas necesarias para blindar el servidor de producción y la aplicación ante cualquier posible ataque o uso inapropiado del sistema.

1.4 Estudios y análisis previos

A continuación, se presentarán aquellas plataformas, tecnologías o herramientas más modernas, potentes y utilizadas hoy en día en el mercado de las aplicaciones web. Se dividirá este apartado en dos; estudios de carácter teórico y estudios de carácter técnico, lo que permitirá tener una primera visión de los lenguajes, *frameworks*, bases de datos, aplicaciones ya desarrolladas u otras herramientas las cuales se aplicarán más detalladamente en los siguientes apartados de este trabajo fin de grado.

1.4.1 Estudios de carácter teórico

En los últimos años el desarrollo de aplicaciones web se ha incrementado notablemente, por lo que la mayoría de aplicaciones que podemos encontrar en la web han sido mejoradas y actualizadas. Entre ellas, también se incluyen aquellas plataformas web dedicadas a la gestión de congresos científico-tecnológicos. Cada vez son más las personas que utilizan estas plataformas con el objetivo de asistir a congresos internacionales y así gestionar la asistencia desde su lugar de residencia.

Plataformas como Aconf, EasyChair, o Primoris ofertan diversos servicios para gestionar eventos o conferencias como la CMMSE. Entre sus servicios está el alojamiento de la aplicación, la gestión de la base de datos, su desarrollo y despliegue y su personalización.

En este tipo de aplicaciones web las organizaciones se registran y contratan un servicio que en algunos casos puede ser gratuito. Se trata de un contrato de software para gestionar conferencias, con el objetivo de ahorrarse el desarrollo de la aplicación por parte de la organización del congreso. Sin embargo, esto tiene la desventaja de no poder ofrecer a los usuarios de la aplicación todas las funcionalidades deseadas o totalmente adaptadas a lo que se necesita. La organización contratante deberá ajustarse al servicio contratado.

Por otro lado, estas plataformas se ofertan en algunos casos gratuitamente con unos servicios mínimos, con la idea de atraer clientes que más tarde necesitarán alguno de los servicios de pago. Es el caso de Aconf y EasyChair.

Otra de las posibilidades existentes en el mercado es el desarrollo de la aplicación de gestión del congreso por parte de la propia organización que lo imparte. Un ejemplo es Sistedes, congreso científico técnico de Ingeniería y Tecnologías del Software.

A continuación, se hará una breve descripción de las plataformas anteriormente mencionadas, así como las herramientas con las que se han desarrollado.

1.4.1.1 Aconf (<https://www.aconf.org>)

Creada oficialmente en el año 2013, esta plataforma es gratuita para congresos académicos. Se trata de un *service as a software* (*Saas*), diseñado para mejorar la eficiencia de la gestión de congresos y conferencias antes, durante y después de su ejecución.

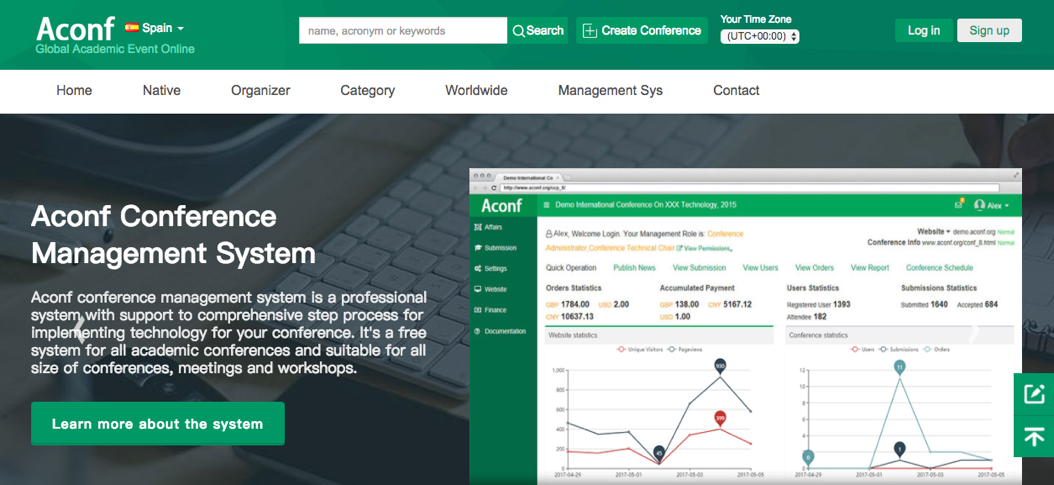


Figura 1 Página inicial de Aconf

Respecto a las tecnologías utilizadas, está basado en una herramienta WYSIWYG (*What You See Is What You Get*). Se trata de un sistema de gestión de contenidos (CMS) que permite escribir un documento viendo directamente el resultado final, es decir, se genera una vista preliminar del código HTML que luego el programa se encarga de generar con su código fuente HTML. En cuanto al resto de tecnologías, las mantienen en secreto.

Entre las ventajas de esta aplicación está la creación de plataformas de gestión de congresos académicos totalmente gratuitos. Su principal objetivo es promover la comunicación académica y construir comunidades académicas.

Una de las desventajas obvias es que no permiten la creación de plataformas no dedicadas a propósitos académicos.

1.4.1.2 EasyChair ([www.easychair.org](http://www.easychair.org))

Junto a Aconf, EasyChair es otro de los portales web más famosos para gestionar congresos. Se encuentra en el mercado desde el 2002, con más de dos millones de usuarios y más de 60.000 conferencias en sus servidores. Al igual que Aconf, oferta un servicio gratuito para conferencias académicas. En cambio, en su web ofertan un servicio mejor para aquellos que necesiten más prestaciones.

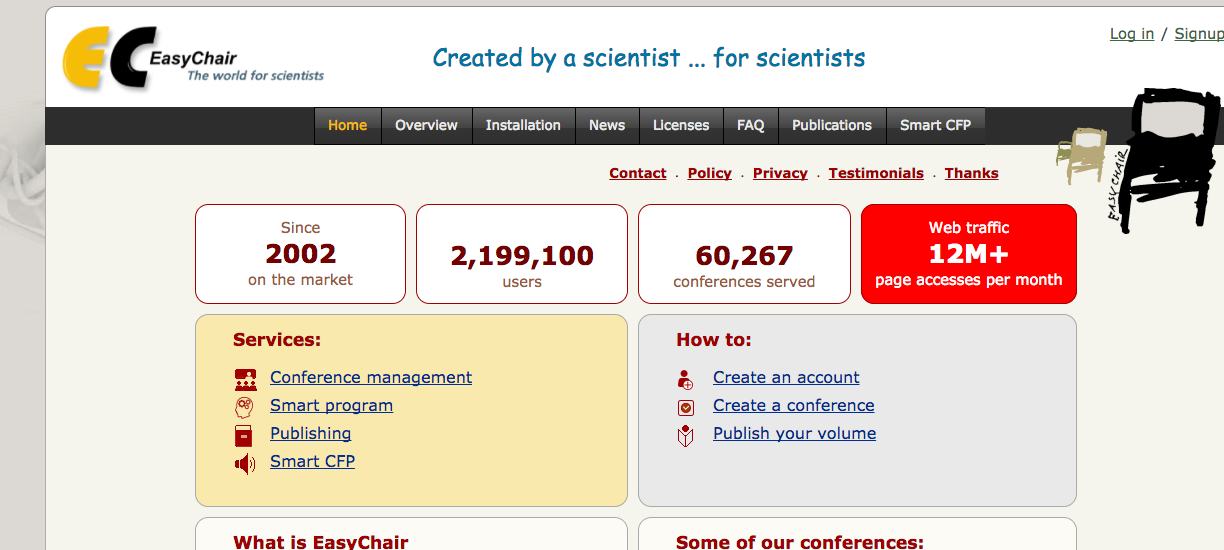


Figura 2 Página inical de EasyChair

El lenguaje para el desarrollo de esta plataforma es PHP. Al igual que Aconf, el resto de tecnologías usadas se mantienen en secreto.

En relación con Aconf, el propósito de sus servicios es similar, con la diferencia de que EasyChair ofrece un servicio de creación de aplicación para aquellas organizaciones no académicas que deseen organizar un congreso.

1.4.1.3 Primoris ([www.insticc.org](https://www.insticc.org))

Esta plataforma que se comercializa con el slogan de la primera de la nueva generación de sistemas de gestión de eventos soporta todo el proceso de gestión de congresos como Aconf y EasyChair. Con una interfaz moderna, gestiona todas las operaciones necesarias para el *front-end* y *back-end* de la aplicación. Organizada en módulos, tiene diferentes alternativas de configuración y personalización.

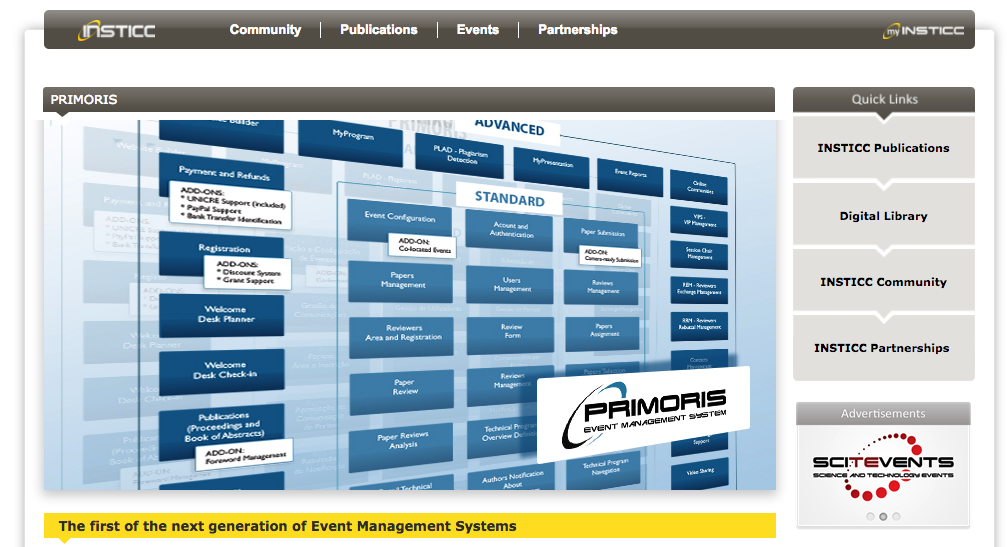


Figura 3 Página inicial de Primoris

La plataforma ha sido desarrollada mediante el entorno de aplicaciones web ASP.NET usando el *framework* .NET de Microsoft.

La diferencia con las otras plataformas ya comentadas es que Primoris no ofrece públicamente sus tarifas. Es por ello por lo que para contratar uno de sus servicios hay que entrar en contacto con el departamento de eventos. Dos de los congresos que utilizan esta plataforma son:

* <http://www.icsoft.org/>
* <http://www.enase.org/>

1.4.1.4 Sistedes (<https://fg.ull.es/sistedes2017>)

Esta plataforma perteneciente a la Universidad de La Laguna de Tenerife fue desarrollada con el objetivo de organizar las jornadas de la Sociedad de Ingeniería del Software y Tecnologías de Desarrollo de Software (SISTEDES). Este evento científico-técnico de carácter nacional engloban diferentes conferencias nacionales sobre ingeniería del software, bases de datos, ciencia e ingeniería de servicios y programación y lenguajes.



Figura 4 Página inicial de Sistedes

La principal diferencia con el resto de plataformas es que está gestionada únicamente para la misma organización, por lo que las tecnologías seleccionadas para su desarrollo son las óptimas para la gestión de las conferencias. En cambio, su desarrollo es más largo.

En cuanto a las tecnologías, Sistedes fue desarrollada con WordPress, un sistema de gestión de contenidos (CMS) muy utilizado a día de hoy. WordPress está basado en PHP y MySQL ejecutado en un servidor Apache. En el lado del cliente, WordPress proporciona un amplio abanico de plantillas creadas por diversos desarrolladores. Estas plantillas suelen estar hechas con código JavaScript, jQuery, HTML5 y CSS3. Algunas de estas plantillas son gratuitas pero la mayoría son de pago.

Sin duda, Sistedes es una de las plataformas web más parecidas al tipo de aplicación que se desea desarrollar en este trabajo fin de grado, ya que su finalidad es ajustar las funcionalidades deseadas por el CMMSE al desarrollo, de modo que no haya restricción a la hora de personalizar la interfaz y la lógica de la aplicación.

Tras analizar varias de las plataformas web más conocidas y usadas actualmente para la gestión de congresos, podemos llegar a la conclusión de que una de las mejoras respecto a otras aplicaciones similares será la usabilidad de la aplicació. Para ello se mejorará la interfaz existente y se implementará un estilo definido para toda la aplicación que ayudará a mejorar la experiencia de usuario.

Una vez expuestas las plataformas anteriormente mencionadas, se hará una recopilación de las mejores y más populares herramientas usadas en la actualidad para desarrollar una aplicación web, ya sean lenguajes de programación web, *frameworks* o lenguajes de bases de datos.

1.4.2 Estudios de carácter técnico

En primer lugar, se diferenciará entre *framework* y lenguaje de programación. En general, un *framework* es aquel entorno o estructura software que cuenta con aquellos componentes necesarios para realizar una aplicación, un *framework* podrá estar desarrollado por uno o más lenguajes de programación distintos y normalmente están estructurados con patrones de diseño. Estos componentes del *framework* se encargan de facilitar la tarea al desarrollador ya que agilizan el desarrollo de una aplicación.

Por otra parte, los lenguajes de programación son utilizados por los desarrolladores para realizar aquellas acciones que se desean ejecutar, es decir, es el lenguaje que interactúa directamente con la máquina.

Por último, se hará hincapié en los gestores de bases de datos, parte fundamental de cualquier sistema informático, encargados del almacenamiento de la información y de su tratamiento.

A continuación, se expondrán más detalladamente estos conceptos.

1.4.2.1 Patrones de diseño

En *frameworks* de ámbito web, es imprescindible utilizar uno de los patrones de arquitectura de software más conocidos, el patrón MVC (modelo-vista-controlador), patrón de diseño por el cual se rigen la mayoría de *frameworks* de desarrollo web.

Como se puede ver en la figura 5, la principal característica del patrón MVC es que está basado en la división de capas y/o conceptos, organizando la aplicación en tres modelos separados: modelo, vista y controlador.

1.4.2.1.1 Modelo

Esta capa se encarga de comunicarse con la base de datos mediante las correspondientes peticiones que hace un usuario, pudiendo acceder a los datos, así como también modificarlos, eliminarlos o crearlos si es preciso.

1.4.2.1.2 Vista

Capa encargada de tratar el código de la interfaz de usuario para presentar los datos que el modelo proporciona. Es lo que el usuario verá de la aplicación.

1.4.2.1.3 Controlador

Esta es una de las capas más importantes, ya que sirve de intermediario entre la vista y el modelo. A diferencia del modelo, el controlador no accede a los datos ni los manipula, si no que envía las correspondientes peticiones originadas por los usuarios al modelo cuando estos solicitan información y también se encarga de enviar los respectivos comandos a las vistas asociadas a estos cuando se solicita un cambio en la forma que estas se presentan.

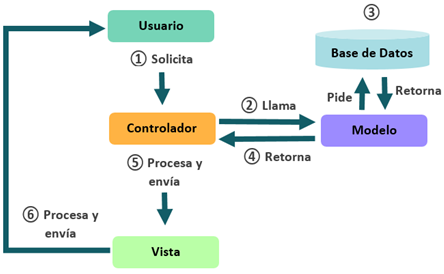


Figura 5 Esquema patrón MVC.

Entre las ventajas de utilizar *frameworks* de desarrollo web que utilicen este patrón de diseño se encuentran las siguientes:

* La rapidez con que se desarrolla la aplicación, dado que esta se divide en módulos diferentes e independientes, los cuales se pueden desarrollar en cualquier momento con independencia de los demás.
* Se integra fácilmente con lenguajes de programación web como PHP, JavaScript, HTML, jQuery y se puede compatibilizar con técnicas de comunicación asíncrona (como AJAX).
* La facilidad para crear múltiples vistas para un modelo. Cada una de ellas será independiente del resto de vistas de la aplicación.
* La organización de código entre las diferentes capas ayuda a una sencilla implementación de nuevas funcionalidades dentro de la aplicación desarrollada.

1.4.2.2 Lenguajes de Programación

Los lenguajes de programación en el ámbito del desarrollo web son aquellos lenguajes que un navegador web podrá entender y, por lo tanto, para los que está preparado para interactuar permitiendo que el usuario visualice desde el navegador un sistema o aplicación que se encuentra alojado en una máquina dentro de la red. Normalmente en un proyecto de desarrollo web se clasifican los lenguajes en dos tipos, lenguajes en el lado del servidor (*back-end*) y lenguajes en el lado del cliente (*front-end*).

Los lenguajes en el lado del servidor (*back-end*) son aquellos lenguajes que se ejecutan en el servidor y se encargan de interactuar con la base de datos, así como también crear las sesiones de usuario o proporcionar los datos necesarios para las vistas de la aplicación. Lenguajes como PHP, Java, Ruby, Python o Node.js son algunos de los lenguajes más utilizados en la actualidad en el lado del servidor.

Por otra parte, los lenguajes en el lado del cliente (*front-end*) son aquellos lenguajes o tecnologías que se ejecutan en el propio navegador web. Los ficheros escritos en estos lenguajes son servidos directamente al navegador para que los interprete. Los navegadores web tienen la capacidad de maquetar y estilizar la vista con la que el usuario está interactuando, con el objetivo de mejorar la experiencia que el usuario tiene con la aplicación. Lenguajes como HTML5, CSS3, JavaScript, TypesScript (librerías como jQuery o *frameworks* como Angular o React) son los principales lenguajes usados en la mayoría de aplicaciones web.

Estos dos tipos de lenguajes no tienen mucho éxito el uno sin el otro, ya que el *back-end* de una aplicación necesitará de los lenguajes del lado del cliente para que el usuario pueda interactuar con la información de la base de datos y el *front-end* necesitará aquella información suministrada por el *back-end* para poder mostrar los resultados de sus interacciones.

A continuación, se listan algunos de los diferentes lenguajes *back-end* con sus correspondientes *frameworks*, los cuales siguen el patrón de diseño MVC y que son muy utilizados en el mercado actualmente:

1.4.2.2.1 PHP: para este lenguaje los *frameworks* más utilizados son Laravel, CakePHP, Symfony o CodeIgniter.

1.4.2.2.2 Java: Struts, Spring o JavaServerFaces son algunos de los *frameworks* más usados.

1.4.2.2.3 Ruby: para Ruby el *framework* más usado es Ruby on Rails.

1.4.2.2.4 Python: para este lenguaje de programación el *framework* más usado en la actualidad es Django.

1.4.2.2.5 Node.js: basado en JavaScript, los *frameworks* más potentes y utilizados son Angular y React.

1.4.2.3 Sistema de gestión de bases de datos

Para finalizar este apartado, una de las partes más importantes en una aplicación web es el sistema de gestión de bases de datos, que almacenará aquella información esencial de la aplicación web de la forma más eficiente y segura posible y que deberá estar disponible en el momento que se necesite.

Normalmente en el desarrollo de aplicaciones web las bases de datos suelen ser relacionales (generalmente SQL ya que es el lenguaje más extendido), sin embargo, en la actualidad muchos desarrolladores optan por una nueva tendencia, las bases de datos no relacionales. La principal diferencia entre estos dos tipos de bases de datos es que las no relacionales no utilizan el modelo relacional que las SQL sí utilizan. A continuación, se exponen algunas de las características más importantes de estos dos tipos:

* Las bases de datos relacionales permiten combinar de una forma eficiente diferentes tablas con el objetivo de extraer información relacionada, mientras que las no relacionales no lo permiten o lo hacen con un peor rendimiento que las relacionales. Ejemplos: PostgreSQL, MySQL, Oracle, etc.
* Las bases de datos no relacionales permiten distribuir grandes cantidades de información de forma más rápida, por lo que este tipo de bases de datos están pensadas para el uso en redes sociales o destinadas al Big Data, es decir, cuando el volumen de datos es de gran tamaño o las estructuras de datos son variables. También cuentan con la capacidad de escalado horizontal, es decir, la capacidad que tienen las bases de datos no relacionales de crecer en la misma máquina, sin tener que almacenarse en máquinas más grandes mientras aumenta su crecimiento. Ejemplos: MongoDB, Cassandra, CouchDB, Redis, etc.

Esta breve introducción al estudio de carácter técnico de este trabajo fin de grado no tiene como objetivo debatir cuál de las tecnologías anteriormente indicadas es mejor. Sin embargo, es importante exponer cuáles son las ventajas e inconvenientes de cada una de ellas. El objetivo final es escoger la tecnología adecuada para la realización de la aplicación web.

En el apartado “3.1.4. Análisis de alternativas” del documento número 3 “Requisitos de usuario y análisis de alternativas” se hará una descripción más detallada de algunos de los lenguajes y herramientas anteriormente mencionados, aportando datos y resultados reales, para posteriormente escoger las tecnologías con las que se desarrollará la aplicación web.

1.5 Organización del documento

En este apartado se mostrará una lista detallada de los diferentes capítulos que componen este trabajo fin de grado, además de una breve descripción de cada uno de ellos.

1.5.1 Documento 1 – Memoria

Se trata del presente capítulo, en él se incluye en primer lugar un resumen del proyecto y una introducción. A continuación, los objetivos y el alcance de este trabajo fin de grado. Después se detallan los estudios y análisis previamente realizados. Por último, se comentarán las conclusiones a las que se ha llegado al finalizar el presente trabajo y se mostrará la organización de los diferentes capítulos que componen este trabajo fin de grado además de la bibliografía consultada.

1.5.2 Documento 2 – Presupuesto y planificación

En este capítulo se detallará la planificación temporal de las diferentes fases del proyecto, desde el inicio hasta su conclusión. A continuación, se describirá el cálculo del coste total del proyecto, así como el coste de los recursos necesarios para el desarrollo de este.

1.5.3 Documento 3 – Requisitos de usuario y análisis de alternativas

En este documento se incluirá la especificación de requisitos de usuario y análisis de alternativas de este trabajo fin de grado. Para acabar el documento se expondrá la alternativa seleccionada.

1.5.4 Documento 4 – Análisis de requisitos del sistema

En este capítulo se recogerán los casos de uso de toda la aplicación, así como también se detallarán los requisitos no funcionales y el modelo de datos.

1.5.5 Documento 5 – Diseño del sistema

En este capítulo se expondrán los diagramas de paquetes, de clases, de iteración, así como también el diseño de la base de datos y el diseño de la interfaz.

1.5.6 Documento 6 – Pruebas

En este documento se detallarán las pruebas que se han hecho durante el desarrollo de este: pruebas unitarias, de integración y sistema, de usabilidad y accesibilidad y de rendimiento.

1.5.7 Documento 7 – Seguridad

En este capítulo se explicarán todas las herramientas utilizadas para blindar el servidor ante cualquier posible fallo de seguridad, ante cualquier ataque masivo o ante cualquier intento de intrusión en la máquina que aloja la aplicación web. Se mostrará una comparativa del antes y el después de las medidas tomadas para blindar el servidor.

1.5.8 Documento 8 – Manuales

En este último documento se detalla una descripción detallada de los pasos que hay que seguir para instalar la aplicación desde cero, además de una recopilación de las librerías más importantes utilizadas en el desarrollo de la aplicación web.

1.6 Conclusiones y ampliaciones

En el presente trabajo fin de grado se han definido claramente los objetivos y el alcance del mismo, por lo que en este apartado se hablará de las conclusiones a las que se ha llegado, además de una pequeña lista de las posibles ampliaciones que pueden llevarse a cabo en esta aplicación.

1.6.1 Conclusiones

Finalizado todo el proceso de implementación y desarrollo asociado a este trabajo fin de grado se puede llegar a la conclusión de que se ha cumplido con los objetivos propuestos inicialmente.

Se han desarrollado una serie de funcionalidades nuevas en la aplicación web y ha sido mejorada la experiencia de usuario para los dos posibles roles definidos (asistentes del congreso y personal de la organización), utilizando tecnologías actualizadas y modernas en el campo de las tecnologías web.

Por otro lado, se han actualizado las tecnologías que se usan en la aplicación web, aumentando la seguridad de la aplicación y su consistencia. Además, se ha conseguido blindar el servidor ante cualquier posible intrusión. La comparativa final entre el antes y el después muestra claramente la mejora de seguridad, eliminando la mayoría de vulnerabilidades que presentaba el sistema.

En definitiva, este trabajo fin de grado ha sido de gran utilidad para la organización ya que se ha conseguido mejorar considerablemente la aplicación y su seguridad con un presupuesto limitado.

1.6.2 Ampliaciones

Con el desarrollo de la aplicación actual finalizado, no cabe duda de que aún existen algunas mejoras que se podrían llevar a cabo. Cualquier aplicación o sistema puede ser mejorado y/o ampliado continuamente, por lo que a continuación se detallan algunas de las posibles ampliaciones y mejoras. Entre ellas se encuentran mejoras de las tecnologías, nuevas funcionalidades, mantenimiento continuo del sistema y mejoras de seguridad.

1.6.2.1 Mejora de la infraestructura

Esta mejora incluye una mejora del *hardware* del sistema. Hoy en día los usuarios de internet demandan una gran velocidad en las operaciones y en la navegación de las páginas y aplicaciones web. Es por lo que una de las posibles mejoras sería el remplazo del disco duro, que actualmente es un disco de grabación magnética (HDD) por una unidad de estado sólido (SSD). Los discos SSD tienen una gran mejora respecto a los discos HDD, por su fiabilidad, su velocidad y su consumo. Otra mejora de la infraestructura sería la interfaz de red, la memoria RAM y/o el procesador del sistema.

En la actualidad existen dispositivos mucho más potentes que mejorarían considerablemente la velocidad total del sistema. En detrimento de estas mejoras está el precio a pagar por ellas. La mejora de estos dispositivos requiere de una gran inversión.

1.6.2.2 Mantenimiento continuo de actualizaciones de *software*

El software implementado en esta aplicación proviene de librerías, lenguajes y sistemas operativos de código abierto. Esta comunidad *open source* es muy amplia por lo que continuamente se renueva el software utilizado y todas sus versiones. Lo ideal para un sistema de estas características es la actualización continua de todas las versiones estables que van saliendo. De esta forma se mejora el rendimiento y la seguridad, ya que estas actualizaciones mejoran ambos aspectos.

1.6.2.3 Diseño responsivo de la interfaz

La interfaz actual soporta una visualización para una resolución de una pantalla grande, de un ordenador. En la actualidad muchos usuarios de la red usan otro tipo de dispositivos para navegar, ya sean *smartphones* o *tablets*.El diseño de estilos para estos dispositivos no es acorde con los estándares actuales de responsividad. Una de las mejoras sería el diseño de hojas de estilos que adecuen la visualización de la interfaz para todos los dispositivos, mejorando aún más la usabilidad.

1.6.2.4 Internacionalización

La aplicación está desarrollada en inglés ya que hoy en día está considerado el idioma universal. Sin embargo, muchos de los usuarios de la aplicación tienen otra lengua nativa.

Esta mejora se basa en la inclusión de un componente dentro de la interfaz pública y de administración que permita cambiar al idioma nativo del usuario. Además, se podría identificar el lenguaje del navegador desde el que se accede a la aplicación para iniciar la misma en el idioma que tenga el navegador por defecto.

1.7 Bibliografía

A continuación, se incluye la bibliografía consultada en la realización de este trabajo fin de grado. En primer lugar, se hará referencia a los libros y/o artículos. En el siguiente punto, se detallarán los sitios web consultados.

1.7.1 Libros y artículos

Durante el desarrollo del proyecto se han consultado los siguientes libros, ya sea en versión papel o en versión digital:

* **Guía de PHP**: Diego Lázaro – Leanpub, 2015
* **THE SCHOCK DOCTRINE**: Naomi Kelin – ST. MARTIN S PRESS, 2008
* **Persistence in PHP with Doctrine ORM**: Kévin Dunglas – PACKT open source, 2013
* **JavaScript and JQuery: Interactive Front-End Web Development**: Jon Duckett – Wiley, 2014
* **PHP y MySQL**: Michele E. Davis, Jon A. Phillips – Anaya Multimedia , 2008

1.7.2 Referencias en internet

En este apartado se detallarán los sitios webs consultados para el desarrollo del proyecto, así como para la documentación y la instalación de las herramientas utilizadas:

* **Programa virtualización de máquinas VirtualBox**: <https://www.virtualbox.org/>
* **Sistema operativo CentOS**:<https://www.centos.org/>
* **Servidor Apache**:<https://httpd.apache.org/>
* **Lenguaje PHP**: <http://php.net/>
* **Doctrine ORM**: <http://www.doctrine-project.org/>
* **Tutoriales web W3Schools**: <https://www.w3schools.com/>
* **Lenguaje de bases de datos MySQL**: <https://www.mysql.com/>
* **Librería Bootstrap**: <http://getbootstrap.com>
* **Librería jQuery**: <https://jquery.com/>
* **Lenguaje JavaScript**: <https://www.javascript.com/>
* **Librería Datatables**: <https://datatables.net/>
* **Documentación e incidencias de librerías en GitHub**: <https://github.com/>
* **Comunidad de desarrolladores**: <https://es.stackoverflow.com/>
* **Información, definiciones y significados**: <https://www.wikipedia.org/>
* **Artículos y blogging sobre infraestructura web**: <https://geekflare.com/>
* **Congreso Sistedes**: <https://fg.ull.es/sistedes2017/>
* **Congreso ICSOFT**: <http://www.icsoft.org/>
* **Congreso ENASE**: <http://www.enase.org/>
* **Sistema de gestión de congresos (Aconf)**: <https://www.aconf.org/>
* **Sistema de gestión de congresos (EasyChair)**: <http://easychair.org/licenses.cgi>
* **Sistema de gestión de congresos (Primoris)**: <https://www.insticc.org/Primoris/Default.aspx/>
* **Comunidad *open source* de seguridad del *software***: <https://www.owasp.org/>
* **Plataforma de *cloud computing* con comunidad de soporte**: <https://www.digitalocean.com>
* **Artículo de usabilidad web**: <https://www.crazyegg.com/blog/principles-website-usability/>
* **Artículo de métodos de usabilidad**: <http://www.usabilityfirst.com/usability-methods/>
* **Herramienta de medición de usabilidad**: <http://www.tawdis.net>

# DOCUMENTO 2: PLANIFICACIÓN Y PRESUPUESTO

**D. VIGIL RODRÍGUEZ, Guillermo**

**TUTOR: D. RANILLA PASTOR, José**

**FECHA: Mayo 2018**

Índice de contenidos

[DOCUMENTO 2: PLANIFICACIÓN Y PRESUPUESTO 38](#_Toc505427004)

[Índice de contenidos 40](#_Toc505427005)

[2.1 Introducción 41](#_Toc505427006)

[2.2 Planificación temporal 41](#_Toc505427007)

[2.2.1 Fase de análisis 42](#_Toc505427008)

[2.2.2 Fase de diseño 42](#_Toc505427009)

[2.2.3 Fase de implementación 43](#_Toc505427010)

[2.2.4 Fase de pruebas 43](#_Toc505427011)

2.2.5 Fase de seguridad 44

[2.3 Presupuesto 44](#_Toc505427012)

[2.3.1 Coste del material software/hardware 44](#_Toc505427013)

[2.3.2 Coste del personal 46](#_Toc505427014)

[2.3.3 Coste total 47](#_Toc505427015)

## 2.1 Introducción

El presente documento se dividirá en dos partes, siendo la primera la planificación temporal del desarrollo de este trabajo fin de grado y la segunda parte una estimación detallada del coste total de la realización de este.

2.2 Planificación temporal

El inicio de este proyecto ha comenzado el 16 de noviembre de 2016, con fecha prevista de finalización para el 15 de mayo de 2018, sumando una duración total de 18 meses aproximadamente. El desarrollo se ha producido con una hora al día de media, sin incluir festivos ni fines de semana o periodos vacacionales.

El desarrollo principal se ha dividido en cinco fases principales: fase de análisis, fase de diseño, fase de implementación, fase de pruebas y fase de seguridad. Además, se ha planificado también la parte correspondiente a la documentación. A continuación, se detallan distintas imágenes pertenecientes al diagrama Gantt creado durante el desarrollo del proyecto, con el objetivo de establecer una planificación detallada y concisa del trabajo.

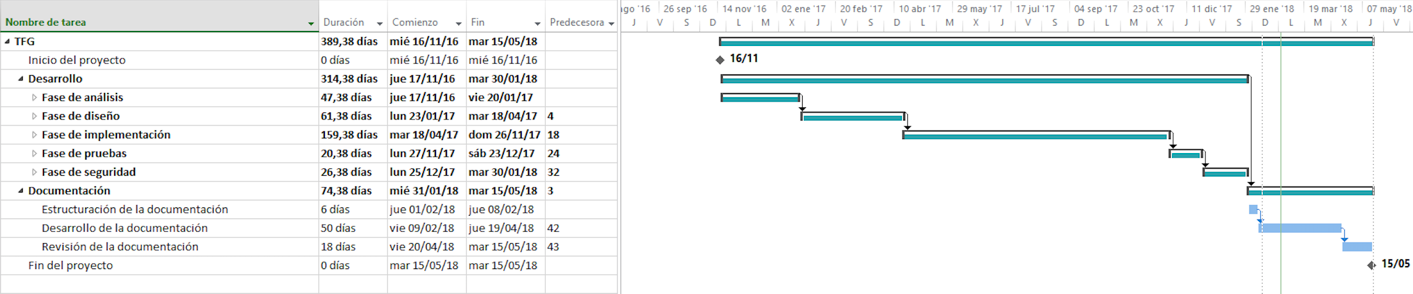


Figura 6 Diagrama Gantt – Visión global

2.2.1 Fase de análisis

Se trata de la fase de análisis del proyecto. En primer lugar, se detalla la planificación de los estudios y análisis previos. Más tarde los requisitos de usuario y el análisis de alternativas. Por último, se planifica el análisis de requisitos del sistema.

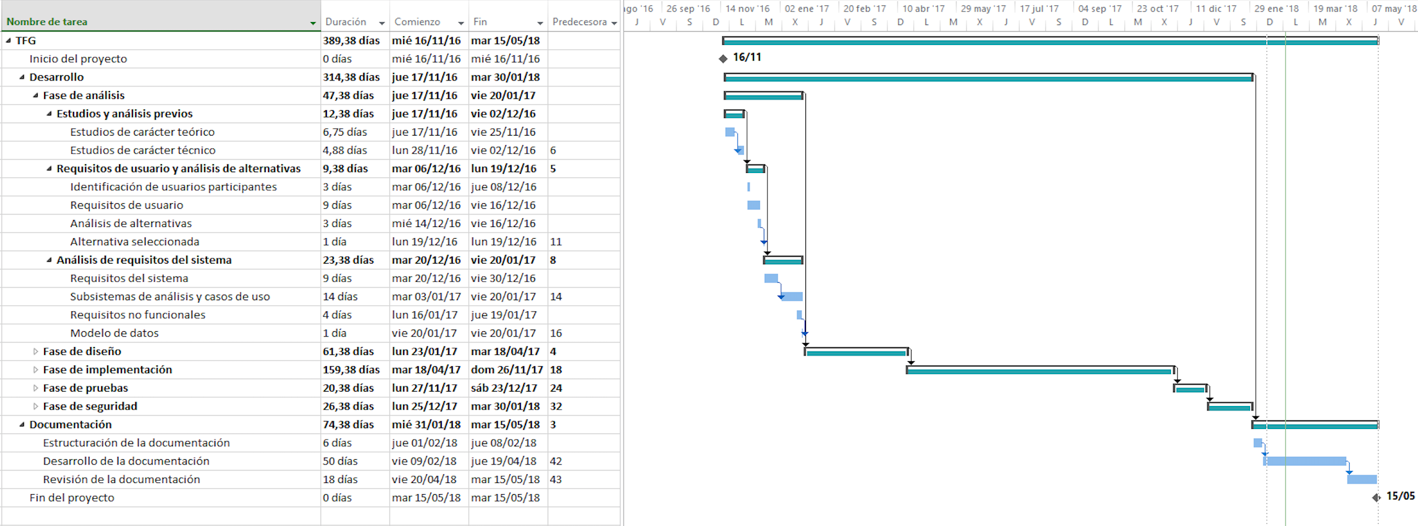


Figura 7 Diagrama Gantt - Fase de análisis

2.2.2 Fase de diseño

En esta fase se planifican los diseños específicos de ingeniería del software, diseño de la base de datos y de la interfaz de usuario.

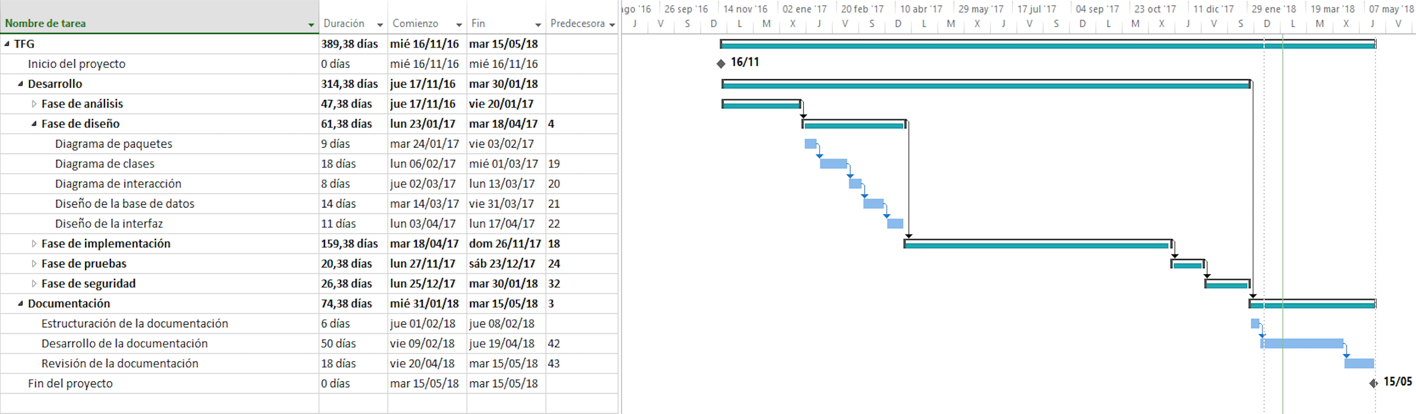


Figura 8 Diagrama Gantt - Fase de diseño

2.2.3 Fase de implementación

Es la fase de creación de la aplicación y de su estructura, y de programación de todas las funcionalidades necesarias en la mejora de la web.

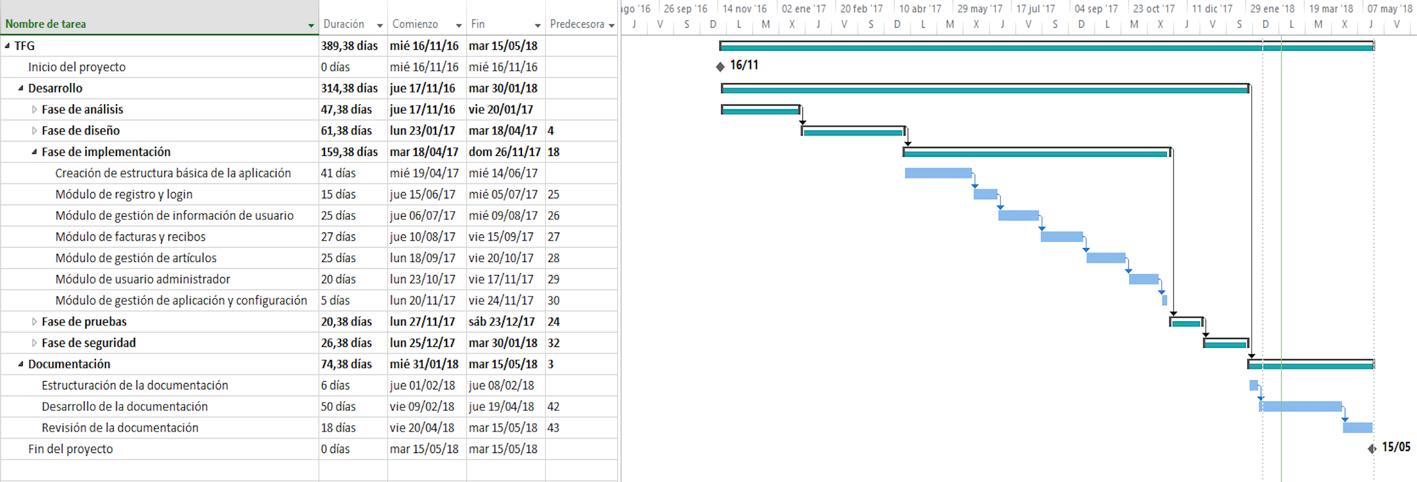


Figura 9 Diagrama Gantt - Fase de implementación

2.2.4 Fase de pruebas

Se trata de la fase de pruebas de la aplicación. Pruebas funcionales y no funcionales.

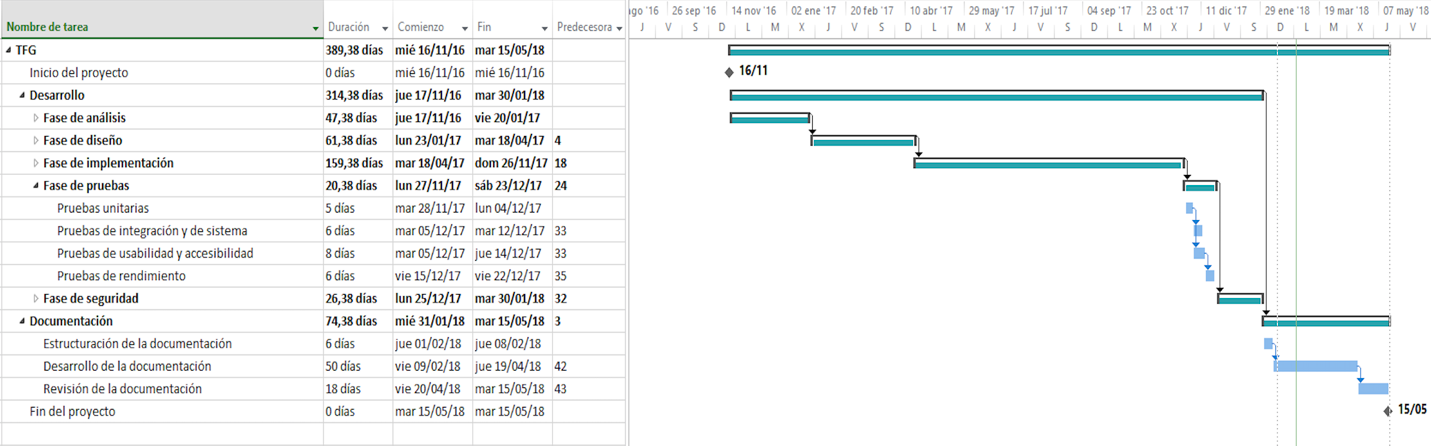


Figura 10 Diagrama Gantt - Fase de pruebas

**2.2.5 Fase de seguridad**

En esta fase se planifican todas las medidas de seguridad empleadas en la mejora de la aplicación web. Se incluyen mejoras de seguridad del servidor, de la aplicación y del software utilizado.

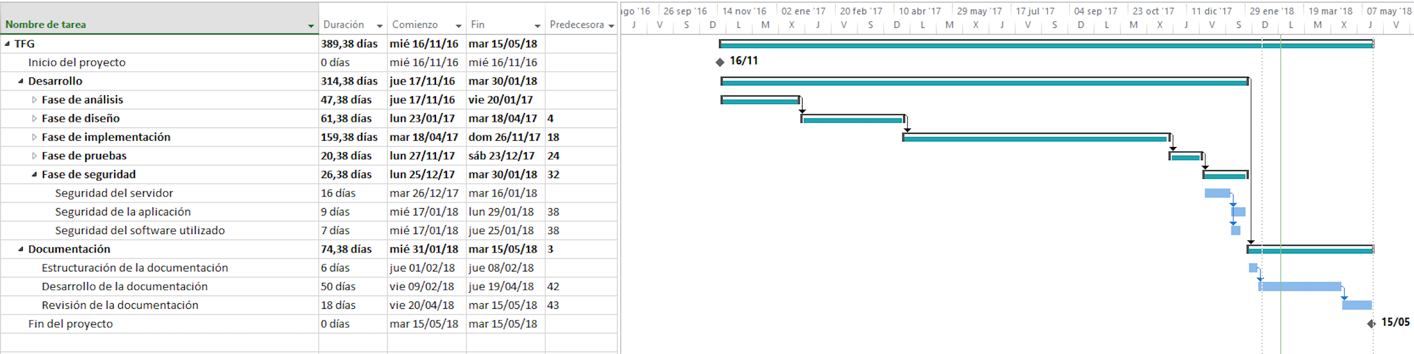
****

Figura 11 Diagrama Gantt - Fase de seguridad

2.3 Presupuesto

El presupuesto del presente trabajo fin de grado se va dividir en dos partes. La primera de ellas correspondiente al coste de los elementos hardware y software del sistema. La segunda parte corresponde al coste del personal involucrado en el desarrollo del mismo. En los siguientes apartados se presentará una estimación detallada del coste de las dos partes anteriormente citadas y del coste total que conlleva el desarrollo de este trabajo fin de grado.

2.3.1 Coste del material software/hardware

El coste del material software y hardware utilizado en el desarrollo de la aplicación web se realizará en este primer apartado.

Cabe destacar que para el desarrollo de la aplicación web se han utilizado únicamente tecnologías de software *open source* por lo que la parte software no tendrá coste alguno, así como también el sistema operativo del equipo y la versión de este con la que se ha trabajado, Linux Ubuntu 14.04 LTS. Por otro lado, el editor de textos utilizado para trabajar con nuestro repositorio Ruby on Rails ha sido Sublime Text 2, un editor de código gratuito que no supone ningún coste adicional al proyecto.

Un aspecto importante a destacar es que el equipo con el que desarrollaremos la aplicación web y con el que también haremos la documentación solo lo utilizaremos el tiempo que tardemos en el desarrollo y en la documentación del proyecto. Por tanto, para calcular el coste que nos supone utilizar el equipo durante el tiempo de desarrollo y documentación del proyecto fin de grado debemos calcular su coste de amortización. Según la Agencia Tributaría y como se especifica la tabla de coeficientes de amortización lineal, aquellos equipos electrónicos e informáticos catalogados como equipos para procesos de información tienen un gasto de amortización del 25% anual (con un máximo de 8 años). Por lo tanto si nuestro equipo (**Lenovo IdeaPad G50-70) tiene un coste de 629€ (IVA incluido), el gasto de amortización asociado a este proyecto será de:**

629 euros x 25% = 157,25 euros anuales, como se ha utilizado sólo 3,5 meses del año, multiplicando 157,25 x 0,29 = **45,60 euros** deducibles degastos de amortización.

Respecto a la conexión internet, se ha incluido en el presupuesto una tarifa de red necesaria para el desarrollo de la aplicación (necesaria para descargar de librerías, actualizaciones, consulta de documentación, etc).

Cabe destacar que si se decide subir la aplicación web a un hosting (servidor en la nube) para que esta esté disponible en internet se debería contratar un servicio de hosting adecuado, lo cual supondría un coste adicional de 10€ aproximadamente mensual.   
<https://www.digitalocean.com/pricing/#droplet>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***#*** | **Descripción** | **Unidad** | **Cantidad** | **Precio/Ud (€)** | **Total** |
| *1* | Lenovo IdeaPad G50-70 Intel i7 - 8GB | ud | 1 | 45,60 | 45,60 € |
| *2* | Tarifa de red | Mes | 3,5 | 25 | 87,5 € |
| *3* | Sistema operativo | Ud | 1 | 0 | 0 € |
| *4* | Framework (RoR) | Ud | 1 | 0 | 0 € |
| *5* | Entorno de desarrollo (Edito de texto) | Ud | 1 | 0 | 0 € |
| *6* |  |  |  | ***SUBTOTAL*** | **133,1 €** |

2.3.2 Coste del personal

En este apartado se calculará el coste del personal involucrado en el desarrollo del proyecto, haciendo distinción entre las distintas tareas que realiza cada perfil. Por un lado tenemos el perfil de analista informático el cual se encargará principalmente de desarrollar las fases de análisis y diseño del proyecto, así como incluirlas en la documentación. Por otro lado tenemos el perfil de desarrollador informático, que se encargará de las fases de implementación y análisis, así como también de incluir estas en la documentación y los manuales de usuario e instalación. Para realizar los correspondientes cálculos se ha estimado una jornada laboral de 4 horas diarias durante toda la duración del proyecto (excluyendo días laborales y festivos), donde en esas 4 horas un 70% se ha destinado a realizar las tareas correspondientes de cada fase y un 30% a la documentación. A un precio de 15€/hora la jornada de trabajo saldría a 60€/día.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Descripción** | **Cantidad (días)** | **Precio/Ud** | **Total** |
| 1 | **Desarrollo de la aplicación** |  |  |  |
| 1.1 | Analista informático | - | - | - |
| 1.1.1 | Fase de análisis | 23 | 60 € | 1.380,00 € |
| 1.1.2 | Fase de diseño | 20 | 60 € | 1.200,00 € |
|  |  |  |  |  |
| 1.2 | Programador informático | - | - | - |
| 1.2.1 | Fase de implementación | 27 | 60 € | 1.620,00 € |
| 1.2.1 | Fase de pruebas | 7 | 60 € | 420,00 € |
|  |  |  |  |  |
| 2 |  |  | **SUBTOTAL** | **4.620,00 €** |

2.3.3 Coste total

En este apartado se calculará el coste total del proyecto, obtenido a partir de la suma del coste de material software/hardware más el coste del personal involucrado, expresado en euros.

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | **Descripción** |  | **Total** |
| *1* | Coste del material (software/hardware) |  | 133,1 € |
| *2* | Coste del personal |  | 4.620,00 € |
| *3* |  | ***SUBTOTAL*** | **4.753,10 €** |
| ***#*** | **Descripción** |  | **Total** |
| *4* |  | *I.V.A (21%)* | 998,151 € |
| ***5*** |  | **TOTAL** | **5.751,25 €** |

# DOCUMENTO 3: REQUISITOS DE USUARIO Y ANÁLISIS DE ALTERNATIVAS

**D. MARTÍNEZ SUÁREZ, Wenceslao**

**TUTOR: Dña. SUAREZ CABAL, María José**

**FECHA: Julio 2017**

Índice de contenidos

[**DOCUMENTO 3: REQUISITOS DE USUARIO Y ANÁLISIS DE ALTERNATIVAS** 40](#_Toc486815152)

[3.1 INTRODUCCIÓN 42](#_Toc486815153)

[3.2 IDENTIFICACIÓN DE USUARIOS PARTICIPANTES EN EL SISTEMA 42](#_Toc486815154)

[3.2.1 ROL DE ESTUDIANTE 42](#_Toc486815155)

[3.2.2 ROL DE EMPRESA 42](#_Toc486815156)

[3.3 REQUISITOS DE USUARIO 43](#_Toc486815157)

[3.3.1 REQUISITOS DE USUARIO COMUNES 43](#_Toc486815158)

[3.3.2 REQUISITOS DE USUARIO – ROL DE ESTUDIANTE 43](#_Toc486815159)

[3.3.2 REQUISITOS DE USUARIO – ROL DE EMPRESA 44](#_Toc486815160)

[3.4 ANÁLISIS DE ALTERNATIVAS 45](#_Toc486815161)

[3.4.1 TECNOLOGÍAS BACK-END 45](#_Toc486815162)

[3.4.2 TECNOLOGÍAS FRONT-END 51](#_Toc486815163)

[3.4.3 BASE DE DATOS 52](#_Toc486815170)

[3.5. DESCRIPCIÓN DE LA ALTERNATIVA SELECCIONADA 54](#_Toc486815171)

[3.5.1 TECNOLOGÍAS BACK-END 54](#_Toc486815172)

[3.5.2 TECNOLOGÍAS FRONT-END 56](#_Toc486815173)

[3.5.3 BASE DE DATOS Y GESTOR DE BASE DE DATOS 57](#_Toc486815174)

[3.5.4 SISTEMA OPERATIVO 58](#_Toc486815175)

3.1 Introducción

En este apartado se presenta la parte de análisis del proyecto, la cual es unas de las partes más importantes de cualquier proyecto de desarrollo software. Definiremos los diferentes roles que puede tener un usuario que utilice la aplicación, así como los requisitos de usuario, un estudio más detallado de las tecnologías y herramientas a utilizar y finalmente cuales de estas hemos seleccionado para el desarrollo de esta.

3.2 Identificación de usuarios participantes en el sistema

Cabe destacar que nuestra aplicación contará con dos roles de usuario:

3.2.1 Rol de estudiante

La aplicación web permitirá crear un perfil de estudiante detallando su perfil profesional, así como los estudios u otras competencias profesionales que curse en la actualidad o aquellos que ya posea el usuario. El estudiante también podrá inscribirse a aquellas ofertas de prácticas que sean de su interés, participando en los procesos de selección respectivos.

3.2.2 Rol de empresa

Respecto al rol de empresa, esta aplicación web permitirá crear un perfil de empresa detallando sus datos así como su sector de trabajo. Las empresas podrán crear ofertas de prácticas laborales para que los estudiantes puedan inscribirse en ellas si lo desean, iniciando respectivamente los procesos de selección pertinentes.

Lógicamente habrá diferencias entre estos dos roles, ya que a pesar de que algunas funcionalidades de la aplicación serán las mismas para ambos, habrá otras que sean específicas únicamente para un usuario con el rol de estudiante y otras para un usuario con el rol de empresa. Estas funcionalidades las veremos detalladas a continuación en el apartado requisitos de usuario.

3.3 Requisitos de usuario

Como hemos mencionado anteriormente los requisitos de usuario son de gran importancia para el buen desarrollo de cualquier producto software. Este tipo de requisitos son declaraciones en lenguaje natural de los servicios que un usuario espera que el sistema provea y aquellas restricciones bajo las que debe funcionar.

A continuación se detallarán los requisitos de usuario divididos en 3 apartados; requisitos de usuarios comunes a los dos roles de usuario existentes, requisitos de usuarios propios del rol de estudiante y por último requisitos de usuario propios del rol de empresa.

3.3.1 Requisitos de usuario comunes

* Registro en la aplicación mediante un formulario web.
* Recuperación de contraseña de usuario.
* Modificación de credenciales de acceso y datos básicos cuando el usuario desee (email y contraseña).
* Eliminación de cuenta de la aplicación web cuando.
* Envío de mensajes con otros usuarios registrados de la aplicación. En el caso de los estudiantes estos podrán mandar mensajes a las empresas, y en el caso de las empresas estas solo podrán mandar mensajes a los estudiantes.

3.3.2 Requisitos de usuario – rol de estudiante

* Configurar y editar su perfil profesional detallando sus datos académicos y otras competencias (estudios, experiencias laborales, cualidades y aficiones e intereses si lo desea).
* Realizar búsquedas filtradas de aquellas ofertas que hayan sido creadas por empresas. Se podrá filtrar la búsqueda por comunidad autónoma, perfil de estudios requerido, remuneración y por el estado de la oferta (oferta abierta o cerrada).
* Inscripción en aquellas ofertas de prácticas que estén abiertas (entrando en el proceso de selección de la misma), así como también des inscribirse de aquellas en las que ya estuviera inscrito con anterioridad.
* Visualización del estado del proceso de selección de la/s oferta/s en las que esté inscrito. Una vez finalizado el plazo de inscripción de la oferta (la empresa establece la duración), el estudiante entra en la primera fase del proceso de selección automáticamente (15 días de duración), y si durante esta primera fase la empresa selecciona al estudiante para la siguiente fase (fase final), el sistema notificará al estudiante con un mensaje y pasará a la fase final. En el caso de que la empresa no seleccionase al estudiante durante esta primera fase, el estudiante quedará fuera del proceso de selección, siendo notificado por el sistema por medio de un mensaje.

Una vez avanzado a la fase final (duración de 15 días), la empresa debe seleccionar al estudiante que disfrutará de la oferta de prácticas. Cuando la empresa seleccione al estudiante a escoger, el sistema se lo notificará con un mensaje, mientras que los demás estudiantes inscritos en esa oferta serán avisados por el sistema mediante un mensaje de que la oferta de prácticas ha sido cerrada y no han sido seleccionados.

* Realizar búsquedas filtradas de aquellas empresas que también estén registradas en la aplicación, así como ver la información relativa a estas (ofertas publicadas, perfil de la empresa). Se podrá filtrar por comunidad autónoma y por el sector de trabajo de la empresa.

3.3.2 Requisitos de usuario – rol de empresa

* Configuración de perfil con una descripción en base a las actividades profesionales a las que se dedique.
* Realizar una búsqueda filtrada de los estudiantes que estén registrados en la aplicación, así como ver su perfil e información, además de la opción de ponerse en contacto con ellos. Se podrá filtrar la búsqueda por comunidad autónoma y por perfil de estudios del estudiante.
* Creación de ofertas de prácticas de trabajo, indicando la necesaria información al respecto (actividades a realizar y duración de las prácticas, horario, ubicación, remuneración, plazo de inscripción, etc.)
* Gestión del proceso de selección por cada oferta creada, donde la empresa, una vez cerrado el plazo de inscripción de la oferta, podrá en primer lugar (primera fase -15 días de duración) elegir aquellos estudiantes que considere oportunos para pasar a la siguiente fase (fase final – 15 días de duración) y posteriormente seleccionar al estudiante finalmente elegido para disfrutar de las prácticas laborales.

3.4 Análisis de alternativas

A continuación se hará un estudio comparativo de las diferentes herramientas nombradas anteriormente. En primer lugar se mostrará una breve descripción de los lenguajes de programación y sus correspondientes frameworks que más se utilizan en el desarrollo web y que formarán la parte back-end de nuestra aplicación:

3.4.1 Tecnologías back-end

**3.4.1.1 PHP:** es el lenguaje más usado para el desarrollo web en la actualidad. Es un lenguaje de código abierto el cual lleva desarrollándose más de 20 años, basado en los lenguajes #C y PERL. PHP cuenta con una gran comunidad de desarrolladores y usuarios a sus espaldas, así como numerosas aplicaciones web de éxito en las que se ha utilizado este lenguaje (Yahoo, Wikipedia, Wordpress o Flickr).

Una de las principales ventajas de PHP es que este lenguaje es uno de los más demandados por las empresas y es compatible con la mayoría de sistemas operativos y plataformas, y cuenta con varios frameworks de desarrollo (Laravel o Symphony) que hacen su uso mucho más ameno.



Figura 11 Logo PHP

**3.4.1.2 Ruby:** es un lenguaje de programación orientado a objetos lanzado al mercado en 1995. Generalmente cuando se habla del lenguaje Ruby en el ámbito de la programación web se suele hablar de Ruby on Rails, framework de desarrollo que se utiliza para construir aplicaciones web bajo el lenguaje Ruby. Este framework es de código libre y en los últimos años ha ganado una notable importancia en el mercado de las aplicaciones web, contando con una gran comunidad de desarrolladores a sus espaldas. Twitter, Github, LinkedIn o SlideShare son algunas de las aplicaciones desarrolladas en Ruby on Rails. Algunas de las ventajas de utilizar este framework para el desarrollo web es la rapidez y facilidad que aporta el entorno de trabajo para construir la aplicación, así como también lo sencillo y rápido que es incluir paquetes de librerías en el proyecto.

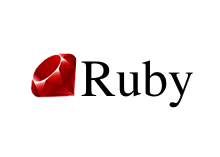


Figura 12 Logo Ruby

**3.4.1.3 Python:** es uno de los lenguajes de alto nivel más populares entre los programadores, el cual fue creado a finales de los 80. Python se caracteriza por ser un lenguaje potente, flexible y fácil de desarrollar.

Aunque no es el lenguaje más utilizado en el ámbito del desarrollo web, Python cuenta un potente framework llamado Django y dispone de una amplia documentación. Aplicaciones web como Instagram, Pinterest o The New York Times son solo algunas de las aplicaciones que se han desarrollado con este lenguaje.

****

Figura 13 Logo Python

**3.4.1.4 ASP.NET:** ASP.NET, el cual es la continuación de ASP, es una plataforma web desarrollada por Microsoft la cual proporciona todos aquellos recursos y servicios necesarios para desarrollar y compilar aplicaciones web. El lenguaje de programación que utiliza este entorno por lo general suele ser #C o Visual Basic. Una importante desventaja es que es un software de código propietario por lo que hay que abonar el coste de aquellas licencias si se desea utilizar este entorno.

**3.4.1.5 Java EE:** Java Enterprise Edition es una plataforma de programación o conjunto de estándares destinada a desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java, la cual hace uso de componentes modulares y se ejecuta sobre un servidor de aplicaciones. El objetivo que se pretender conseguir usando este entorno es el poder desarrollar aplicaciones que sean portables entre plataformas y escalables. Esta plataforma es de código abierto lo que no supone ningún coste adicional a la hora de utilizarla.

Algunas de las tecnologías o frameworks más usadas y destinadas al desarrollo web que se integran con los estándares de la plataforma Java EE son: JavaServer Faces framework, Spring framework o Struts framework.



Figura 14 Logo Java EE

A continuación se hará una comparativa real entre dos de los lenguajes y frameworks más usados expuestos anteriormente. En primer lugar se han incluido los resultados de un benchmark que mide el rendimiento proporcionado por lenguajes como PHP y frameworks como Ruby on Rails o Laravel respecto a la capacidad de procesamiento de peticiones por segundo.

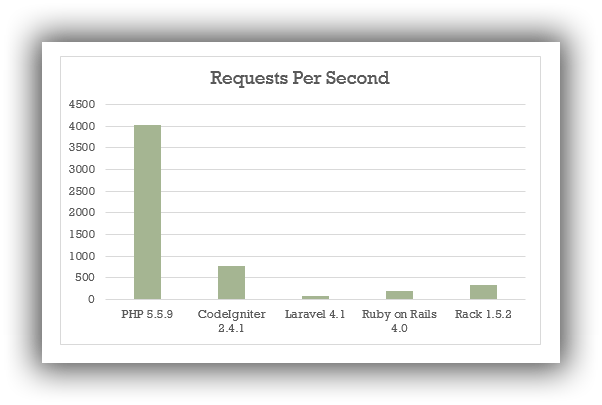


Figura 15 Gráfico benchmark de rendimiento.

Podemos observar que el lenguaje PHP es el que más peticiones por segundo es capaz de procesar, y es normal, ya que este lenguaje ha sido diseñado específicamente para el desarrollo web y proporciona un buen rendimiento en este aspecto, pero cuando este se integra en frameworks como CodeIgniter o Laravel, el rendimiento baja drásticamente. Por otro lado tenemos el framework anteriormente mencionado Ruby on Rails, que según podemos observar proporciona un rendimiento mejor que Laravel pero menor que CodeIgniter.

No menos importante, lo pequeña o grande que sea la comunidad de desarrolladores detrás de cada lenguaje o framework es de gran importancia, ya que cuanta más documentación, información o problemas resueltos haya sobre ese lenguaje o framework, más ayuda encontraremos y más fácil será desarrollar la aplicación web. En la siguiente imagen se puede observar una comparativa sobre el número de preguntas hechas en internet a cerca del lenguaje PHP y sus diferentes frameworks, así como también para Ruby on Rails.

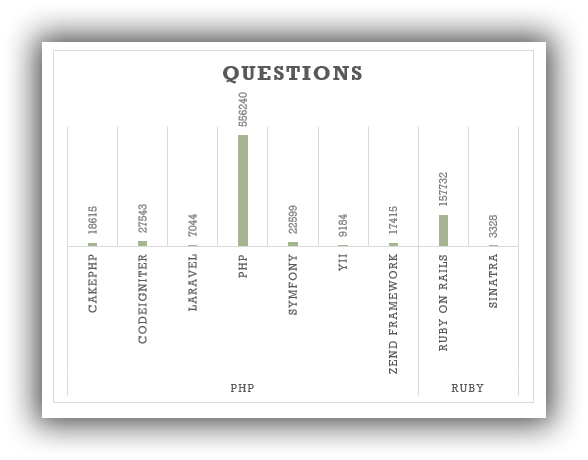


Figura 16 Gráfico de preguntas realizadas en Internet.

Podemos observar que PHP tiene más preguntas relacionadas que cualquiera de los frameworks en los que se usa y que Ruby on Rails también. No obstante, si hacemos una comparativa del número de preguntas que se hacen en Internet respecto a los diferentes frameworks, Ruby On Rails es el que marca la diferencia siendo el framework sobre el que más preguntas se hace, creando así una amplia variedad de información y ayuda acerca de este.

3.4.2 Tecnologías front-end

Una vez analizados los distintos lenguajes y plataformas que formarán nuestro back-end, veremos las distintas tecnologías del lado del cliente (front-end) que más se utilizan hoy en día. El front-end de una aplicación es de suma importancia, ya que las tecnologías que lo componen se encargan de mostrar la parte visual de nuestra aplicación al usuario. Estas son algunos de los lenguajes más utilizados hoy en día:

3.4.2.1 HTML (HyperText Markup Language): mundialmente conocido, el lenguaje HTML es un estándar que define una estructura básica y un código (HTML) por el que se define el contenido de una página web. Con este lenguaje el principal objetivo que se busca es el dar formato a los diversos contenidos de una página web y presentarlos ante un navegador.

3.4.2.2 XML (eXtensive Markup Language): este lenguaje fue desarrollado para organizar contenidos y describir la información de una página web. Aunque su sus características puedan ser similares a las del HTML, el XML suele funcionar como un complemento del HTML por lo general.  
http://www.maestrosdelweb.com/xmlvshtml/

3.4.2.3 Javascript: este lenguaje, principalmente influenciado por Java, es un lenguaje que ejecuta el código en el lado del cliente (front-end) en cuanto a aplicaciones web se refiere. Se ha convertido en uno de los lenguajes más populares de internet y con él se pretende mejorar la experiencia del usuario, mejorando la interfaz de usuario y haciendo las páginas webs más dinámicas.

3.4.2.4 AngularJS: este framework, el cual utiliza el lenguaje JavaScript, es uno de los frameworks que más fama ha cogido en el último año para el desarrollo front-end de aplicaciones web dinámicas. AngularJS, que actualmente está en la versión 2.0, permite usar HTML como su lenguaje de plantilla y permite extender la sintaxis de HTML para configurar los componentes de su aplicación de forma clara y dinámica.

3.4.2.5 CSS (Cascading Style Sheets): este lenguaje se utiliza para establecer el diseño visual de las páginas web (escritas por lo general en HTML o XML). Se suele combinar con HTML y Javascript, con el objetivo de crear páginas con una interfaz de usuarios atractiva.

3.4.2.6 Bootstrap: este framework de código abierto lanzado en el año 2011 es una de las herramientas de diseño web que más se ha incorporado en proyectos de desarrollo web en estos últimos años. Bootstrap proporciona un conjunto de hojas de estilo que contienen definiciones básicas de estilo para todos los componentes de HTML y CSS (tipografía, formularios, botones, etc.)



Figura 17 Logo Bootstrap

Este framework facilita y mejora la tarea de maquetación de una aplicación web, proporcionando un diseño “responsive” el cual se adapta a todo tipo de dispositivos.

3.4.3 Base de datos

Por último y como mencionamos en el apartado 1.4 de “Estudios y análisis previos” perteneciente al documento 1 “Memoria”, uno de los elementos clave en una aplicación web es el sistema de gestión de bases de datos.

MySQL, PostgreSQL, MariaDB o Microsoft SQL Server son algunos de los lenguajes de bases de datos relacionales más usados, mientras que por otro lado MongoDB, CouchDB o Redis son uno de los lenguajes de bases de datos no relacional líder en la actualidad.

A continuación se muestra un benchmark de rendimiento, donde podremos ver una comparativa de los lenguajes relacionales anteriormente mencionados, PostgreSQL, MariaDB y MySQL. Este benchmark mide el tiempo que tarda (en milisegundos) cada lenguaje en hacer diferentes operaciones de lectura y escritura.



Figura 18 Gráfico comparativo de rendimiento.

Podemos observar que salvo en la operación de escritura donde MySQL proporciona un mejor rendimiento, PostgreSQL es el lenguaje que mejor rendimiento proporciona a la hora de hacer consultas simples y complejas en base de datos, seguidas en segundo lugar por MariaDB.

Otra comparación necesaria es entre MySQL y MongoDB, ya que estos últimos años el lenguaje no relacional MongoDB ha ganado mucho protagonismo, debido a su escalabilidad y rendimiento frente a grandes volúmenes de datos y estructuras variables. En la siguiente tabla podemos ver dos tipos de operaciones (insert y select) hechas por estos dos lenguajes, para 10,100 y 1000 columnas. Los resultados están expresados en milisegundos.

Tabla 1 Comparativa entre MongoDB y MySQL

|  |  |  |
| --- | --- | --- |
| **10 COLUMNAS** | **MYSQL** | **MongoDB** |
| insert | 1702 ms | 11 ms |
| select | 11 ms | 12 ms |
| **100 COLUMNAS** |  |  |
| insert | 8171 ms | 10 ms |
| select | 167 ms | 60 ms |
| **1000 COLUMNAS** |  |  |
| insert | 94813 ms | 13 ms |
| select | 1013 ms | 677 ms |

Tras observar los resultados podemos ver que MongoDB es notablemente mejor que MySQL respecto a operaciones tipo “select” e "insert” sobre la base de datos.

3.5. DESCRIPCIÓN DE LA ALTERNATIVA SELECCIONADA

Una vez realizado el estudio de alternativas escogeremos entre aquellas tecnologías o herramientas que más se ajusten para el desarrollo de este proyecto. A continuación se detallarán aquellas herramientas que hemos seleccionado para el back-end y front-end, así como la base de datos y el sistema operativo.

3.5.1 Tecnologías back-end

Tras analizar las diferentes alternativas para el back-end de nuestro proyecto con los respectivos frameworks existentes, utilizaré el lenguaje Ruby con Ruby on Rails como framework principal para el desarrollo de la aplicación. A pesar de que PHP o Java son lenguajes realmente buenos así como sus diferentes frameworks, Ruby on Rails se caracteriza por la facilidad y rapidez que proporciona para desarrollar aplicaciones web. A continuación se expondrán algunas de las ventajas de usar Ruby On Rails:

* Este framework sigue el patrón MVC visto anteriormente y proporciona una estructura estándar de aplicación web al inicio de cualquier proyecto, haciendo que el programador no tenga que partir de cero.
* Existen cientos de paquetes de librerías (en ruby llamadas gemas) que se pueden incorporar fácilmente a nuestro proyecto, las cuales reciben un soporte continuo y nos permiten cubrir necesidades como la autenticación, seguridad o incluir cualquier otra herramienta como librerías de JavaScript a nuestro proyecto.
* Una de los aspectos más interesantes de Ruby on Rails respecto a la programación es la limpieza y sencillez con la que el código se escribe, evitando la duplicidad de este y reutilizando los componentes a medida que se van creando.
* Otro de los puntos fuertes de Ruby on Rails, como ya hemos mencionado en el anterior apartado, es que aparte de ser de código abierto, la comunidad de desarrolladores que respalda esta herramienta es muy grande, ofreciendo un buen soporte y ayuda para nuevos desarrolladores que se adentren con este framework.
* Debido a su crecimiento en los últimos 10 años esta herramienta se ha convertido en la elección de muchos desarrolladores o empresas para crear su aplicación web, lo que ha hecho que actualmente muchas compañías demanden un perfil de desarrollador web más concreto con conocimiento de este framework.



Figura 19 Logo del fraemework Ruby On Rails

3.5.2 Tecnologías front-end

Las tecnologías elegidas que cubrirán las necesidades en el lado del cliente para este proyecto serán varias de las anteriormente mencionadas. HTML y CSS serán los lenguajes principales con las que el proyecto contará y por otro lado se ha decidido incluir la librería JavaScript y el framework de diseño Bootstrap, los cuales aportarán un diseño dinámico a la aplicación web con el objetivo de mejorar la experiencia de usuario.



Figura 20 Logos HTML5, Javascript y CSS3

3.5.3 Base de datos y gestor de base de datos

Un aspecto que condiciona en gran medida la elección del tipo de base de datos en el desarrollo de una aplicación web es el tipo de aplicación que vamos a crear y que finalidad tendrá. Debido a que nuestra aplicación contará con elementos que se deberán relacionar (usuarios de diferentes roles, mensajería entre usuarios, inscripciones en ofertas de prácticas de trabajo, procesos de selección, curriculums, etc.) el tipo de base de datos que utilicemos debe ser relacional.  
 Al haber escogido Ruby on Rails como framework principal de desarrollo, este nos brinda una base de datos relacional por defecto (SQLite3), la cual se puede sustituir fácilmente por cualquier otra. Según los estudios de las diferentes bases de datos relacionales que hemos hecho en el apartado anterior, PostgreSQL parece la base de datos más adecuada para integrar con el framework Ruby on Rails, ya que cuenta con un rendimiento mejor que MySQL.

Aparte de los resultados expuestos entre PostgreSQL y MySQL en el apartado anterior, otra de las ventajas que tiene utilizar PostgreSQL es que debido al crecimiento que ha tenido en los últimos años, las últimas versiones de Ruby on Rails incluyen un soporte para las diferentes funcionalidades que ofrece PostgreSQL como el tratamiento de arrays o la creación de diferentes tipos de claves primarias. Por último, otro aspecto a mencionar es la buena concurrencia que PostgreSQL proporciona al sistema si hay varios usuarios conectados a la aplicación web al mismo tiempo, característica esencial para aplicaciones con múltiples usuarios simultáneos.

Otra de las ventajas de escoger PostgreSQL como base de datos de nuestra aplicación es la herramienta pgAdmin III, entorno de escritorio visual que utilizaremos para gestionar la base de datos de la aplicación de forma rápida, sencilla y gráfica.



Figura 21 Logo PostgreSQL

3.5.4 Sistema operativo

Dos de los sistemas operativos más utilizados para el desarrollo de aplicaciones web con Ruby on Rails son Linux o mac OS. En este caso escogeremos Linux como sistema operativo, más concretamente la distribución Ubuntu 14.04 LTS, ya que esta es una de las distribuciones más sencillas de usar y cuenta con una amplia documentación.



Figura 22 Logo Ubuntu 14.04 LTS

# DOCUMENTO 4: ANÁLISIS DE REQUISITOS DEL SISTEMA

**D. MARTÍNEZ SUÁREZ, Wenceslao**

**TUTOR: Dña. SUAREZ CABAL, María José**

**FECHA: Julio 2017**

Índice de contenidos

[**DOCUMENTO 4: ANÁLISIS DE REQUISITOS DEL SISTEMA** 59](#_Toc486815176)

[4.1 INTRODUCCIÓN 61](#_Toc486815177)

[4.2 SUBSISTEMAS DE ANÁLISIS Y CASOS DE USO 61](#_Toc486815178)

[4.2.1 ACCESO DE USUARIOS 62](#_Toc486815179)

[4.2.1.1 REGISTRO DE USUARIO 62](#_Toc486815180)

[4.2.1.2 LOGIN DE USUARIO 63](#_Toc486815181)

[4.2.1.3 RECUPERACIÓN DE CONTRASEÑA DE USUARIO 64](#_Toc486815182)

[4.2.2 GESTIÓN DE LA INFORMACIÓN DE USUARIO 64](#_Toc486815183)

[4.2.2.1 MODIFICAR INFORMACIÓN BÁSICA DE CUENTA DE USUARIO 65](#_Toc486815184)

[4.2.2.2 CONFIGURAR PERFIL PROFESIONAL DE USUARIO 66](#_Toc486815185)

[4.2.2.3 BORRAR CUENTA DE USUARIO 66](#_Toc486815186)

[4.2.3 GESTIÓN DE OFERTAS DE PRÁCTICAS 67](#_Toc486815187)

[4.2.3.1 CREACIÓN DE OFERTAS DE PRÁCTICAS 68](#_Toc486815188)

[4.2.3.2. INSCRIPCIÓN EN OFERTAS DE PRÁCTICAS 69](#_Toc486815189)

[4.2.3.3 PROCESO DE SELECCIÓN – ROL DE ESTUDIANTE 69](#_Toc486815190)

[4.2.3.4 PROCESO DE SELECCIÓN – ROL DE EMPRESA 70](#_Toc486815191)

[4.2.3.5 BÚSQUEDA DE ESTUDIANTES – ROL DE EMPRESA 71](#_Toc486815192)

[4.2.3.6 BÚSQUEDA DE EMPRESAS – ROL DE ESTUDIANTE 72](#_Toc486815193)

[4.2.4 COMUNICACIÓN ENTRE USUARIOS 72](#_Toc486815194)

[4.2.4.1 ENVÍO DE MENSAJES ENTRE ESTUDIANTES Y EMPRESAS 72](#_Toc486815195)

[4.2.4.2 GESTIÓN DE MENSAJES 73](#_Toc486815196)

[4.3 REQUISITOS NO FUNCIONALES 74](#_Toc486815197)

[4.4 MODELO DE DATOS 74](#_Toc486815198)

[4.4.1 DESCRIPCIÓN DE LOS MODELOS 75](#_Toc486815199)

[4.4.1.1 SUBSISTEMA DE SESIÓN 75](#_Toc486815200)

[4.4.1.2 SUBSISTEMA DE GESTIÓN DE USUARIOS 75](#_Toc486815201)

[4.4.1.3 SUBSISTEMA DE GESTIÓN DE INFORMACIÓN DE USUARIO 76](#_Toc486815202)

[4.4.1.4 SUBSISTEMA DE GESTIÓN DE OFERTAS DE PRÁCTICAS 78](#_Toc486815203)

[4.4.1.5 SUBSISTEMA DE GESTIÓN DE MENSAJES 79](#_Toc486815204)

4.1 Introducción

En primer lugar, haremos una identificación de los subsistemas de análisis, con lo que descompondremos nuestro sistema en diferentes subsistemas con el objetivo de facilitar el análisis. Una vez identificados los distintos subsistemas procederemos a la obtención de requisitos de sistema utilizando la técnica de casos de uso.

Cabe destacar que los casos de uso describen las funciones que el sistema debe ejecutar cuando se realiza algún proceso concreto en él por parte de un usuario. Generalmente se desglosa en una secuencia de iteraciones las cuales se desarrollarán entre el propio sistema y un usuario, con el fin de especificar la comunicación y el comportamiento que debe tener el sistema a medida que el usuario va interactuando con él.

4.2 Subsistemas de análisis y casos de uso

A continuación se expondrán los subsistemas de la aplicación web que conformarán nuestro sistema principal:

1. Acceso de usuarios.
2. Gestión de la información de usuarios.
3. Gestión de ofertas de prácticas.
4. Comunicación entre usuarios.

A continuación se describirán los respectivos casos de uso de cada subsistema, los cuales irán previamente acompañados de diagramas para dar una visión inicial de cada subsistema.

4.2.1 Acceso de usuarios

A continuación se presentarán los casos de uso que analizaremos para el subsistema “Acceso de usuarios”, el cual engloba aquellas funcionalidades relacionadas con el acceso a la aplicación web (registro y login). Como podemos ver en el siguiente diagrama, este subsistema contará con dos casos de uso principales:

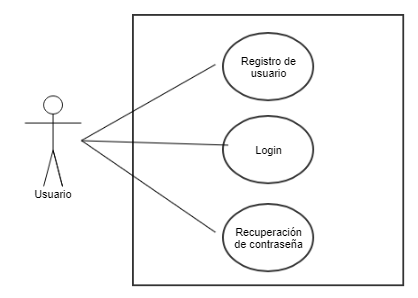


Figura 23 Subsistema acceso de usuarios

4.2.1.1 Registro de usuario

* Actores: estudiante o empresa.
* Precondiciones: que el usuario no esté registrado en la aplicación.
* Descripción: el usuario se podrá registrar en la aplicación, creando un perfil de estudiante o un perfil de empresa.
* Flujo normal del caso de uso:  
  1. El usuario accede a la pantalla principal de la aplicación web.  
  2. El usuario accede a la página de registro.  
  3. El usuario elige entre la opción de registrarse como estudiante o como empresa, accediendo al formulario de registro correspondiente.  
  4. El usuario rellena el formulario de registro.  
  5. El usuario envía el formulario rellenado.  
  6. El sistema valida la información y registra al usuario en la aplicación.  
  7. El sistema notifica al usuario de que se ha registrado correctamente.
* Excepciones:  
  5. Al enviar el formulario, si el usuario especifica un correo electrónico perteneciente a otro usuario ya registrado en la aplicación el sistema notificará al usuario este suceso comunicándole que introduzca otro correo electrónico. Lo mismo ocurrirá si el usuario no rellena alguno de los campos obligatorios del formulario (nombre, contraseña, edad, etc).  
  5. Si el usuario accede a otro apartado o página de la aplicación antes de darte al botón de guardar el formulario de registro no se enviará y el usuario no se registrará en el sistema.

4.2.1.2 Login de usuario

* Actores: empresa o estudiante.
* Precondiciones: el usuario debe estar registrado en la plataforma.
* Descripción:   
  El usuario podrá iniciar sesión en la aplicación web.
* Flujo normal del caso de uso:  
  1. El usuario accede a la pantalla principal de la aplicación web.  
  2. El usuario accede a la página de login.  
  3. El usuario rellena el formulario de login (email y contraseña).  
  4. El usuario envía el formulario rellenado.  
  5. El sistema valida los datos enviados por el usuario e inicia sesión al usuario.  
  5. El sistema notifica al usuario el correcto inicio de sesión.
* Excepciones:  
  4. Al enviar el formulario, si el usuario introduce la dirección de email o la contraseña de forma incorrecta, el sistema notificará al usuario este suceso comunicándole que introduzca de nuevo sus credenciales de forma correcta para poder iniciar sesión.  
  4. Si el usuario accede a otro apartado o página de la aplicación antes de darte al botón de enviar el formulario los datos introducidos no se enviará y el usuario no iniciará sesión.

4.2.1.3 Recuperación de contraseña de usuario

* Actores: empresa o estudiante.
* Precondiciones: el usuario debe estar registrado en la plataforma.
* Descripción:   
  El usuario podrá establecer una nueva contraseña en el caso de haberla extraviado.
* Flujo normal del caso de uso:  
  1. El usuario accede a la pantalla principal de la aplicación web.  
  2. El usuario accede a la página de login.  
  3. El usuario accede a la pantalla de recuperar contraseña y envía su dirección de correo electrónico.  
  4. El sistema le envía un email al correo electrónico establecido por el usuario con una URL única para reestablecer su contraseña.  
  5. El usuario abre el mail y accede a la URL generada por la aplicación.  
  6. El usuario establece una nueva contraseña y guarda los cambios.  
  7. El sistema actualiza la contraseña de usuario existente por la nueva contraseña.
* Excepciones:  
  3. Sí el usuario envía al sistema un correo electrónico no válido, este no podrá acceder al email de recuperación.  
  6. Si el usuario no accede a la URL generada y cambia la contraseña antes de dos horas de haber solicitado la recuperación de esta, el link expirará, con lo que el usuario tendrá que volver a solicitar la recuperación de contraseña.

4.2.2 Gestión de la información de usuario

A continuación se presentarán los casos de uso que analizaremos para el subsistema “Gestión de la información de usuario”, el cual engloba aquellas funcionalidades que un usuario puede establecer respecto a su información (información de cuenta o información académica o profesional). En el siguiente diagrama podemos ver los casos de uso principales de este subsistema.

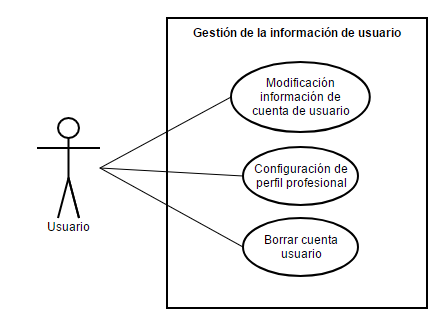


Figura 24 Subsistema de gestión de la información de usuario

4.2.2.1 Modificar información básica de cuenta de usuario

* Actores: empresa o estudiante.
* Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.
* Descripción: el usuario podrá cambiar la información básica de su cuenta (nombre, email, contraseña, fecha de nacimiento, etc.).
* Flujo normal del caso de uso:  
  1. El usuario accede a su perfil.  
  2. El usuario accede al apartado “Mi cuenta”.  
  3. El usuario modifica aquellos campos de su cuenta que desee cambiar.  
  4. El usuario guarda los cambios.  
  5. El sistema valida la información y actualiza los datos.  
  6. El sistema notifica al usuario del éxito de la operación.
* Excepciones:  
  4. Si el usuario accede a otro apartado o página de la aplicación antes de darte al botón de guardar los nuevos datos introducidos no se guardarán.

4.2.2.2 Configurar perfil profesional de usuario

* Actores: empresa o estudiante.
* Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.
* Descripción:   
  1. Rol de estudiante: el usuario podrá establecer aquella información de carácter profesional como estudios, experiencias laborales, títulos u otras cualidades.  
  2. Rol de empresa: el usuario podrá establecer aquella información de carácter profesional, como las actividades profesionales a las que se dedique.
* Flujo normal del caso de uso:  
  1. El usuario accede a su perfil.  
  2. El usuario accede al apartado “Mi CV”.  
  3. El usuario añadirá aquella información de carácter académica/profesional que desee establecer en su perfil, creando tantas competencias como crea necesario.  
  4. El usuario guardará los cambios.  
  5. El sistema guardará la información enviada por el usuario y notificará a este del éxito del guardado.
* Excepciones:  
  4. Si el usuario accede a otro apartado o página de la aplicación antes de darte al botón de guardar los nuevos datos introducidos no se guardarán.

4.2.2.3 Borrar cuenta de usuario

* Actores: empresa o estudiante.
* Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.
* Descripción:   
  El usuario podrá borrar su cuenta si ya no desea seguir formando parte de la plataforma, borrando todos los datos asociados que tenga (información, inscripciones a ofertas, ofertas creadas y mensajes).
* Flujo normal del caso de uso:  
  1. El usuario accede a su perfil.  
  2. El usuario accede al apartado “Mi cuenta”.  
  3. El usuario selecciona borrar su cuenta.   
  4. El usuario acepta la confirmación del cuadro de diálogo.  
  5. El sistema borra la cuenta del usuario, previamente expulsándole de la sesión iniciada.

4.2.3 Gestión de ofertas de prácticas

A continuación se presentarán los casos de uso que analizaremos para el subsistema “Gestión de ofertas de prácticas”, el cual engloba aquellas funcionalidades relacionadas con la gestión de ofertas de prácticas de trabajo desde el punto de vista del rol de estudiante (inscripción en ofertas, proceso de selección) hasta el punto de vista del rol de empresa (creación de ofertas, búsqueda de estudiantes, proceso de selección, etc.). En el siguiente diagrama podremos ver los diferentes casos de uso que veremos en este apartado:

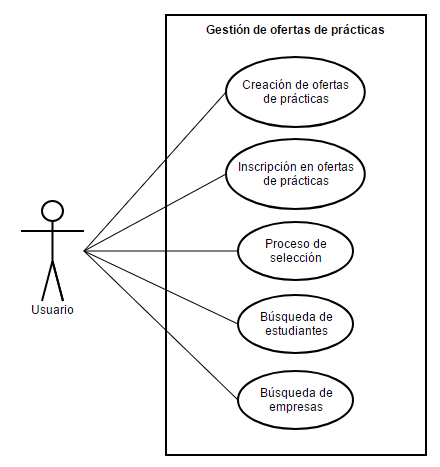


Figura 25 Gestión de las ofertas de prácticas.

4.2.3.1 Creación de ofertas de prácticas

* Actores: empresa.
* Precondiciones: el usuario debe de estar registrado en la aplicación web y haber iniciado sesión.
* Descripción: la empresa podrá crear aquellas ofertas de trabajo que desee, especificando correctamente los requerimientos de estas.
* Flujo normal del caso de uso:  
  1. El usuario empresa accede a su perfil.  
  2. El usuario empresa accede al apartado “Mis ofertas”.  
  3. El usuario empresa creará la oferta rellenando los datos de esta según sus características.  
  4. El usuario guardará la oferta creada.  
  5. El sistema validará la información enviada por la empresa respecto a la oferta y la publicará, notificando al usuario de esto.
* Excepciones:  
  4. Si el usuario cambia de página antes de presionar en el botón “Publicar”, la oferta no se creará.  
  5. Al enviar el formulario de la oferta, si el usuario no introduce aquellos campos requeridos o los introduce de forma incorrecta, el sistema notificará al usuario este suceso comunicándole que introduzca los datos de forma correcta para poder crear la oferta.

4.2.3.2. Inscripción en ofertas de prácticas

* Actores: estudiante.
* Precondiciones: el usuario debe de estar registrado en la aplicación web y haber iniciado sesión.
* Descripción: el usuario podrá inscribirse y también buscar aquellas ofertas que sean de su interés. En el caso de utilizar el buscador, podrá filtrar la búsqueda por diferentes características (perfil requerido, comunidad autónoma, remuneración, estado de la oferta).
* Flujo normal del caso de uso:  
  1. El usuario accede a la página de ofertas  
  3. El usuario podrá ver la información detallada de cualquiera de las ofertas que se le presentan en la página principal de ofertas seleccionando cualquiera de ellas  
  4. Una vez accedido a la oferta de prácticas, el usuario se puede inscribir una oferta de prácticas (siempre que el proceso de inscripción no haya cerrado).  
  5. El sistema inscribirá al usuario en el proceso de selección de la oferta de prácticas y se lo notificará.
* Flujo alternativo:  
  3. En caso de que el usuario estudiante quiera buscar ofertas concretas, podrá hacer una búsqueda filtrada de aquellas ofertas por perfil requerido, comunidad autónoma, remuneración o estado de la oferta.

4.2.3.3 Proceso de selección – Rol de estudiante

* Actores: estudiante.
* Precondiciones: el estudiante debe de estar registrado en la aplicación web, haber iniciado sesión y haberse inscrito en alguna oferta de prácticas de trabajo.
* Descripción: una vez cerrado el plazo de inscripción de la/s oferta/s de prácticas en las que el estudiante se haya inscrito, este podrá ver en todo momento como avanza el proceso de selección de la/s misma/s.
* Flujo normal del caso de uso:  
  1. El estudiante accede a su perfil.  
  2. El estudiante accede al apartado “Mis ofertas”.  
  3. El estudiante tendrá un listado de aquella/s oferta/s a la/s que se ha apuntado, pudiendo ver el estado de la misma en todo momento.  
  4. Una vez cerrado el plazo de inscripción de la oferta de prácticas el estudiante entrará automáticamente en el proceso de selección de la misma.  
  5. El estudiante avanzará a la primera fase del proceso de selección una vez que la empresa elija al estudiante para avanzar a la siguiente fase.  
  6. Una vez acabada la primera fase, el estudiante pasará a la fase final donde será finalmente seleccionado por la empresa para disfrutar de la oferta de prácticas. El sistema le notificará al estudiante con un mensaje interno así como también enviará un email a su correo personal establecido en su cuenta para avisarle de que tiene una notificación en la aplicación.
* Flujo alternativo:  
  primera faseel estudiante no es seleccionado por avanzar a la fase final  
  6. Una vez acabada la fase final, el estudiante no es seleccionado por la empresa para disfrutar de la beca de prácticas. la empresa no selecciona al estudiante para avanzar a la fase final. El sistema se lo notificará al estudiante con un mensaje interno y con otro email a su correo personal.
* Excepciones:  
  3. El usuario puede cancelar la inscripción del proceso de selección de la oferta en cualquier momento, quedando fuera del proceso sin posibilidad de volver a entrar en él.

4.2.3.4 Proceso de selección – Rol de empresa

* Actores: empresa.
* Precondiciones: el usuario debe de estar registrado en la aplicación web, haber iniciado sesión y haber creado alguna oferta de prácticas de trabajo.
* Descripción: una vez cerrado el plazo de inscripción de la/s oferta/s de prácticas que la empresa haya creado, esta podrá ver en todo momento como avanza el proceso de selección de la/s misma/s y gestionarlo/s.
* Flujo normal del caso de uso:  
  1. La empresa accede a su perfil.  
  2. La empresa accede al apartado “Mis ofertas”.  
  3. El sistema proporcionará a la empresa tendrá un listado de aquella/s oferta/s que haya creado, pudiendo ver el estado de la misma en todo momento y la información relativa a esta (estudiantes inscritos, perfiles de estos estudiantes, etc.).  
  4. Una vez finalizado la fase de inscripción de estudiantes en la oferta, el sistema comenzará la primera fase del proceso de selección, donde la empresa debe elegir aquellos usuarios (en un plazo de 15 días) que pasen a la fase final.  
  5. Una vez acabada la primera fase, el sistema comenzará la fase final (15 días), donde la empresa deberá seleccionar al estudiante que finalmente disfrutará de la oferta de prácticas.  
  6. Una vez acabada la fase final, el sistema cerrará el proceso de selección y se lo notificará a la empresa por medio de un mensaje interno y un email a su correo personal.
* Flujo alternativo:  
  4. En el caso de que la empresa no seleccione ningún estudiante durante el plazo de la primera fase del proceso de selección, la duración de esta fase se ampliará en 15 días más.  
  5. En caso de que la empresa no elija un usuario final en la fase final del proceso de selección, esta fase se ampliará 15 días más de duración.
* Excepciones:  
  La empresa El sistema notificará a los estudiantes inscritos de esto por medio de un mensaje.

4.2.3.5 Búsqueda de estudiantes – Rol de empresa

* Actores: empresa
* Precondiciones: el usuario debe de estar registrado en la aplicación web, haber iniciado sesión.
* Descripción: la empresa podrá ver y buscar aquellos estudiantes que sean de su interés. En el caso de utilizar el buscador, se podrá filtrar la búsqueda por diferentes características (perfil académico, comunidad autónoma o cualquier palabra clave).
* Flujo normal del caso de uso:  
  1. La empresa accede a la pantalla principal de la aplicación web.  
  2. La empresa accede a la página de Estudiantes.  
  3. La empresa podrá ver la información detallada de cualquiera de los estudiantes que se le presentan en la página principal de estudiantes seleccionándolos.
* Flujo alternativo:  
  3. En caso de que el usuario empresa quiera buscar estudiantes concretos, podrá hacer una búsqueda filtrada de aquellos estudiantes por perfil requerido, comunidad autónoma o cualquier palabra clave, como por ejemplo: php, desarrollo web, autocad, java, etc.

4.2.3.6 Búsqueda de empresas – Rol de estudiante

* Actores: estudiante.
* Precondiciones: el usuario debe de estar registrado en la aplicación web, haber iniciado sesión.
* Descripción: el estudiante podrá ver y buscar aquellas empresas que sean de su interés. En el caso de utilizar el buscador, se podrá filtrar la búsqueda por diferentes características (sector de actividad, comunidad autónoma o cualquier palabra clave).
* Flujo normal del caso de uso:  
  1.El estudiante accede a la página de Empresas.  
  3. El usuario estudiante podrá ver la información detallada de cualquiera de las empresas que se le presentan en la página principal de empresas, seleccionando la que desee ver.
* Flujo alternativo:  
  3. En caso de que el estudiante quiera buscar empresas concretos, podrá hacer una búsqueda filtrada de aquellas empresas por sector de actividad, comunidad autónoma o cualquier palabra clave como por ejemplo: php, desarrollo web, autocad, java, etc.

4.2.4 Comunicación entre usuarios

A continuación se presentarán los casos de uso que analizaremos para el subsistema “Comunicación de usuarios”, el cual engloba aquellas funcionalidades relacionadas con la comunicación entre los usuarios del sistema (envío de mensajes).

4.2.4.1 Envío de mensajes entre estudiantes y empresas

* Actores: estudiante y empresa.
* Precondiciones: el usuario debe de estar registrado en la aplicación web y haber iniciado sesión.
* Descripción: el estudiante podrá enviar aquellos mensajes que desee a cualquier empresa y viceversa.
* Flujo normal del caso de uso:  
  1. El estudiante/empresa accede a la pantalla principal de la aplicación web.  
  2. El estudiante/empresa accede a la página de estudiantes o empresas (según le corresponda.  
  3. El estudiante/empresa podrá ver la información detallada de cualquiera de los estudiantes/empresas que se le presentan en la página principal.  
  4. Una vez dentro de la página del usuario deseado, se le podrá enviar un mensaje (con asunto y cuerpo del mensaje).  
  5. Una vez rellenado el asunto y cuerpo del mensaje, se podrá enviar el mensaje al destinatario.  
  6. El sistema mandará el mensaje interno al destinatario así como también un email a su cuenta de correo personal notificándole de que tiene un mensaje nuevo.
* Flujo alternativo:  
  5. En caso de que el usuario que manda el mensaje cambie de página o no envíe el mensaje una vez escrito, este no se mandará ni tampoco el email a su cuenta personal.

4.2.4.2 Gestión de mensajes

* Actores: usuario (rol de estudiante y rol de empresa)
* Precondiciones: el usuario debe de estar registrado en la aplicación web y haber iniciado sesión.
* Descripción: el usuario podrá ver aquellos mensajes que reciba o aquellos que él envíe. El usuario también podrá responder a aquellos mensajes que reciba.
* Flujo normal del caso de uso:  
  1. El usuario accede a su perfil.  
  2. El usuario accede al apartado “Mis mensajes”.  
  3. El usuario podrá ver aquellos mensajes que haya recibido o aquellos que haya enviado.  
  4. El usuario podrá responder a aquellos mensajes que haya recibidos haciendo click en “Ver mensaje” y en “Responder mensaje” una vez haya accedido al mensaje.  
  5. Una vez rellenado el asunto y cuerpo del mensaje, se hará click en “Enviar” para que el mensaje se envíe a su destinatario.  
  6. El sistema mandará el mensaje a su destinatario, así como también le enviará un email a su correo personal notificándole que tiene un nuevo mensaje.
* Flujo alternativo:  
  4. En caso de que el usuario que manda el mensaje cambie de página o no le de al botón “Enviar” una vez escrito el mensaje, este no se mandará.

4.3 Requisitos no funcionales

Este tipo de requisitos componen una parte importante del análisis de requisitos del sistema, ya que estos representan las características generales y restricciones de nuestra aplicación. Se centran en aspectos como la eficiencia, seguridad, usabilidad, rendimiento, etc., que el sistema debe garantizar.

A continuación se detallarán aquellos requisitos no funcionales del sistema que deben estar presentes para garantizar la mayor calidad de la aplicación web.

1. La aplicación debe proporcionar una experiencia de usuario positiva a este, durante y después de la iteración con esta.
2. La aplicación debe garantizar el concepto de usabilidad al usuario permitiéndole alcanzar el objetivo que busca sin dificultades
3. La aplicación debe garantizar la total protección de los datos de acceso del usuario, utilizando la encriptación de contraseñas de usuario con la función Hash Bcrypt.
4. El sistema ocultará aquellas funcionalidades solo proporcionadas para los usuarios registrados (creación de ofertas, inscripción en ofertas o envío de mensajes).

Los requisitos no funcionales vistos anteriormente que tienen que ver con la experiencia de usuario así como la usabilidad de la aplicación se llevarán a cabo mediante el seguimiento de una guía heurística basada en diferentes criterios de usabilidad y accesibilidad que cubrirán aquellas características que debe tener una aplicación web.

4.4 MODELO DE DATOS

A continuación expondremos el modelo de datos de la aplicación web, el cual se diseña con el objetivo de tener una representación abstracta de los datos de la aplicación y las relaciones existentes entre ellos.

4.4.1 DESCRIPCIÓN DE LOS MODELOS

Para representar correctamente la estructura del modelo de datos de la aplicación, haremos una descomposición de este en diferentes modelos los cuales están catalogados en diferentes subsistemas.

4.4.1.1 Subsistema de sesión

Tabla 2 Modelo de sesión.

|  |
| --- |
| Nombre |
| **Session** |
| Descripción y finalidad |
| Este modelo guarda la sesión del usuario en el sistema. Siempre que el usuario inicie sesión en la aplicación el sistema creará en la base de datos una sesión única que lo identifique, comprobando que esté registrado en base de datos. Cuando el usuario finalice sesión (logout) el sistema borrará la sesión creada de la bbdd. |
| Atributos |
| N/A |

4.4.1.2 Subsistema de gestión de usuarios

Tabla 3 Modelo de usuario

|  |
| --- |
| Nombre |
| **User** |
| Descripción y finalidad |
| Este modelo se utiliza para representar los diferentes usuarios de nuestra aplicación, ya sean estudiantes o empresas. Se encargará de validar a los usuarios para que sean únicos respecto a su email y contraseña, de asignarles el rol correspondiente y también guardar la principal información de este. |
| Atributos |
| **id** :integer **name** :character varying **email** :character varying **password\_digest** :character varying **created\_at** :timestamp  **updated\_at** :timestamp  **role\_id** :integer **last\_name** :character varying **birth\_date** :character varying **province** :integer **city** :character varying **deleted\_at** :timestamp  **universidad** :integer **grado** :integer **description** :character varying |

Tabla 4 Modelo de rol.

|  |
| --- |
| Nombre |
| **Role** |
| Descripción y finalidad |
| Este modelo se utiliza para almacenar los diferentes roles con los que contará nuestra aplicación. Cada vez que se cree un usuario se referenciará a este modelo para asignarle un rol u otro en función de las características del usuario. |
| Atributos |
| **id** :integer **name** :character varying **created\_at** :timestamp  **updated\_at** :timestamp |

4.4.1.3 Subsistema de gestión de información de usuario

Tabla 5 Modelo de estudios

|  |
| --- |
| Nombre |
| **Study** |
| Descripción y finalidad |
| Este modelo se utiliza para almacenar los diferentes estudios existentes (grados y másteres) con los que contará la aplicación. Según la elección del estudiante al configurar su perfil académico o según la empresa al configurar el perfil de trabajador que busca, se hará referencia a este modelo para establecer las referencias. Este modelo será importante a la hora de realizar búsquedas filtradas, ya que nos permitirá filtrar estudiantes, empresas o ofertas laborales en función de los estudios. |
| Atributos |
| **id** :integer **name** :character varying **created\_at** :timestamp  **updated\_at** :timestamp |

Tabla 6 Modelo de universidad

|  |
| --- |
| Nombre |
| **University** |
| Descripción y finalidad |
| Este modelo se utiliza para almacenar las diferentes universidades españolas existentes en la actualidad, permitiendo así al estudiante escoger la universidad a la que pertenece entre una gran cantidad de posibilidades. Este modelo será importante a la hora de realizar búsquedas filtradas de estudiantes, ya que nos permitirá filtrar estos por la universidad donde estén estudiando. |
| Atributos |
| **id** :integer **name** :character varying **created\_at** :timestamp  **updated\_at** :timestamp |

Tabla 7 Modelo de provincia

|  |
| --- |
| Nombre |
| **Province** |
| Descripción y finalidad |
| Este modelo se utiliza para almacenar las diferentes comunidades autónomas que conforman el territorio español, permitiendo así al estudiante y a la empresa escoger la comunidad autónoma a la que pertenezcan o se encuentren. Este modelo será importante a la hora de realizar búsquedas filtradas de estudiantes, empresas y ofertas, ya que nos permitirá filtrar estos por la comunidad autónoma que hayan establecido. |
| Atributos |
| **id** :integer **name** :character varying **created\_at** :timestamp  **updated\_at** :timestamp |

Tabla 8 Modelo de cv\_skills

|  |
| --- |
| Nombre |
| **CV\_Skills** |
| Descripción y finalidad |
| Este modelo se utiliza para representar y almacenar las diferentes competencias profesionales que el estudiante establece y las cuales formarán parte de su perfil profesional dentro de la aplicación. |
| Atributos |
| **id** :integer **user\_id** :integer **nombre** :character varying **description** :character varying **date** :timestamp  **created\_at** :timestamp  **updated\_at** :timestamp |

4.4.1.4 Subsistema de gestión de ofertas de prácticas

Tabla 9 Modelo de oferta

|  |
| --- |
| Nombre |
| **Offer** |
| Descripción y finalidad |
| Este modelo se utiliza para representar y almacenar las diferentes ofertas creadas por las empresas en la aplicación. |
| Atributos |
| **id** :integer **user\_id** :integer **name** :character varying **perfil\_id** :integer **description** :character varying **duratio**n :character varying **start\_date** :character varying **finish\_date** :character varying **hours** :integer **location** :character varying **salary** :integer **created\_at** :timestamp  **updated\_at** :timestamp  **requirement**s :character varying **finish\_inscription\_date** :character varying |

Tabla 10 Modelo de inscripción en oferta (proceso de selección)

|  |
| --- |
| Nombre |
| **Offer\_inscriptions** |
| Descripción y finalidad |
| Este modelo se utiliza para representar los diferentes procesos de selección pertenecientes a las ofertas de prácticas creadas por las empresas en la aplicación. |
| Atributos |
| **id** : integer **offer\_id** : integer **user\_id** : integer **users\_id\_phase1** : integer **phase1** : boolean **phase1\_date** :timestamp **users\_ids\_phase2** :integer **phase2** : boolean **phase2\_date** :timestamp  **end\_date** :timestamp  **winner\_id** :integer **created\_at** :timestamp **updated\_at** :timestamp **ended** :boolean |

4.4.1.5 Subsistema de gestión de mensajes

Tabla 11 Modelo de mensajes

|  |
| --- |
| Nombre |
| **Messages** |
| Descripción y finalidad |
| Este modelo se utiliza para representar y almacenar lo diferentes mensajes que los usuarios se envían en el sistema. |
| Atributos |
| **id** :integer **sender\_id** :integer **recipient\_id** :integer **time\_send** :timestamp **subjec**t :character varying **body** :character varying **readed** :Boolean **created\_at** :timestamp **updated\_at** :timestamp |

Por último y para ver unificados los diferentes modelos anteriormente expuestos, utilizaremos el modelo entidad-relación o también conocido como diagrama entidad relación, de tal forma que veremos los datos de una forma unificada, centrándonos en las estructuras lógicas y dejando a un lado los aspectos físicos de la aplicación.

Este modelo se centra en entidades y en sus relaciones. Las entidades que son aquellas personas, objetos o conceptos que son distinguibles de los demás objetos (una persona, un coche, una empresa), los cuales tendrán atributos que las definan y las relaciones son aquellas asociaciones existentes entre dos o más entidades.

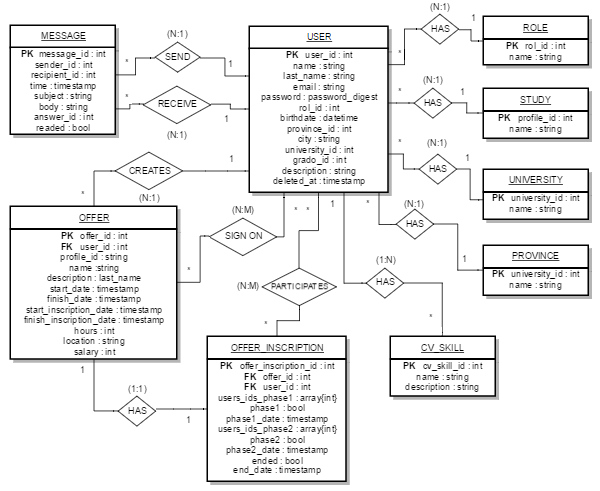


Figura 26 Diagrama E/R de la base de datos

# DOCUMENTO 5: DISEÑO DEL SISTEMA

**D. MARTÍNEZ SUÁREZ, Wenceslao**

**TUTOR: Dña. SUAREZ CABAL, María José**

**FECHA: Julio 2017**

Índice de contenidos

[**DOCUMENTO 5: DISEÑO DEL SISTEMA** 82](#_Toc486815205)

[5.1 INTRODUCCIÓN 85](#_Toc486815206)

[5.2 DIAGRAMA DE PAQUETES 85](#_Toc486815207)

[5.2.1 USER 86](#_Toc486815208)

[5.2.2 VISTAS 86](#_Toc486815209)

[5.2.3 OFFERS 86](#_Toc486815210)

[5.2.4 OFFER\_INSCRIPTIONS 86](#_Toc486815211)

[5.2.5 MESSAGES 86](#_Toc486815212)

[5.2.6 SESSIONS 87](#_Toc486815213)

[5.3 DIAGRAMA DE CLASES 87](#_Toc486815214)

[5.4 DIAGRAMA DE INTERACCIÓN 88](#_Toc486815215)

[5.5 DISEÑO DE LA BASES DE DATOS 90](#_Toc486815216)

[5.5.1 SISTEMA GESTOR DE LA BASE DE DATOS 90](#_Toc486815217)

[5.5.2 INTEGRACIÓN DEL SGBD EN NUESTRO SISTEMA 90](#_Toc486815218)

[5.6 DISEÑO DE LA INTERFAZ 93](#_Toc486815219)

[5.6.1 INTERFACES PÚBLICOS 93](#_Toc486815220)

[5.6.1.1 PANTALLA DE BIENVENIDA O HOME 94](#_Toc486815221)

[5.6.1.2 PANTALLA DE LOGIN 94](#_Toc486815222)

[5.6.1.3 PANTALLA DE RECUPERAR CONTRASEÑA 95](#_Toc486815223)

[5.6.1.4 PANTALLA DE REGISTRO 97](#_Toc486815224)

[5.6.2 INTERFACES PRIVADOS 99](#_Toc486815225)

[5.6.2.1 INTERFACES DEL ROL DE ESTUDIANTE 99](#_Toc486815226)

[5.6.2.1.1 PANTALLA DE INICIO 99](#_Toc486815227)

[5.6.2.1.2 PANTALLA DE OFERTAS 100](#_Toc486815228)

[5.6.2.1.3 PANTALLA DE EMPRESAS Y ENVÍO DE MENSAJES 103](#_Toc486815229)

[5.6.2.1.4 PANTALLA DE PERFIL 106](#_Toc486815230)

[5.6.2.1.5 CONFIGURACIÓN DE CUENTA 107](#_Toc486815231)

[5.6.2.1.6 PERFIL ACADÉMICO 107](#_Toc486815232)

[5.6.2.1.7 OFERTAS DE USUARIO 109](#_Toc486815233)

[5.6.2.1.8 MENSAJES 109](#_Toc486815234)

[5.6.2.2 INTERFACES DEL ROL DE EMPRESA 110](#_Toc486815235)

[5.6.2.2.1 PANTALLA DE INICIO 110](#_Toc486815236)

[5.6.2.2.2 PANTALLA DE ESTUDIANTES Y ENVÍO DE MENSAJES 111](#_Toc486815237)

[5.6.2.2.3 PANTALLA DE OFERTAS 113](#_Toc486815238)

[5.6.2.2.4 PANTALLA DE PERFIL 113](#_Toc486815239)

[5.6.2.2.5 PANTALLA DE CONFIGURACIÓN DE CUENTA 114](#_Toc486815240)

[5.6.2.2.6 PANTALLA DE OFERTAS 115](#_Toc486815241)

[5.6.2.2.7 PANTALLA DE PROCESOS DE SELECCIÓN 116](#_Toc486815242)

[5.6.2.2.8 PANTALLA DE MENSAJES 119](#_Toc486815243)

## 5.1 Introducción

En este apartado nos centraremos en el diseño de la aplicación web, describiendo los diferentes diagramas (de paquetes, clases, de interacción) esenciales, así como el diseño de la base de datos y los diferentes diseños preliminares y definitivos de las interfaces de usuario, lo que ayudará a comprender mejor el proceso de diseño que se ha seguido en el desarrollo de esta aplicación web.

5.2 Diagrama de paquetes

Como ya hemos mencionado y descrito anteriormente, nuestra aplicación web sigue el patrón de diseño MVC (Modelo-Vista-Controlar), por lo que los distintos módulos de la aplicación van a estar claramente diferenciados y separados lógicamente.

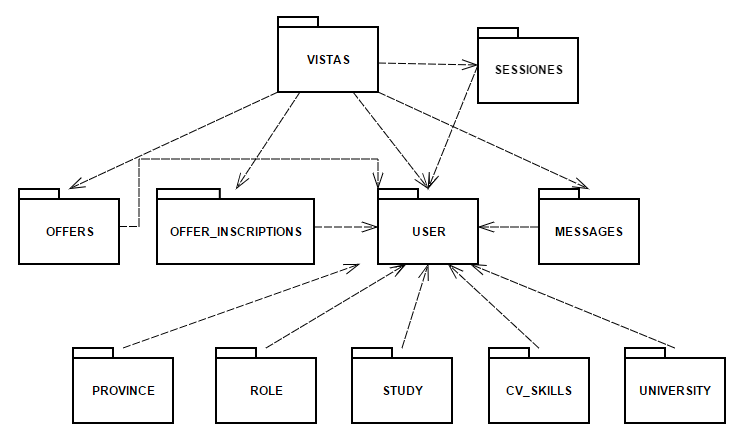


Figura 27 Diagrama de paquetes del sistema

A continuación se hará una breve descripción de los paquetes más importantes de esta aplicación:

5.2.1 User

Este paquete engloba las vistas, modelo y controlador de usuario, los cuales son necesarios para la gestión de estos (registrar usuarios, configuración de cuenta, configuración de perfil profesional, etc). Los usuarios podrán tendrán asociadas (dependiendo del rol que tenga) varias competencias de usuario (cv\_skills), una provincia (province), unos estudios principales (study), un rol (role) y una universidad (university). También puede tener asociadas varias ofertas de prácticas (offers) o participar en varios procesos de selección (offer\_inscriptions). Así mismo los usuarios pueden recibir o enviar varios mensajes (messages) diferentes.

5.2.2 Vistas

Este paquete forma la parte gráfica de la aplicación web, es decir, la capa de presentación de la aplicación web. Las vistas se encargan de recibir los datos de la aplicación enviados desde el controlador para mostrárselos al usuario.

5.2.3 Offers

Este paquete engloba las vistas, modelo y controlador de las ofertas de prácticas, las cuales son necesarios para la creación y gestión de estas. Las ofertas de prácticas están asociadas a un usuario con rol de empresa y tienen asociadas un proceso de selección (offer\_inscription).

5.2.4 Offer\_inscriptions

Este paquete engloba las vistas, modelo y controlador de los procesos de selección pertenecientes a las ofertas de prácticas, las cuales son necesarios para la gestión de estos. Estos procesos de selección están asociados a una oferta de prácticas y a un usuario con rol de empresa. Cada proceso de selección se especificará en diferentes fases mediante intervalos de tiempo fijados por el sistema.

5.2.5 Messages

Este paquete engloba las vistas, modelo y controlador de la gestión de mensajes entre usuarios. Un usuario podrá enviar un mensaje a un usuario así como recibir todos aquellos mensajes que le envíen otros usuarios.

5.2.6 Sessions

Es el paquete encargado de gestionar las sesiones de usuario en la aplicación web. Se encarga de guardar en base de datos las sesiones iniciadas de cada usuario y de validar si la sesión es la correcta mediante el identificador único que posee cada sesión.

5.3 Diagrama de clases

En el apartado “4.4 Modelo de datos” del documento 4 “Análisis de requisitos del sistema” definimos el modelos de datos que debería seguir la aplicación web mediante la descripción individual de cada modelo y posteriormente con un diagrama entidad-relación que relacionase estos. Este apartado nos dio una idea general y bastante cercana del modelo de datos que debe seguir la aplicación y por consiguiente sus correspondientes clases, ya que estas se encargan de representar a las tablas de la base de datos.

A continuación se expondrá un diagrama de clases generado con la herramienta RailRoady, la cual sirve para generar automáticamente y de forma muy sencilla diagramas de clases de cualquier aplicación web que se desarrolle en Ruby on Rails. Esta herramienta proporciona un archivo “.svg” que puede ser leído por cualquier navegador o programa de tratamiento de imágenes. A continuación se adjunta el diagrama generado de esta aplicación web:



Figura 28 Diagrama de clases de la aplicación.

5.4 Diagrama de interacción

En este apartado se presentará el diagrama de interacción del sistema el cual define como se va a llevar a cabo la interacción del usuario con el sistema de forma general. Este diagrama se verá fuertemente influenciado por el patrón MVC, ya que cada vez que el usuario interactúe con el sistema éste seguirá el diseño del patrón MVC en el tratamiento de la información requerida por el usuario.

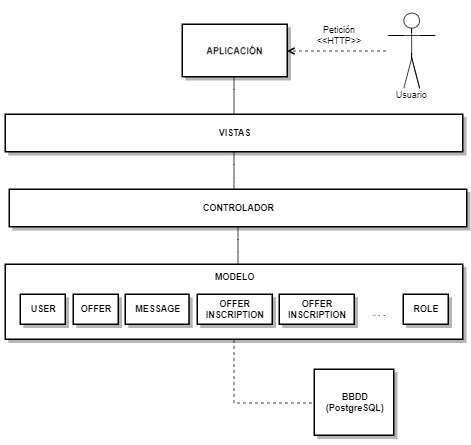


Figura 29 Diagrama de interacción sistema-usuario

Cuando el usuario realice algún tipo de petición HTTP a la aplicación web a través de cualquiera de sus vistas el controlador será quién reciba la notificación del evento hecho por el usuario, encargándose posteriormente de gestionarlo.

El controlador seguidamente accederá al modelo, ya sea para crear o modificar en él algún dato pedido por el usuario (crear oferta de prácticas, modificar competencia profesional, etc) o bien para solicitar información de algún modelo (ver perfil de usuario estudiante, ver información de una oferta de prácticas, etc.). El modelo es quién se encarga de gestionar los datos de la aplicación, sin embargo este no tiene conocimiento directo sobre las vistas.

Por último y una vez obtenidos los datos necesarios del modelo, el controlar delega toda la información obtenida a las correspondientes vistas que deberán generar la interfaz apropiada para el usuario.

5.5 Diseño de la bases de datos

A continuación se hará una descripción del sistema de gestión usado para la base de datos de la aplicación así como la integración de esta en nuestro sistema.

5.5.1 Sistema Gestor de la Base de Datos

Como ya hemos mencionado anteriormente el sistema gestor escogido que administrará la base de datos PostgreSQL de nuestra aplicación es el pgAdmin III, el cual proporciona una librería gráfica multiplataforma lo que permite usarse en Linux, Windows, Mac OS, Solaris y muchos más sistemas operativos.

Este gestor tiene multitud de características, desde permitir al usuario escribir consultas SQL simples hasta desarrollar bases de datos complejas. También incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y muchas más características.

Sin duda con este gestor y su multitud de características garantizaremos una buena gestión de nuestra base de datos.

5.5.2 Integración del SGBD en nuestro sistema

A continuación se mostrará la base de datos de nuestra aplicación web, llamada tfg\_v1\_development y la cual estará compuesta por las tablas correspondientes a los modelos que hemos visto en los apartados anteriores. Uno de los archivos principales de todo proyecto en Ruby on Rails es el “schema.rb” archivo el cual representa en código fuente la estructura de la base de datos de nuestro proyecto. Gracias a este archivo tendremos una visión actualizada de nuestra base de datos en todo momento, hagamos cambios en ella o no.

ActiveRecord**::**Schema**.**define**(**version: 20170620162038**)** **do**

# These are extensions that must be enabled in order to support this database

enable\_extension "plpgsql"

create\_table "cv\_skills"**,** force: :cascade **do** **|**t**|**

t**.**integer "user\_id"

t**.**string "nombre"

t**.**string "description"

t**.**datetime "date"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

t**.**index **[**"user\_id"**],** name: "index\_cv\_skills\_on\_user\_id"**,** using: :btree

**end**

create\_table "studies"**,** force: :cascade **do** **|**t**|**

t**.**string "name"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

**end**

create\_table "messages"**,** force: :cascade **do** **|**t**|**

t**.**integer "sender\_id"

t**.**integer "recipient\_id"

t**.**datetime "time\_send"

t**.**string "subject"

t**.**string "body"

t**.**boolean "readed"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

**end**

create\_table "offer\_inscriptions"**,** force: :cascade **do** **|**t**|**

t**.**integer "offer\_id"

t**.**integer "user\_id"

t**.**integer "users\_id\_phase1"**,** default: **[],** array: **true**

t**.**boolean "phase1"

t**.**datetime "phase1\_date"

t**.**integer "users\_ids\_phase2"**,** default: **[],** array: **true**

t**.**boolean "phase2"

t**.**datetime "phase2\_date"

t**.**datetime "end\_date"

t**.**integer "winner\_id"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

t**.**boolean "ended"

t**.**index **[**"offer\_id"**],** name: "index\_offer\_inscriptions\_on\_offer\_id"**,** using: :btree

t**.**index **[**"user\_id"**],** name: "index\_offer\_inscriptions\_on\_user\_id"**,** using: :btree

**end**

create\_table "offers"**,** force: :cascade **do** **|**t**|**

t**.**integer "user\_id"

t**.**string "name"

t**.**integer "perfil\_id"

t**.**string "description"

t**.**string "duration"

t**.**string "start\_date"

t**.**string "finish\_date"

t**.**integer "hours"

t**.**string "location"

t**.**integer "salary"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

t**.**string "requirements"

t**.**string "finish\_inscription\_date"

t**.**index **[**"user\_id"**],** name: "index\_offers\_on\_user\_id"**,** using: :btree

**end**

create\_table "provinces"**,** force: :cascade **do** **|**t**|**

t**.**string "name"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

**end**

create\_table "roles"**,** force: :cascade **do** **|**t**|**

t**.**string "name"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

**end**

create\_table "universidads"**,** force: :cascade **do** **|**t**|**

t**.**string "name"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

**end**

create\_table "users"**,** force: :cascade **do** **|**t**|**

t**.**string "name"

t**.**string "email"

t**.**string "password\_digest"

t**.**datetime "created\_at"**,** null: **false**

t**.**datetime "updated\_at"**,** null: **false**

t**.**integer "role\_id"

t**.**string "last\_name"

t**.**string "birth\_date"

t**.**integer "province"

t**.**string "city"

t**.**datetime "deleted\_at"

t**.**integer "universidad"

t**.**integer "grado"

t**.**string "description"

t**.**index **[**"role\_id"**],** name: "index\_users\_on\_role\_id"**,** using: :btree

**end**

add\_foreign\_key "cv\_skills"**,** "users"

add\_foreign\_key "offer\_inscriptions"**,** "offers"

add\_foreign\_key "offer\_inscriptions"**,** "users"

add\_foreign\_key "offers"**,** "users"

**end**

5.6 Diseño de la interfaz

A continuación se mostrará detalladamente el diseño de las diferentes interfaces de esta aplicación web, desde los interfaces genéricos hasta los diferentes interfaces diseñados para cada rol de usuario.

Todos las interfaces han sido diseñadas utilizando el lenguaje CSS y aprovechando los diseños que proporciona el framework de diseño Bootstrap.

5.6.1 Interfaces públicos

En primer lugar detallaremos aquellos interfaces que son públicos, es decir, están visibles a cualquier usuario que interactúe con la aplicación sin tener que estar registrado o haber iniciado sesión.

5.6.1.1 Pantalla de bienvenida o home

La pantalla de bienvenida o home ha sido diseñada con el objetivo de dar una imagen sencilla a la vez que moderna al usuario que acceda por primera vez a la aplicación, así como también introducir el objetivo principal de esta y hacia quién va destinada, animando al usuario a registrarse en ella.



Figura 30 Captura de pantalla de la pantalla principal de la aplicación

5.6.1.2 Pantalla de login

La pantalla de login sigue un diseño simple a la vez que limpio, donde el usuario deberá introducir sus credenciales de acceso (email y login) para acceder a la parte privada de la aplicación en el caso de estar registrado.

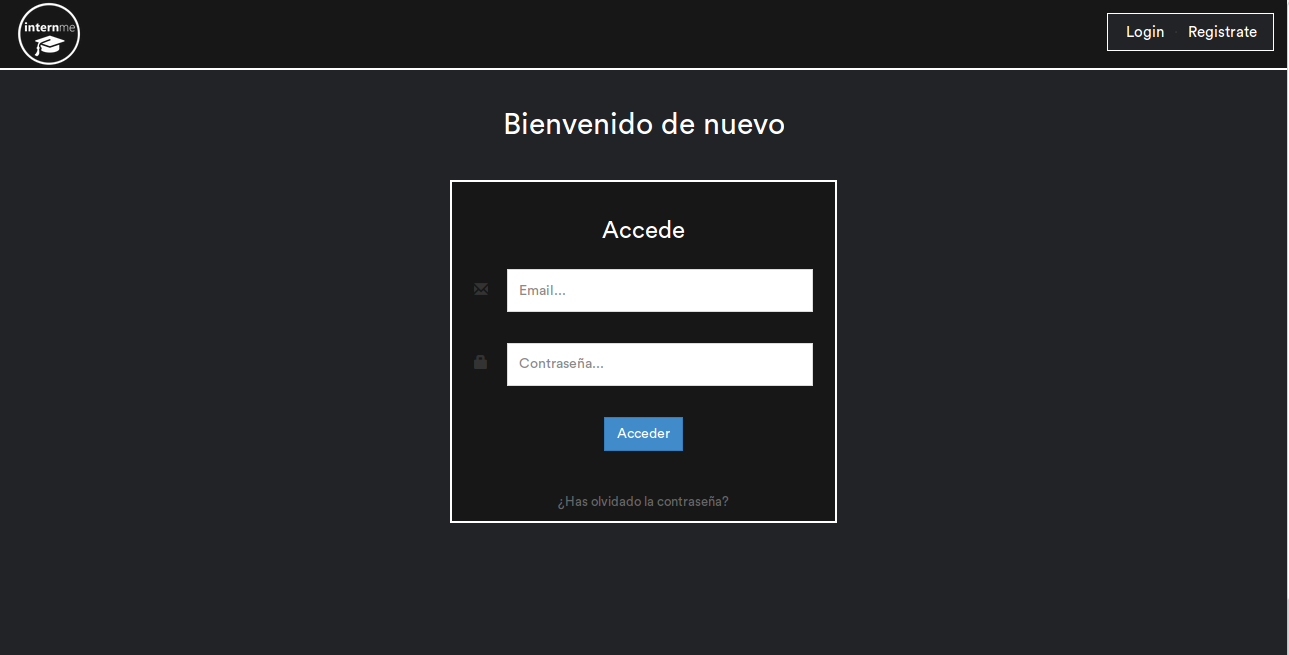


Figura 31 Captura de pantalla de la pantalla de login de la aplicación.

5.6.1.3 Pantalla de recuperar contraseña

A continuación se detallarán las pantallas relativas a la gestión de recuperación de contraseña.



Figura 32 Pantalla de recuperar contraseña 1/2.

Una vez recibido el email con la URL, la vista que se genera para cambiar la contraseña es la siguiente:

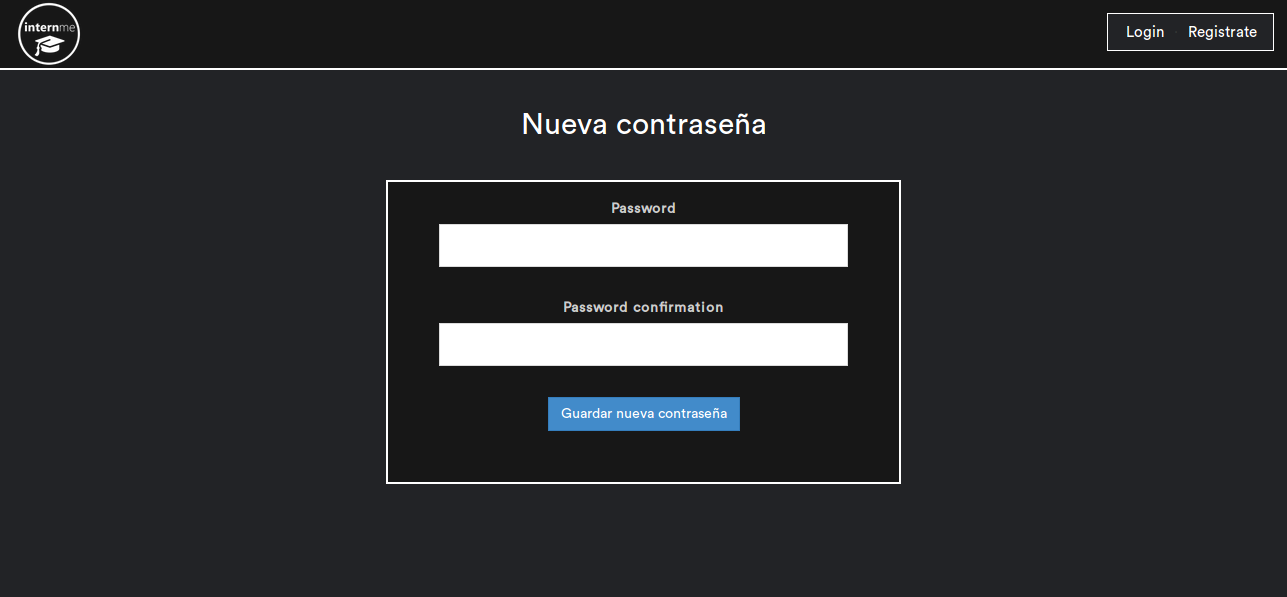


Figura 33Pantalla de recuperar contraseña 2/2.

5.6.1.4 Pantalla de registro

La pantalla principal de registro se ha dividido en dos contenedores con el objetivo de diferenciar el tipo de usuario que se puede registrar en la aplicación web.

****

Figura 34 Captura de pantalla de la pantalla de registro principal de la aplicación

Por un lado tenemos el contenedor izquierdo que hará referencia al formulario de registro de estudiantes y por otro lado tenemos el contenedor derecho que hará referencia al formulario de registro de empresas.

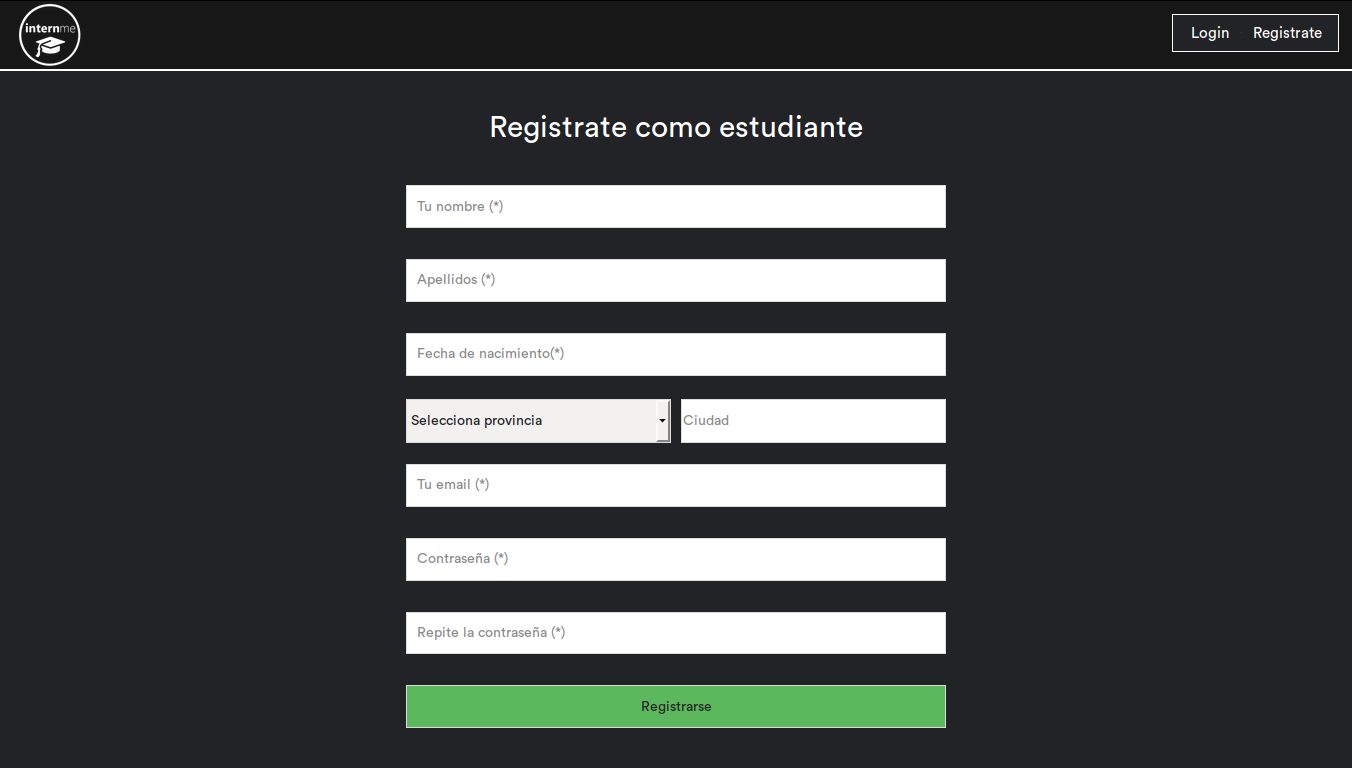


Figura 35 Captura de pantalla de la pantalla de registro de estudiante.

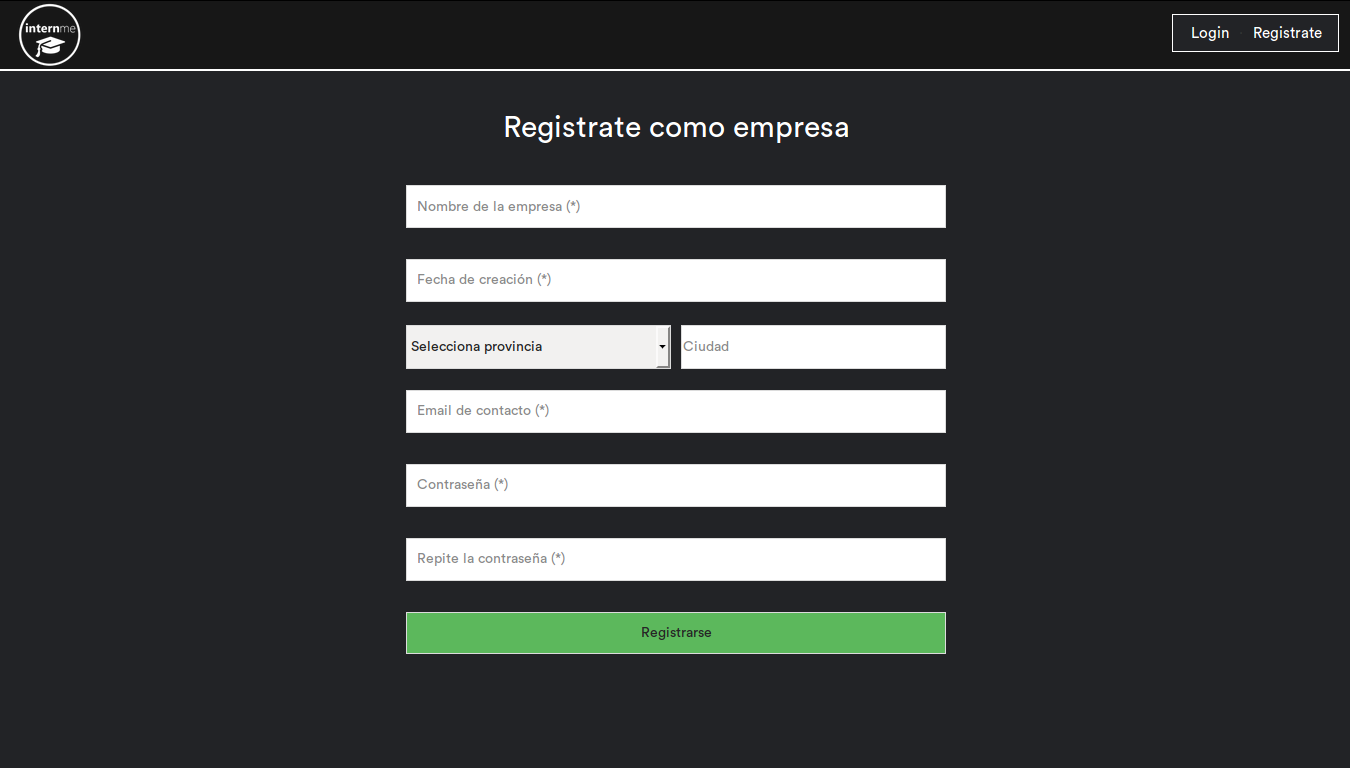


Figura 36 Captura de pantalla de la pantalla de registro de empresa.

5.6.2 Interfaces privados

Una vez vistos los interfaces públicos de la aplicación describiremos aquellos interfaces privados, los cuales solo son visibles por aquellos usuarios que estén registrados en la aplicación y que hayan iniciado sesión.

En este apartado debemos diferenciar aquellos interfaces diseñados para los usuarios con el rol de estudiante y aquellos interfaces diseñados con el rol de empresa, ya que la mayoría de ellos no serán iguales.

5.6.2.1 Interfaces del rol de estudiante

Una de las principales diferencias que encontramos entre los interfaces de estos dos roles es el menú principal de la aplicación, ya que para el rol de estudiante tendrá la opción de acceder a la página principal de empresas y ofertas mientras que para el rol de empresa tendrá la opción de acceder a la página principal de estudiantes y ofertas.

5.6.2.1.1 Pantalla de inicio

En la siguiente imagen podremos ver lo anteriormente expuesto para el rol de usuario, así como el mensaje de confirmación de sesión iniciada que el sistema proporciona cuando el usuario hace login correctamente.

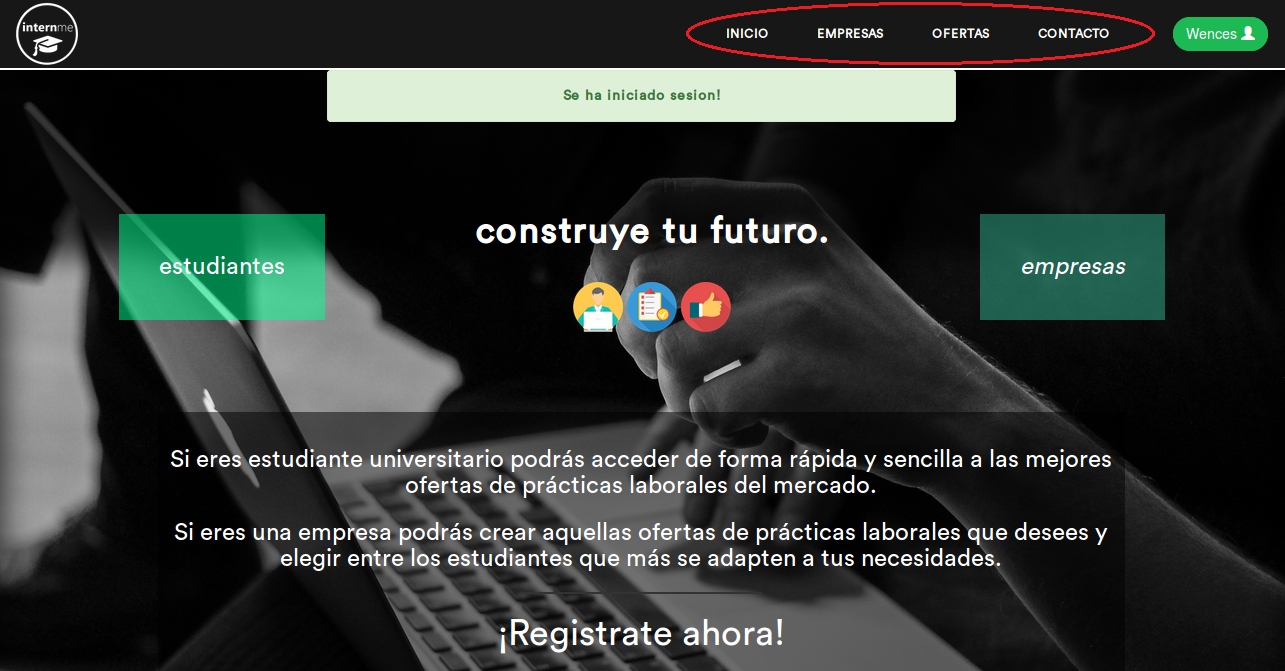


Figura 37 Captura de pantalla de la pantalla de inicio con el rol de estudiante.

5.6.2.1.2 Pantalla de ofertas

El estudiante podrá acceder a la página de ofertas haciendo click en el enlace “Ofertas” ubicado en el menú superior de la página. En ella accederá a todas aquellas ofertas de prácticas de trabajo creadas por las empresas, teniendo la posibilidad de filtrar aquellas ofertas en el buscador por diferentes campos (palabra clave, estudios, provincia o ciudad).



Figura 38 Captura de pantalla de la pantalla de oferta de empresa.

El estudiante podrá ver detalladamente la información completa de cualquier oferta haciendo click en “Ver más” dentro del cuadro de la oferta:



Figura 39 Captura de pantalla de la pantalla de detalle de la oferta seleccionada.

Una vez dentro de la oferta seleccionada, el estudiante puede hacer click en el botón “Inscribirse en la oferta” para inscribirse en esta y participar así en el proceso de selección de esta. A continuación podemos ver en la siguiente imagen que el sistema notifica al usuario por medio de un mensaje que la inscripción se ha realizado correctamente.



Figura 40 Captura de pantalla de la pantalla de detalle de la oferta seleccionada una vez hecha la inscripción.

5.6.2.1.3 Pantalla de empresas y envío de mensajes

Al igual que para las ofertas, el estudiante podrá ver el perfil de todas aquellas empresas registradas en la plataforma haciendo click en el enlace “Empresas” ubicado en el menú superior de la página, así como hacer una búsqueda filtrada (palabra clave, perfil del estudiante buscado, provincia o ciudad) de aquellas empresas que sean de su interés. Por último, el estudiante tendrá la posibilidad de enviar un mensaje a aquellas empresas que desee.

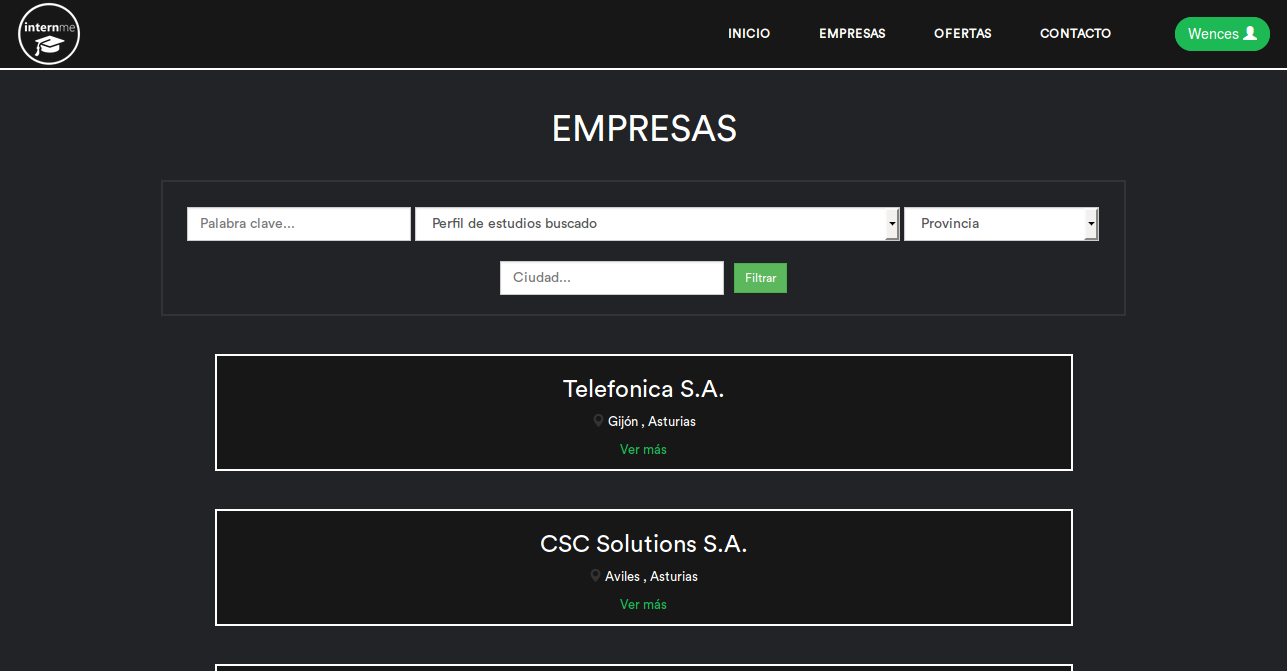


Figura 41 Captura de pantalla de la pantalla principal de empresas con el rol de estudiante.



Figura 42 Captura de pantalla de la pantalla de detalle de la empresa seleccionada.

El diseño que se ha optado para el envío de mensajes ha sido por medio de ventanas modales con peticiones Ajax. Esto se ha hecho con el objetivo de mejorar la experiencia de usuario y hacer la iteración con la página más fluida, ya que con este tipo de peticiones la página no requiere cargarse de nuevo. Cabe destacar que este diseño de interfaces será el mismo tanto para estudiantes como para empresas.

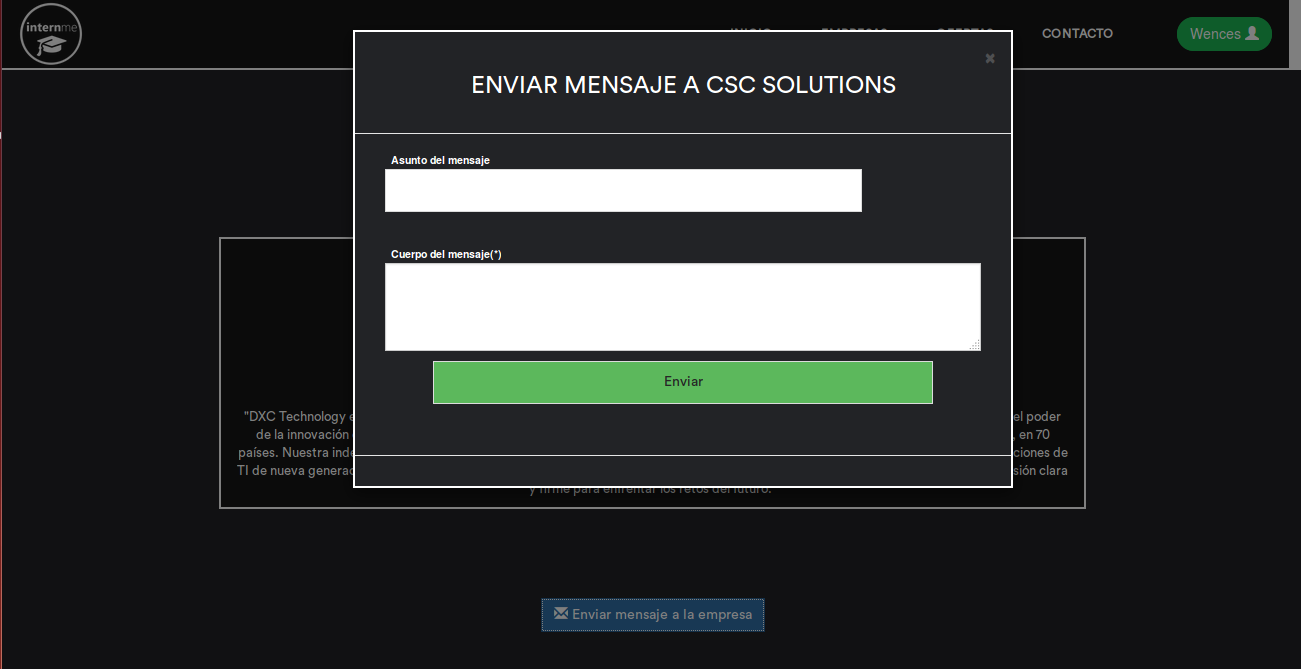


Figura 43 Captura de pantalla de la ventana modal de envío de mensaje.

Una vez enviado el mensaje, el sistema notificará al estudiante del correcto envío del mensaje.

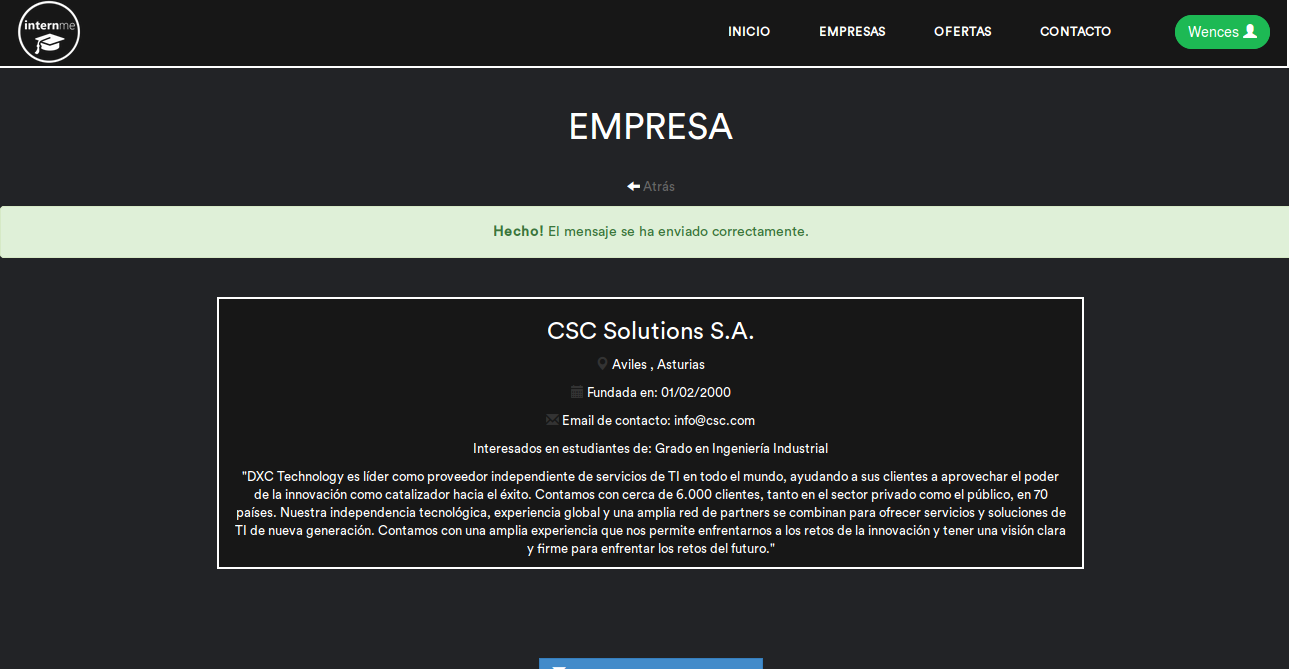


Figura 44 Captura de pantalla de la pantalla de la empresa seleccionada una ve enviado el mensaje.

5.6.2.1.4 Pantalla de perfil

Una de las partes más importantes de la aplicación web es la gestión del perfil del usuario, tanto para estudiantes como para empresas. Se ha decidido hacer un diseño en forma de paneles con sus correspondientes pestañas.

Estos paneles cambiarán a medida que se seleccione una pestaña u otra, permitiendo al usuario navegar entre ellos de forma rápida y sencilla, ya que una vez más la información llevada a los diferentes paneles se ha hecho por medio de peticiones Ajax al servidor, por lo que la página no necesitará recargarse de nuevo cuando cambiemos de pestaña y panel.

5.6.2.1.5 Configuración de cuenta

En este primer panel llamado “Mi perfil” el usuario podrá modificar aquellos datos principales de su cuenta, como su nombre, email, ciudad de origen, edad, contraseña, etc. Se ha utilizado un diseño simple de formulario como los que se han utilizado en el resto de pantallas de la plataforma.



Figura 45 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis Datos”.

5.6.2.1.6 Perfil académico

Este es uno de los paneles más importantes para el estudiante, ya que “Mi CV” es el panel donde el usuario puede definirse profesionalmente e incluir todas aquellas competencias profesionales con las que cuente. Se podría decir que es su curriculum en la aplicación web.

En primer lugar el estudiante podrá seleccionar aquellos estudios universitarios que esté cursando y escribir una breve descripción de sí mismo. A continuación el estudiante podrá poner tantas competencias profesionales como quiera, ya sean otros estudios universitarios realizados con anterioridad, títulos de idiomas, experiencias laborales previas, etc.



Figura 46 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mi CV”.

5.6.2.1.7 Ofertas de usuario

En este tercer panel “Mis ofertas” el estudiante podrá ver un resumen de aquellas ofertas a las que se ha inscrito con anterioridad, así como el estado en el que está cada oferta. Se ha optado por una tabla para representar estos datos.



Figura 47 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis Ofertas”.

5.6.2.1.8 Mensajes

Este último panel “Mis Mensajes” almacenará los mensajes recibidos y enviados por el usuario en la aplicación. El diseño se ha hecho mediante dos tablas, una para los mensajes recibidos, con la posibilidad de leerlos y responderlos así como de ver si ya han sido leídos, y otra tabla para los mensajes que el estudiante ha enviado con anterioridad.



Figura 48 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis Mensajes”.

5.6.2.2 Interfaces del rol de empresa

5.6.2.2.1 Pantalla de inicio

Como mencionamos anteriormente, el rol de empresa tendrá la opción de acceder a la página principal de estudiantes y ofertas en el menú principal.



Figura 49 Captura de pantalla de la pantalla de inicio con el rol de empresa.

5.6.2.2.2 Pantalla de estudiantes y envío de mensajes

La empresa ver el perfil de todos aquellos estudiantes registrados en la plataforma haciendo click en el enlace “Estudiantes” ubicado en el menú superior de la página, así como hacer una búsqueda filtrada (palabra clave, universidad, estudios, provincia o ciudad) de aquellos estudiantes que sean de su interés.

Por último, la empresa (al igual que el estudiante como ya hemos visto) tendrá la posibilidad de enviar un mensaje a aquellos estudiantes que desee.

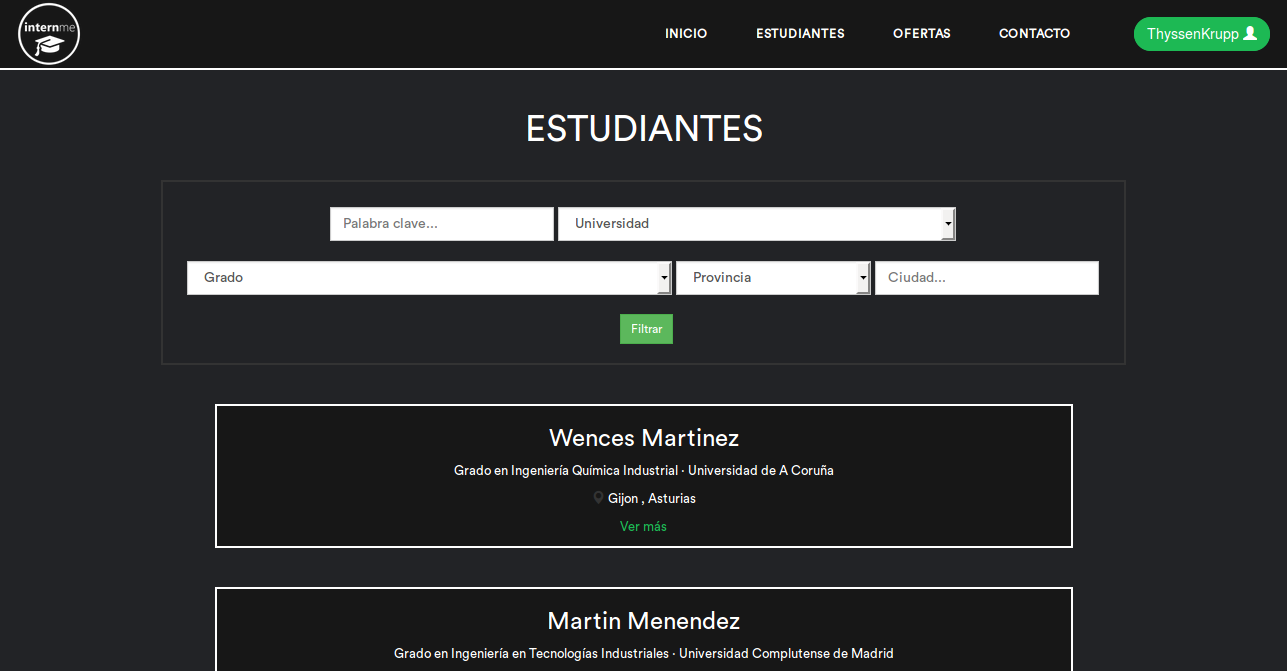


Figura 50 Captura de pantalla de la pantalla principal de estudiantes con el rol de empresa.

La empresa podrá ver detalladamente la información completa de cualquier estudiante haciendo click en “Ver más” dentro del cuadro de este, así como también enviarle un mensaje haciendo click en “Enviar mensaje al estudiante”:



Figura 51 Captura de pantalla de la pantalla de detalle del estudiante seleccionado.

5.6.2.2.3 Pantalla de ofertas

El diseño y funcionamiento de la pantalla de ofertas es exactamente igual tanto para empresas como para estudiantes, por tanto no se pondrán las respectivas interfaces dado que son las mismas que las del apartado XXX del rol de estudiante.

5.6.2.2.4 Pantalla de perfil

Al igual que para el estudiante el apartado “Mi perfil” es uno de los apartados más importantes dentro de la aplicación web para la empresa, ya que desde él podrá gestionar todos los procesos de selección resultantes de las ofertas de prácticas de trabajo que haya creado así como gestionar el envío de mensajes y detallar que perfil de estudiantes busca, así como establecer una breve descripción de la empresa.

5.6.2.2.5 Pantalla de configuración de cuenta

En este primer panel llamado “Mis Datos” el usuario podrá modificar aquellos datos principales de su cuenta, como su nombre, email, ciudad de origen, edad, contraseña, etc. En este caso, la empresa también podrá establecer aquel perfil de estudiante en el que esté interesado y también establecer una breve descripción de la empresa. Se ha utilizado un diseño simple de formulario como los que se han utilizado en el resto de pantallas de la plataforma.



Figura 52 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis datos”.

5.6.2.2.6 Pantalla de ofertas

Al igual que para el estudiante, la podrá ver un resumen de aquellas ofertas que haya creado, pudiendo ver el estado en el que está cada oferta.

Por otra parte y en este mismo apartado, la empresa crear ofertas de prácticas de trabajo haciendo click en el botón “Crear nueva oferta”, lo que cargará un formulario para que la empresa cree una nueva oferta que se publicará en la aplicación una vez rellene los datos requeridos y envíe el formulario.

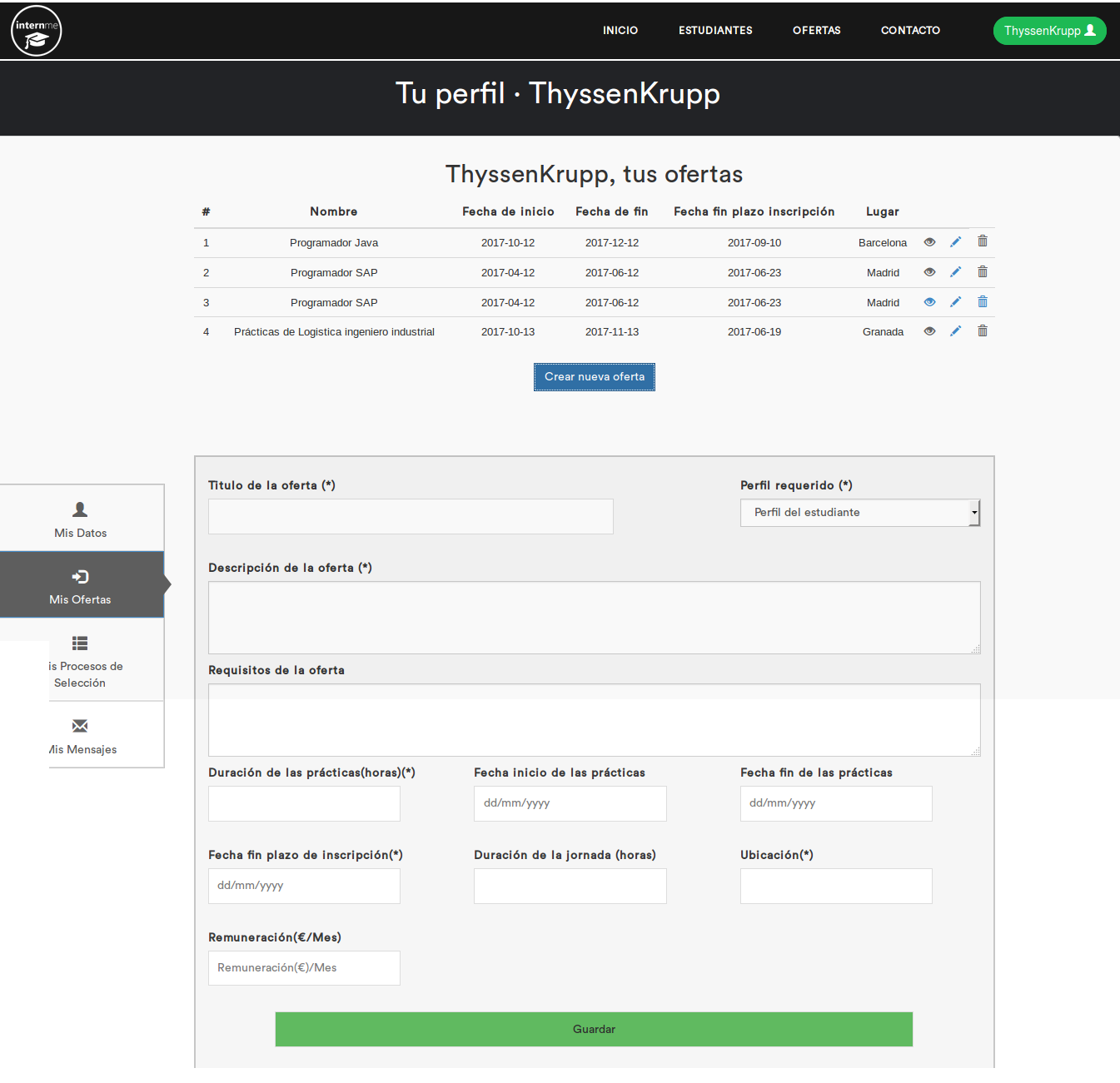


Figura 53 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis ofertas” con el formulario de crear nueva oferta abierto.

5.6.2.2.7 Pantalla de procesos de selección

Este apartado llamado “Procesos de selección” englobará todos aquellos procesos de selección derivados de las ofertas de prácticas que la empresa ha creado y debe gestionar. Como diseño del panel se ha optado por incluir una tabla con todos aquellos procesos de selección y sus diferentes estados, incluyendo la posibilidad de gestionar las respectivas fases de cada proceso de selección.

Cada vez que se haga click en “Gestionar” en alguno de los procesos de selección se nos cargará más abajo un panel que contendrá aquellos estudiantes que estén en la fase actual del proceso de selección, con el objetivo de que la empresa pueda gestionar esa fase del proceso decidiendo entre ellos y pudiendo ver el perfil de cada estudiante en todo momento a través de una ventana modal.

A continuación se muestra un ejemplo de gestión de un proceso de selección con un estudiante inscrito en la oferta de prácticas.

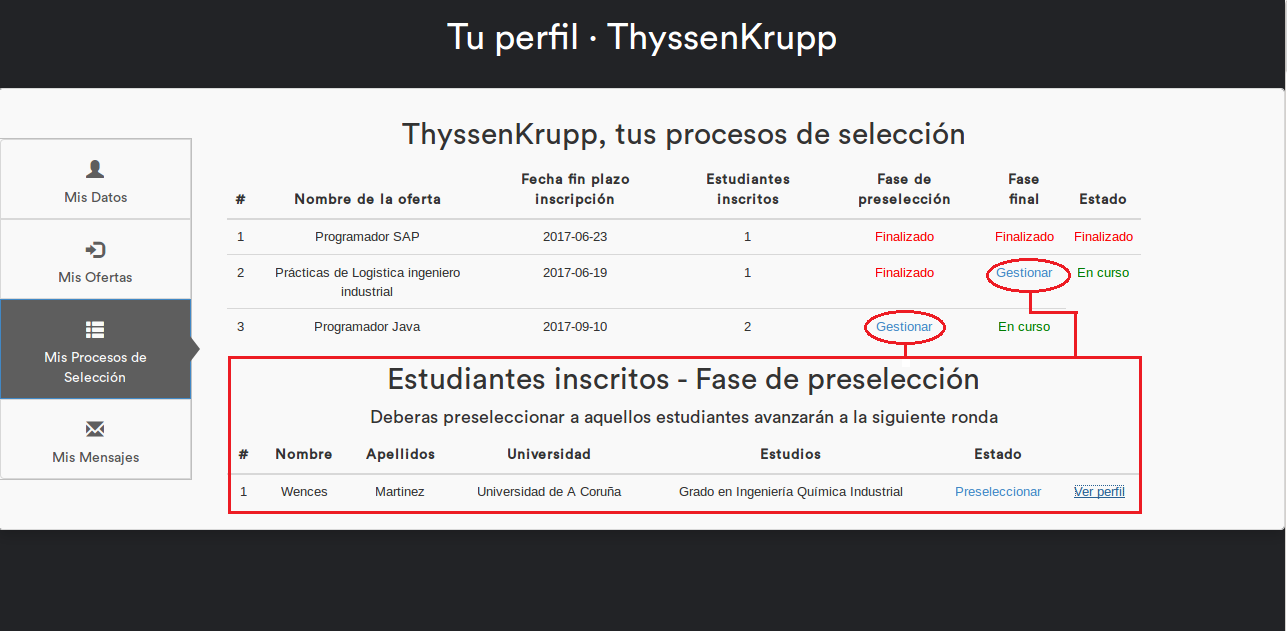


Figura 54 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis procesos de selección” con el contenedor de la fase de preselección de una oferta abierto.

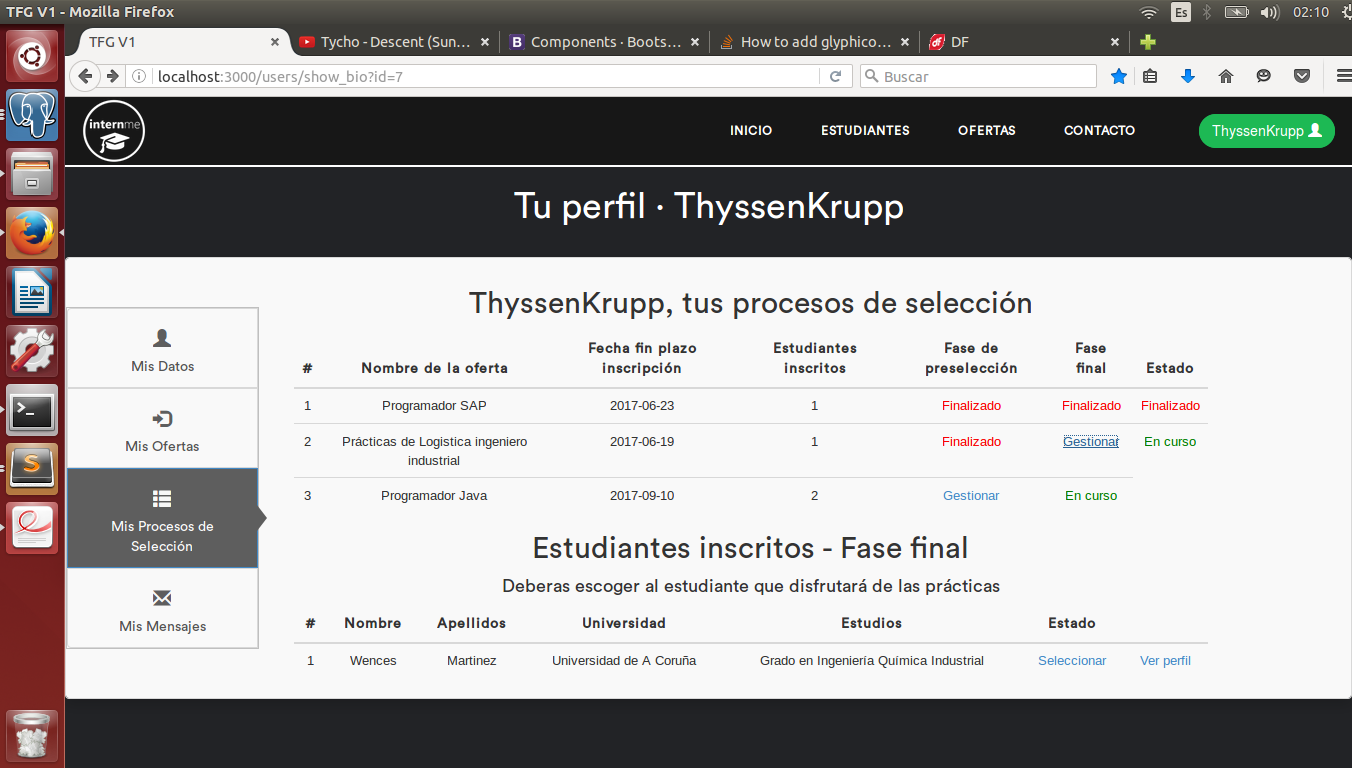


Figura 55 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis procesos de selección” con el contenedor de la fase final de selección de una oferta abierto.

Como podemos ver en la siguiente imagen, la empresa también podrá ver el perfil de aquellos usuarios pertenecientes a las fases del proceso de selección. El diseño de estas interfaces ha seguido una línea prácticamente idéntica a la del envío de mensajes, ya que se ha hecho mediante ventanas modales, lo que hará más fácil y rápida la visualización de los perfiles de los estudiantes, ya que no se cambiará de página en ningún momento.



Figura 56 Captura de pantalla de la pantalla principal de mi perfil en la pestaña “Mis procesos de selección” con la ventana modal de perfil del estudiante seleccionado abierta.

5.6.2.2.8 Pantalla de mensajes

Al igual que para los estudiantes, la pestaña de “Mis Mensajes” está diseñada exactamente igual que como hemos visto anteriormente en el apartado XX para los usuarios con rol de estudiante.

# DOCUMENTO 6: PRUEBAS

**D. MARTÍNEZ SUÁREZ, Wenceslao**

**TUTOR: Dña. SUAREZ CABAL, María José**

**FECHA: Julio 2017**

Índice de contenidos

[**DOCUMENTO 6: PRUEBAS** 120](#_Toc486815244)

[6.1 INTRODUCCIÓN 122](#_Toc486815245)

[6.2 PRUEBAS UNITARIAS 122](#_Toc486815246)

[6.3 PRUEBAS DE INTEGRACIÓN Y DE SISTEMA 124](#_Toc486815247)

[6.4 PRUEBAS DE USABILIDAD Y ACCESIBILIDAD 130](#_Toc486815248)

[6.4.1 CRITERIOS GENERALES 130](#_Toc486815249)

[6.4.2 CRITERIOS DE IDENTIDAD E INFORMACIÓN 131](#_Toc486815250)

[6.4.3 CRITERIOS DE LENGUAJE Y REDACCIÓN 132](#_Toc486815251)

[6.4.4 CRITERIOS DE ROTULADO 133](#_Toc486815252)

[6.4.5 CRITERIOS DE NAVEGACIÓN 133](#_Toc486815253)

[6.4.6 LAY-OUT DE LA PÁGINA 134](#_Toc486815254)

[6.4.7 CRITERIOS DE BÚSQUEDA 135](#_Toc486815255)

[6.4.8 CRITERIOS DE AYUDA 135](#_Toc486815256)

[6.4.9 ACCESIBILIDAD 136](#_Toc486815257)

[6.4.10 CRITERIOS DE CONTROL Y RETROALIMENTACIÓN 137](#_Toc486815258)

[6.5 PRUEBAS DE RENDIMIENTO 137](#_Toc486815259)

## 6.1 Introducción

En este apartado abordaremos todas aquellas pruebas realizadas durante el desarrollo de este proyecto, las cuales han garantizado que el sistema funcione de la mejor manera posible ante las diversas situaciones que se pueden dar, respondiendo tal y como se espera.

El desarrollo de las pruebas se ha dividido en cuatro apartados principales: pruebas unitarias, pruebas de integración y sistema, pruebas de rendimiento y por último pruebas de usabilidad y accesibilidad.

6.2 Pruebas unitarias

Dentro de un sistema con un diseño estructurado, las pruebas unitarias son aquellas pruebas realizadas para las diferentes funcionalidades de cada unidad del sistema, garantizando que estas funcionen correcta y eficientemente por separado.

Para abordar el desarrollo de este tipo de pruebas utilizaremos el TDD o Test Driven Development, disciplina de desarrollo de software en la cual los desarrolladores escriben multitud de casos de prueba para los requerimientos funcionales de una aplicación. El TDD tiene como objetivo garantizar un código limpio, simple y testeable, asegurando la integridad de las diferentes funcionalidades de la aplicación. Una de las mejoras formas de aplicar el TDD a nuestra aplicación en Ruby on Rails es haciendo uso de la gema Rspec, librería creada para realizar tests unitarios para los modelos y test funcionales para los controladores.

Una vez instalada la librería en el proyecto, se debe hacer una configuración inicial de esta para llevar a cabo las pruebas. Se creará una base de datos de pruebas que seguirá la misma estructura que tenemos en la base de datos principal de la aplicación, con las mismas tablas. El hecho de que la base de datos de desarrollo y de pruebas sean iguales nos garantizará que las pruebas llevadas a cabo se asemejarán a la configuración real que tenemos en nuestro proyecto.

Una de las ventajas que tiene Ruby on Rails relacionadas con las pruebas es que esta funcionalidad ya la tiene integrada en el sistema, ya que siempre que se crea un proyecto nuevo el framework incluye una carpeta llamada “test” la cual almacenará todas aquellas pruebas que el desarrollar realice. En la carpeta “test” habrá otra carpeta llamada “models” destinada a albergar las pruebas para los modelos de la aplicación y otra carpeta llamada “controllers” donde se establecerán aquellas pruebas para los controladores.

En el caso de nuestro proyecto nos interesa hacer pruebas de modelos las cuales nos servirán para corregir todos aquellos fallos relacionados con la creación de estos y la modificación de su información. A continuación se detallará un ejemplo de prueba para el modelo “offer”.

El código que se puede ver a continuación se corresponde con un extracto de nuestro modelo “offer”, donde podremos ver que contiene los campos *name, description, perfil\_id, duration, location y finish\_inscription\_date* con validación de presencia activada, lo que quiere decir que nunca se creará un objeto Offer si alguno de estos campos llega vacío.

**class** **Offer** **<** ApplicationRecord

#Libreria de búsqueda PgSearch

include PgSearch

pg\_search\_scope :search\_by\_description**,** :against **=>** **[**:name**,** :description**,** :location**],** using: **{** tsearch: **{** any\_word: **true** **}** **}**

#Relaciones

belongs\_to :user

has\_many :offer\_inscription**,** dependent: :destroy

# Restricciones de validación

validates :name**,** :description**,** :perfil\_id**,** :duration**,** :location**,** :finish\_inscription\_date**,** presence: **true**

validates :duration**,** :numericality **=>** **{**:only\_integer **=>** **true}**

validates :hours**,** :numericality **=>** **{**:only\_integer **=>** **true}**

validates :salary**,** :numericality **=>** **{**:only\_integer **=>** **true}**

validates\_format\_of :start\_date**,** :with **=>** /\d{2}\/\d{2}\/\d{4}/

validates\_format\_of :finish\_date**,** :with **=>** /\d{2}\/\d{2}\/\d{4}/

validates\_format\_of :finish\_inscription\_date**,** :with **=>** /\d{2}\/\d{2}\/\d{4}/

Una vez visto los diferentes campos y sus validaciones podemos crear nuestra prueba para los campos con presencia requerida del modelo Offer

describe Offer do

it "is valid with a name, description, perfil\_id,

duration, location and finish\_inscription\_date" do

offer **=** Offer.new**(**

name**:** 'Practicas de PHP',

description**:** 'Oferta de prácticas de PHP',

perfil\_id**:** '3',

duration**:** '300',

finish\_inscription\_date**:** '20/5/2017'**)**

expect**(**offer**)**.to be\_valid

end

it "is invalid without a description" do

offer **=** Offer.new**(**description**:** nil**)**

offer.valid**?**

expect**(**offer.errors**[:**firstname**])**.to include**(**"can't be blank"**)**

end

end

En este ejemplo de prueba lo que hemos hecho ha sido crear un nuevo objeto Offer con datos de ejemplo. Seguido de esto le especificamos el comportamiento que esperamos que tenga con el método “expect” pasándole nuestro objeto Offer recién creado y especificándole que queremos que sea válido con “be\_valid”, ya que lo que nos interesa es verificar que la validación de presencia “presence: true” se cumple.

Una vez construida la prueba podremos ejecutarla y observar si ha ido bien o no, observando aquellos fallos que tengamos en el caso de que la ejecución de la prueba no haya sido exitosa.

Al igual que acabamos de ver, para este proyecto se han hecho diferentes pruebas de esta índole en los modelos más importantes de la aplicación como User, Offer\_Inscription o Message.

6.3 Pruebas de integración y de sistema

En este apartado veremos las pruebas de integración y de sistema, lo que nos permitirá probar la funcionalidad de todo el sistema en conjunto. Gracias a este tipo de pruebas se puede comprobar como es la iteración entre dos o más unidades del sistema, verificando así que los componentes de la aplicación funcionan correctamente en conjunto.

El desarrollo de estas pruebas se ha hecho sobre los casos de uso más importantes de la aplicación:

Tabla 12 Pruebas de integración y sistema

|  |  |
| --- | --- |
| **Caso de uso – Registro de usuario** | |
| **Prueba** | **Resultado Esperado** |
| Añadir un usuario nuevo (no existente) | El sistema registra al nuevo usuario correctamente. |
|  | **Resultado obtenido** |
|  | El sistema registra al usuario correctamente. |
| **Prueba** | **Resultado esperado** |
| Añadir un usuario ya existente en el sistema (mismo email) | El sistema no registra al usuario y se lo notifica. |
|  | **Resultado obtenido** |
|  | Se cumple el resultado esperado, el sistema no registra al usuario y se recibe una notificación. |
| **Prueba** | **Resultado esperado** |
| Añadir un usuario campos requeridos vacíos. | El sistema no registra al usuario y notifica que campos faltan por rellenar. |
|  | **Resultado obtenido** |
|  | Tal y como se esperaba el sistema no añade al usuario y le notifica los fallos. |
| **Caso de uso – Login de usuario** | |
| **Prueba** | **Resultado esperado** |
| Intento de login con un usuario existente en el sistema | El usuario se logea correctamente en el sistema. |
|  | **Resultado obtenido** |
|  | El usuario se logea correctamente en el sistema siempre que introduzca la contraseña asociada a su cuenta, si no, el sistema informará de que las credenciales con las que el usuario intenta acceder son erróneas. |
| **Prueba** | **Resultado esperado** |
| Intento de acceso a una sección no autorizada por el usuario | El usuario es redirigido a la pantalla de inicio y se le notifica que no tiene permisos para acceder a esa sección. |
|  | **Resultado obtenido** |
|  | Efectivamente no se le permite el acceso al usuario y se le notifica que no tiene permisos. |
| **Caso de uso – Recuperar contraseña** | |
| **Prueba** | **Resultado esperado** |
| Recuperación de contraseña usuario registrado. | El sistema permite al usuario cambiar la contraseña. |
|  | **Resultado obtenido** |
|  | Efectivamente el usuario puede cambiar la contraseña de acceso. |
| **Prueba** | **Resultado esperado** |
| Recuperación de contraseña usuario no registrado. | El sistema no envía el email con la URL de modificación de la contraseña debido a que el email introducido por el usuario no está registrado. |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema no envía el email al usuario con la URL y no permite la recuperación de contraseña. |
| **Caso de uso – Modificar información básica de cuenta de usuario** | |
| **Prueba** | **Resultado esperado** |
| Se edita la información del usuario. | El sistema actualiza al usuario en base a la nueva información introducida. |
|  | **Resultado obtenido** |
|  | Efectivamente se actualiza la información de usuario. |
| **Caso de uso – Configurar perfil profesional de usuario** | |
| **Prueba** | **Resultado esperado** |
| Un usuario estudiante añade o edita aquellas competencias profesionales que considere oportunas. | El sistema guarda o actualiza las competencias de forma correcta. |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema guarda o actualiza las competencias. |
| **Prueba** | **Resultado esperado** |
| El usuario estudiante borra aquellas competencias profesionales que considere oportunas. | El sistema borrará aquellas competencias que el usuario haya seleccionado. |
|  | **Resultado obtenido** |
|  | Las competencias se borran correctamente. |
| **Caso de uso – Creación de ofertas** | |
| **Prueba** | **Resultado esperado** |
| Usuario empresa crea una oferta de prácticas | El sistema crea la oferta correctamente |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema crea la oferta sin problemas. |
| **Prueba** | **Resultado esperado** |
| Usuario empresa crea una oferta de prácticas con campos requeridos vacíos | El sistema no permite crear la oferta y le notifica al usuario que campos debe rellenar |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema no permite crear la oferta y notificará al usuario al respecto. |
| **Prueba** | **Resultado esperado** |
| El usuario empresa elimina una oferta de prácticas que haya creado anteriormente. | El sistema elimina la oferta y notificará de esto tanto al usuario empresa como a los usuarios estudiantes que se hayan inscrito en la oferta o estén participando en algunas de las fases del proceso de selección de esta. La notificación a los usuarios estudiantes se hará por medio de mensajes automáticos que mandará el sistema. |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema borra la oferta y notifica tanto a la empresa como a los estudiantes, a estos últimos por medio de un mensaje. |
| **Caso de uso – Inscripción en ofertas** | |
| **Prueba** | **Resultado esperado** |
| El usuario estudiante se inscribe en una oferta de prácticas. | El sistema inscribe al estudiante en la oferta |
|  | **Resultado obtenido** |
|  | Efectivamente el estudiante se escribe en la oferta y se lo notifica |
| **Prueba** | **Resultado esperado** |
| El usuario estudiante se des inscribe de una oferta en la que ya estaba escrito. | El sistema des inscribe al estudiante de la oferta y se lo notifica. |
|  | **Resultado obtenido** |
|  | Efecitvamente el sistema des inscribe al estudiante de la oferta y se lo notifica. |
| **Caso de uso – Proceso de selección** | |
| **Prueba** | **Resultado esperado** |
| Al crear una oferta se crea automáticamente un proceso de selección para esta | El sistema creará un proceso de selección asociado a la oferta de prácticas que se acaba de crear. |
|  | **Resultado obtenido** |
|  | Efectivamente se crea un proceso de selección cuando se crea una nueva oferta. |
| **Prueba** | **Resultado esperado** |
| El usuario empresa elige a aquellos estudiantes de la primera fase del proceso de selección (fase de preselección) para avanzar a la fase final | El sistema avanza de fase a aquellos estudiantes seleccionados por la empresa en la fase de preselección y se lo notifica a estos por medio de un mensaje. El sistema también notificará a aquellos estudiantes no seleccionados por medio de mensajes cuando finalice el plazo de la fase de preselección. |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema avanza de fase a aquellos estudiantes seleccionados y se lo notifica por medio de mensajes. También notifica a los estudiantes no seleccionados. |
| **Prueba** | **Resultado esperado** |
| El usuario empresa elige a aquel estudiante de la fase final del proceso de selección (fase de preselección) que será el ganador de las prácticas. | El sistema selecciona al estudiante ganador y se lo notifica por medio de un mensaje. El sistema también notificará a aquellos estudiantes no seleccionados por medio de mensajes cuando finalice el plazo de la fase final. |
|  | **Resultado obtenido** |
|  | Efectivamente el sistema selecciona al usuario ganador de las prácticas y se lo notifica por medio de un mensaje. También notifica a los estudiantes no seleccionados. |
| **Caso de uso – Envío de mensajes** | |
| **Prueba** | **Resultado esperado** |
| El usuario envía un mensaje a otro. | El mensaje se envía correctamente. |
|  | **Resultado obtenido** |
|  | Efectivamente el mensaje se envía correctamente. |
| **Prueba** | **Resultado esperado** |
| Envío de mensaje con los campos requeridos vacíos. | El sistema no envía el mensaje y se lo notifica al usuario. |
|  | **Resultado obtenido** |
|  | Efectivamente el mensaje no se envía y se notifica al usuario los campos que debe rellenar obligatoriamente. |
| **Caso de uso – Gestión de mensajes** | |
| **Prueba** | **Resultado esperado** |
| El usuario responde a un mensaje | El sistema envía el mensaje de respuesta |
|  | **Resultado obtenido** |
|  | Efectivamente el mensaje se envía correctamente. |

6.4 Pruebas de usabilidad y accesibilidad

A continuación se detallará la evaluación que se ha seguido para los aspectos de usabilidad y accesibilidad de la aplicación web. Se ha decidido seguir una guía heurística basada en los siguientes criterios:

* Generales.
* Identidad e información.
* Lenguaje e información.
* Rotulado.
* Navegación.
* Lay-out de la página.
* Búsqueda.
* Ayuda.
* Accesibilidad.
* Control y retroalimentación.

6.4.1 Criterios generales

Tabla 13 Pruebas de usabilidad y accesibilidad - Criterios generales

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?** Al ser una plataforma pensada para el ámbito profesional esta se ha desarrollado siguiendo unos objetivos claros y concisos, ofreciendo servicios únicamente relacionados con los objetivos previamente establecidos. | **Sí** |
| **¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?** La aplicación no se ha subido a un servidor online, por lo tanto no se ha podido establecer una URL principal, pero las url’s de sus páginas internas son intuitivas, cortas y concisas, siendo fácilmente recordables por el usuario. | **Sí** |
| **¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web?** La pantalla principal contiene una información breve y concisa de los objetivos principales de la aplicación para los diferentes tipos de usuarios que la utilizarán. | **Sí** |
| **¿La estructura general del sitio web está orientada al usuario?** Dado que la aplicación está pensada para que la utilicen estudiantes y empresas, esta se ha diseñado con una estructura totalmente pensada y orientada al usuario. | **Sí** |
| **¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web?** Dado que se trata de una aplicación de ámbito profesional esta se ha diseño con colores oscuros en su mayoría a la vez que claros, intentando transmitir seriedad y evitando colores llamativos y fuera de lugar que puedan confundir al usuario. | **Sí** |
| **¿Es coherente el diseño general del sitio web?** Si lo es, ya que todas las páginas internas de la aplicación siguen una línea de diseño idéntica, incluyendo elementos limpios y con una estructura similar. | **Sí** |
| **¿Es reconocible el diseño general del sitio web?** Dado que el diseño de la aplicación web se ha hecho en su mayoría con colores oscuros, esto proporcionará una clara diferenciación frente a muchas páginas web del mercado, haciendo que el usuario pueda reconocer su diseño más fácilmente. | **Sí** |
| **¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza?** Dado que esta aplicación web ha sido fruto de la realización de un proyecto puramente académico no se ha contemplado el desarrollo de futuras actualizaciones, excluyendo así cualquier tipo de indicación a cerca de estas. | **No** |

6.4.2 Criterios de identidad e información

Tabla 14Pruebas de usabilidad y accesibilidad - Criterios de identificación e información

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?** Sí, dado que el logo de la aplicación web siempre se mostrará en el menú principal de navegación ubicado en la parte superior de la página. | **Sí** |
| **El Logotipo, ¿es significativo, identificable y suficientemente visible?** El logotipo cuenta con la palabra “intern” que significa becario o alumno en prácticas y la palabra “me” que se refiere a uno mismo, lo cual manda un mensaje significativo a cerca de uno de los principales objetivos de la aplicación, que los estudiantes puedan encontrar prácticas de trabajo. A parte de esto, el logotipo cuenta con una buena calidad, por tanto se puede ser sin problemas en el menú superior de navegación. | **Sí** |
| **El eslogan o tagline, ¿expresa realmente qué es la empresa y qué servicios ofrece?** Dado el slogan pensado para la aplicación “construye tu futuro” podemos afirmar que expresa uno de los principales servicios de la aplicación, y es que los estudiantes vayan construyendo su futuro (laboral) por medio de la realización de prácticas. | **Sí** |
| **¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster',...?** Como se ha mencionado anteriormente, dado que esta aplicación web ha sido fruto de la realización de un proyecto puramente académico no se ha contemplado la opción de crear algún enlace con información sobre la empresa o sitio web. | **No** |

6.4.3 Criterios de lenguaje y redacción

Tabla 15 Pruebas de usabilidad y accesibilidad - Criterios de lenguaje y redacción

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿El sitio web habla el mismo lenguaje que sus usuarios?** Sí, dado que es una plataforma web pensada para tener estudiantes y empresas del territorio español. Establecer la plataforma en otros idiomas es una posible ampliación que no se debería descartar. | **Sí** |
| **¿Emplea un lenguaje claro y conciso?** Sí, dado que las palabras, oraciones o cualquier otro texto incluido en la aplicación se ha escrito de forma concisa con el fin de explique en el menor número de palabras el mensaje que quiere transmitir | **Sí** |
| **¿Es amigable, familiar y cercano?** Sí, dado que se el usuario tiene muchas interacciones con el sistema, se ha optado por redactar aquellos mensajes o notificaciones que le lleguen al usuario de una forma correcta pero a la vez amigable y cercana. | **Sí** |

6.4.4 Criterios de rotulado

Tabla 16 Pruebas de usabilidad y accesibilidad - Criterios de rotulado

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **Los rótulos, ¿son significativos?** Respecto a los rótulos de los botones o enlaces si se consideran significativos, ya que se han escrito de tal forma que el usuario vea de forma clara y sencilla la acción a realizar en cada uno de ellos. | **Sí** |
| **¿Usa rótulos estándar?** Sí, ya que para el usuario será más fácil navegar por la aplicación web si conoce de ver en otras webs la mayoría de los rótulos. | **Sí** |
| **El título de las páginas, ¿Es correcto? ¿Ha sido planificado?** Sí, cada página o pestaña conllevan un título que describe en su totalidad el contenido de esta. | **Sí** |

6.4.5 Criterios de navegación

Tabla 17 Pruebas de usabilidad y accesibilidad - Criterios de navegación

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿Los enlaces son fácilmente reconocibles como tales? ¿su caracterización indica su estado (visitados, activos,...)?** Sí, ya que se ha optado por un diseño distinto a la hora de incluir enlaces en la aplicación (distinto color, subrayado, etc). | **Sí** |
| **En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística?** Sí, dado que el menú principal de la aplicación solo tiene 3 elementos para cada rol de usuario, evitando así cualquier tipo de sobrecarga con muchos subapartados. | **Sí** |
| **¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace?** Sí, ya que el enlace expresa la acción a realizar, por lo que el usuario puede hacerse una idea lo que la aplicación le va a mostrar. | **Sí** |
| **¿Se ha controlado que no haya enlaces que no llevan a ningún sitio?** Sí, todos aquellos elementos que se consideran enlaces han sido probados para que cumplan su función y evitar así enlaces que no funcionen. | **Sí** |
| **¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación?** A pesar de que no se han incluido migas de pan en la aplicación lo que si se ha incluido han sido títulos claros en todas las páginas para que el usuario sepa en todo momento donde se encuentra. | **Sí** |
| **¿Se ha evitado la redundancia de enlaces?** Sí, dado que en las diferentes páginas de la aplicación no se repiten enlaces con la misma funcionalidad. | **Sí** |
| **¿Se ha controlado que no haya páginas "huérfanas"?** Sí, todas las páginas incluidas en la aplicación se han revisado para asegurar que no haya páginas huérfanas accesibles por los usuarios. | **Sí** |

6.4.6 Lay-out de la página

Tabla 18 Pruebas de usabilidad y accesibilidad - Criterios de layout de la página

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia? (como por ejemplo la zona central de la página)** Sí, dado que el contenido principal se ha estructurado de tal forma que siempre quede en la zona central de las páginas. | **Sí** |
| **¿Se ha evitado la sobrecarga informativa?** Sí, como se ha explicado antes, cualquier información relativa a los contenidos de la página web se ha minimizado de tal forma que sea breve y clara. | **Sí** |
| **¿Es una interfaz limpia, sin ruido visual?** Sí, dado que no se incluyen más elementos que los requeridos, es decir, no hay imágenes u otros elementos extras sin importancia que distraigan la atención del usuario y que sobrecarguen la página. | **Sí** |
| **¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?** Sí, se ha hecho un diseño de interfaz que incluya márgenes entre los diferentes elementos de tal forma que la información no esté aglutinada y sea más fácil de leer por el usuario. | **Sí** |
| **¿Se ha controlado la longitud de página?** Sí, se ha intentado en todo momento desarrollar las diferentes páginas de la aplicación web con el objetivo de incluir toda la información más importante en el menor espacio posible de cara a que el usuario no tenga que interactuar con páginas de mucha longitud. | **Sí** |

6.4.7 Criterios de búsqueda

Tabla 19 Pruebas de usabilidad y accesibilidad - Criterios de búsqueda

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿La búsqueda se encuentra fácilmente accesible?** Los apartados de búsqueda se corresponden con las búsquedas de estudiantes, empresas y ofertas, los cuales están ubicados al principio de la página correspondiente. | **Sí** |
| **¿Es fácilmente reconocible como tal?** Sí, ya que en los diferentes inputs de la búsqueda se indica en todo momento que se trata de un buscador. | **Sí** |
| **¿Permite la búsqueda avanzada?** Sí, en el sentido de que a la búsqueda se le puede aplicar un filtrado por varios campos. | **Sí** |
| **¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?** Sí | **Sí** |
| **¿La caja de texto es lo suficientemente ancha?** Sí, ya que todos los inputs de texto son lo suficientemente anchos como para albergar más de 7 u 8 palabras. | **Sí** |
| **¿Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada?** Sí, el sistema notificará al usuario cuando no se encuentren resultados para la búsqueda realizada. | **Sí** |

6.4.8 Criterios de ayuda

Tabla 20 Pruebas de usabilidad y accesibilidad - Criterios de ayuda

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **Si posee una sección de Ayuda, ¿Es verdaderamente necesaria?** Sí, ya que al ser una aplicación web que requiere un alto grado de interacción entre el usuario y esta, será necesario explicar el funcionamiento de las funcionalidades a los distintos usuarios | **Sí** |
| **En enlace a la sección de Ayuda, ¿Está colocado en una zona visible y "estándar"?** Sí, ya que está colocado en el menú superior de la aplicación, el cual es visible en todo momento. | **Sí** |

6.4.9 Accesibilidad

Tabla 21 Pruebas de usabilidad y accesibilidad - Criterios de accesibilidad

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?** Si, la fuenta y el tamaño de esta se han escogido de forma que el usuario no tenga problemas de legibilidad. | **Sí** |
| **¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleados facilitan la lectura?** Sí, ya que se utiliza un ancho de línea que facilitará la lectura al usuario, así como el resultado de la tipografía en las frases u palabras de importancia. | **Sí** |
| **¿Existe un alto contraste entre el color de fuente y el fondo?** Sí, ya que para el usuario será más fácil leer los textos de la página si las letras se distinguen bien del fondo. El fondo negro irá acompañado de una tipografía clara y el fondo blanco ira acompañado de una tipografía oscura. | **Sí** |
| **¿Incluyen las imágenes atributos 'alt' que describan su contenido?** N/A | **-** |
| **¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla?** Sí, ha sido probado para los diferentes navegadores más usados hoy en día. | **Sí** |
| **¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar plugins adicionales?** Sí, esta aplicación web no requiere de plugins adicionales para su uso. |  |
| **¿Se ha controlado el peso de la página?** Ya que esta aplicación no cuenta apenas imágenes u otros elementos de peso el peso de la página se reduce al texto que hay en ella, por tanto es ligera. | **Sí** |
| **¿Se puede imprimir la página sin problemas?** Sí | **Sí** |

6.4.10 Criterios de control y retroalimentación

Tabla 22 Pruebas de usabilidad y accesibilidad - Criterios de control y retroalimentación

|  |  |
| --- | --- |
| **Criterios** | **¿Se ha cumplido?** |
| **¿Tiene el usuario todo el control sobre el interfaz?** Sí, ya que no cuenta con banners o pop-ups ajenos al control del usuario. | **Sí** |
| **¿Se informa constantemente al usuario acerca de lo que está pasando?** Sí, cada vez que el usuario interactúe con el sistema, este se lo notificará de una forma u otra. | **Sí** |
| **¿Se informa al usuario de lo que ha pasado?** Sí, el sistema notificará al usuario en todo momento de los eventos que ocurran en la aplicación resultado de la interactuación entre este y el sistema. | **Sí** |
| **Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema?** Sí, se han diseñado mensajes estándar que incluyen el aviso del error así como su solución, de una forma no alarmista. | **Sí** |
| **¿Posee el usuario libertad para actuar?** Sí, ya que no hay ninguna funcionalidad por la que el usuario deba esperar (aunque no quiera). El usuario puede hacer en todo momento lo que quiera dentro de la aplicación. | **Sí** |
| **¿Se ha controlado el tiempo de respuesta?** Debido a la ligereza del lenguaje y entorno utilizado, así como de la rapidez de la base de datos en ejecutar las consultas, esta aplicación genera tiempos de respuestas rápidos en sus peticiones (milisegundos). | **Sí** |

6.5 Pruebas de rendimiento

Una de las características de rendimiento más importantes de esta aplicación web tiene que ver con Javascript, concretamente con la técnica de desarrollo web AJAX (Asynchronous Javascript and XML). Está técnica tiene una gran importancia ya que se ha añadido en la mayoría de funcionalidades de la aplicación, haciendo a esta más ligera de cara a la interacción con el usuario con tiempos de respuesta menores.

La principal característica de Ajax es que permite comunicarse con el servidor (pasándole los datos que se requieran en la petición) y retornar los datos que el servidor proporciona, pero sin necesidad de recargar la página web. Es decir, explicándolo con un ejemplo concreto de nuestra aplicación web, si un usuario crea una oferta de prácticas o envía un formulario para actualizar la información de su cuenta, solo se modificarán aquellos elementos del DOM de la página que nosotros queramos, sin necesidad de volver a recargar la página web y por ente todos sus elementos, proporcionando un rendimiento mucho mayor a la aplicación.

A parte de utilizar técnicas como Ajax para mejorar el rendimiento de a aplicación web, se han llevado a cabo otro tipo de mejoras para aumentar el rendimiento de nuestra aplicación:

* No duplicar código JavaScript (duplicidad de funciones).
* No incluir elementos pesados que ralenticen la aplicación (imágenes grandes, vídeos, etc.)
* Cargar las llamadas a las hojas de estilo CSS al principio de la página y agrupar las llamadas, para que la página cargue los estilos de los elementos lo antes posible.
* Hacer las llamadas a los scripts de Javascript lo más abajo posible en cada página, ya que como la mayoría de funciones actúan sobre los elementos del árbol DOM de la página, estos tendrán que estar cargados previamente para que luego las funciones Javascript los localicen sin problema.

En el caso de tener la aplicación web subida en un servidor online, sería interesante utilizar la herramienta Google PageSpeed Insights (<https://developers.google.com/speed/pagespeed/insights/>) ya que esta se encarga de mostrarnos la velocidad de acceso que tiene nuestra página web así como de analizar sugerir que componentes debemos cambiar para mejorar el rendimiento.

# DOCUMENTO 7: MANUALES

**D. MARTÍNEZ SUÁREZ, Wenceslao**

**TUTOR: Dña. SUAREZ CABAL, María José**

**FECHA: Julio 2017**

Índice de contenidos

[**DOCUMENTO 7: MANUALES** 139](#_Toc486815260)

[7.1 INTRODUCCIÓN 141](#_Toc486815261)

[7.2 LIBRERIAS 141](#_Toc486815262)

[7.2.1 BOOTSTRAP 141](#_Toc486815263)

[7.2.2 BYEBUG 141](#_Toc486815264)

[7.2.3 JQUERY 142](#_Toc486815265)

[7.2.4 NESTED FORMS 142](#_Toc486815266)

[7.2.5 PARANOIA 142](#_Toc486815267)

[7.2.6 POSTGRESQL 142](#_Toc486815268)

[7.2.7 POSTGRESQL SEARCH 142](#_Toc486815269)

[7.2.8 RAILROADY 143](#_Toc486815270)

[7.2.9 RSPEC 143](#_Toc486815271)

[7.2.10 WILL\_PAGINATE 143](#_Toc486815272)

[7.3 MANUAL DE INSTALACIÓN Y EJECUCIÓN 143](#_Toc486815273)

[7.3.1 BASE DE DATOS 143](#_Toc486815274)

[7.3.2 RUBY VERSION MANAGER 144](#_Toc486815275)

[7.3.3 EJECUCIÓN 145](#_Toc486815276)

7.1 Introducción

En el apartado XXX del documento XXX presentamos las tecnologías y herramientas utilizadas en el desarrollo de esta aplicación web con sus versiones. A continuación y con el objetivo de incluir aquella información necesaria para la instalación y la utilización del sistema, se describirán las librerías utilizadas en él y se detallarán los pasos a seguir para la correcta instalación en un equipo.

7.2 Librerías

A continuación se detallarán aquellas librerías, conocidas cono gemas en Ruby on Rails, más importantes que han sido utilizadas para el desarrollo de los diferentes módulos de la aplicación. Estas gemas se declaran en el fichero Gemfile, ubicado en la raíz del proyecto. Cada vez que queramos agregar una gema al proyecto tendremos que agregarlas al fichero Gemfile, y en el caso de desinstalarlas tendremos que borrarlas de ese fichero. Para instalar aquellas gemas incluidas en el Gemfile debemos ejecutar la siguiente orden en la consola de comandos:

$ bundle install

7.2.1 Bootstrap

Esta gema se ha instalado debido a la integración del framework de diseño Bootstrap. Gracias a esta gema y como se ha dicho en apartados anterior, se han podido integrar las diferentes hojas de estilo CSS y el diseño responsive con el que cuenta.

7.2.2 Byebug

Esta gema ha jugado un papel importante en el desarrollo de este trabajo fin de grado, y es que esta actúa como un depurador, controlando la ejecución del sistema en todo momento y proporcionando puntos de interrupción a ubicar donde el desarrollador desee, con el objetivo de controlar la traza de ejecución entre las diferentes llamadas del sistema.

7.2.3 jQuery

Esta gema se ha utilizado para importar la biblioteca de Javascript Jquery, la cual permite interactuar con los elementos DOM de una página web, el manejo de eventos o agregar la técnica AJAX. Actualmente es la biblioteca más usada de Javascript en la actualidad.

7.2.4 Nested forms

Esta gema se ha incluido en el sistema para la utilización de formularios anidados, los cuales se trabajan con modelos relacionados y generan una mejor experiencia de usuario, ya que se podrán integrar formularios de un modelo específico dentro del formulario otro modelo diferente en el caso de estar relacionados.

7.2.5 Paranoia

Esta gema permite almacenar aquellos objetos eliminados del sistema haciéndolos invisibles, de tal forma que si se produce el borrado de un objeto de forma inintencionada, este objeto seguirá en la base de datos con toda su información.

7.2.6 PostgreSQL

Esta gema sirve para establecer la conexión entre la aplicación web y nuesta base de datos PostgreSQL.

7.2.7 PostgreSQL Search

Esta gema se ha implementado para la creación de los buscadores en las páginas de Estudiantes, Empresas y Ofertas de la aplicación. Esta gema hace las búsquedas dentro de la base de datos PostgreSQL más fáciles y sencillas.

7.2.8 RailRoady

Esta gema se ha utilizado para la generación de diagramas UML de los modelos y controladores de la aplicación.

7.2.9 Rspec

Como hemos explicado detalladamente EN EL APARTADO DE PRUEBAS UNITARIAS, esta gema se ha utlizado para la realización de pruebas unitarias en los diferentes modelos más importantes de la aplicación.

7.2.10 Will\_paginate

Esta gema proporciona una API para desarrollar la paginación de objetos en nuestra aplicación web, tanto en las vistas como en los controladores.

7.3 Manual de instalación y ejecución

En primer lugar y como mencionamos en el apartado XX, el desarrollo de esta aplicación web se ha realizado sobre una distribución Linux, concretamente en Ubuntu 14.04 LTS, por tanto la instalación del entorno de trabajo se desarrollará sobre esta distribución.

7.3.1 Base de datos

En primer lugar instalaremos el sistema de bases de datos PostgreSQL y la herramienta pgAdmin3, con la que podremos gestionar gráficamente sus bases de datos. En la consola de comandos como super-usuario debemos introducir las siguientes líneas y ejecutarlas:

$ sudo apt-get update

$ sudo apt-get install postgresql postgresql-contrib libpq-dev

Una vez instalado PostgreSQL, arrancamos en su consola de comandos y configuramos la contraseña para nuestro usuario de postgres, en este caso llamado ‘postgresql’:

$ sudo -u postgres psql postgres

postgres**=**# \password postgres

Enter new password:

Enter it again:   
postgres**=**# \q

Una vez configurado nuestro usuario de postgres, instalamos pgAdmin3:

$ sudo apt-get install pgadmin3

7.3.2 Ruby Version Manager

Una vez completada la instalación de PostgreSQL y pgAdmin3, procederemos a instalar el framework Ruby on Rails. Una de las ventajas por las que se caracteriza la instalación de este framework es la rapidez con que se puede hacer, ya que utilizando ‘Ruby Version Manager (RVM)’ podremos realizar la instalación de Ruby on rails de manera sencilla. RVM actúa como un gestor de versiones que te permite instalar y mantener varias versiones de ruby a la vez. A continuación se detallan los pasos:

1. Primer instaamos los paquetes del sistema operativo requeridos por RVM:

$ sudo apt-get install -y git-core subversion

2. En segundo lugar instalaremos la firma requerida por RVM y procederemos a la instalación principal. La segunda orden que veremos a continuación instala las últimas versiones estables de rvm, ruby y Rails.

$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3

$ \curl -sSL https://get.rvm.io **|** bash -s stable --rails --ruby

3. Por último, la versión de Ruby que se ha utilizado en el desarrollo de este proyecto ha sido la versión ‘2.3.3’. La instalaremos de la siguiente forma:

$ rvm install 2.3.3

4. Una vez instalada la versión de Ruby con la que trabajaremos procederemos a instalar Rails dentro de la carpeta donde esté ubicado el trabajo. Para ello haremos uso de la gema bundler, instalándola en primer lugar.

$ sudo gem install bundler  
$ bundle install

La segunda orden ‘bundle install’ instalará Rails, así como todas sus gemas declaradas en el fichero Gemfile y sus dependencias.

7.3.3 Ejecución

Una vez hemos instalado el entorno de forma correcta, procederemos a arrancar la aplicación. En primer lugar debemos configurar la base de datos y luego crearla.

La configuración de la base de datos para el entorno de desarrollo está en el archivo ‘database.yml’ dentro de la carpeta ‘config’ ubicada en el directorio raíz de la aplicación. La configuración debe seguir estas declaraciones:

**development**:

**adapter**: postgresql

**encoding**: unicode

**database**: tfg\_v1\_development

**pool**: 5

**host**: localhost

**username**: postgres  
 **password**: hubb16

Una vez asegurados de que tenemos correctamente configurado el archivo ‘database.yml’ debemos abrir la consola de comandos y ubicados dentro de la carpeta donde esté el proyecto ejecutar las siguientes órdenes:

$ rake db:create  
$ rake db:seeds

La primera orden creará nuestra base de datos, mientras que la segunda la poblará con la información de prueba que he añadido en el fichero ‘seeds.rb’ ubicado en la carpeta ‘/db’.

Una vez instalados el entorno y la base de datos, podremos arrancar nuestra aplicación. El puerto por defecto donde Rails lanza nuestra aplicación es el puerto 3000, por lo tanto en nuestro navegador accederemos a nuestra aplicación escribiendo ‘localhost:3000’, siempre y cuando tengamos arrancado nuestro servidor, ejecutando para ello en la consola de comandos la orden ‘rails s’.