

“Año del Buen Servicio al Ciudadano”

**UNIVERSIDAD ANDINA NÉSTOR CÁCERES VELÁSQUEZ
FACULTAD DE INGENIERÍA DE SISTEMAS
C.A.P INGENIERÍA DE SISTEMAS**



INFORME DE PRÁCTICAS PRE PROFESIONALES

Prácticas realizadas en:
WELL DONE SOLUTIONS SAC

Elaborado por:
MAMANI PAUCAR, wilder

PARA OPTAR EL GRADO DE BACHILLER EN INGENIERÍA DE SISTEMAS

JULIACA - PERU
2017

A mi Familia por enseñarme el camino correcto.

Índice de figuras

5.1. CU-1 Seguridad	9
5.2. CU-2 Creación de Usuarios	10
5.3. CU-3 Casos de Uso primordiales	10
5.4. DC-A Diagrama de clases	11
5.5. DC-B Diagrama de clases completo	12
5.6. DP-1 Diagrama BPM	13
5.7. DA-1 Arquitectura del Sistema	14
5.8. DA-2 Arquitectura del Sistema (Diagrama de secuencia)	14
5.9. Sitio Web de Bootstrap	15
5.10. Carga de página tradicional	16
5.11. Carga de página SPA	17
7.1. Pantalla contribución backend	21
7.2. Pantalla contribución frontend	22
7.3. Pantalla login	23
7.4. Pantalla nueva empresa	23
7.5. Pantalla nuevo paciente	24
7.6. Pantalla exámenes a realizar	24
7.7. Pantalla examen audiometría	25
7.8. Pantalla antecedentes ocupacionales	25
7.9. Pantalla backend código principal	26
7.10. Pantalla backend código controlador audiometría	26
7.11. Pantalla backend código entidad contacto	27
7.12. Pantalla frontend código principal	27
7.13. Pantalla frontend código controlador espirometría	28
7.14. Pantalla frontend código comunicación con el backend	28

Índice General

1. Presentación	1
1.1. Objetivo del Informe	1
1.2. Periodo de Prácticas Pre Profesionales	1
1.3. Institución y Área de Trabajo	1
1.4. Funciones del Área de Trabajo	1
1.4.1. Planificar, organizar, diseñar e implementar software a medida	1
2. Datos Generales del Practicante	2
3. Aspectos Generales de la Empresa	3
3.1. Descripción	3
3.2. Ubicación	3
3.3. Teléfono	3
3.4. Portal Web	3
3.5. Actividades que Realiza	3
3.5.1. Desarrollo de software	3
3.5.2. Cursos presenciales	4
3.5.3. Creación de portales corporativos	4
4. Actividades Realizadas	5
4.1. Obtención de requisitos	5
4.2. Análisis de requisitos	6
4.3. Diseño de software	6
4.4. Desarrollo Frontend y Backend	6
4.5. Despliegue de sistemas en la Nube (vps)	7
5. Descripción del Proyecto Realizado	8
5.1. Objetivo	8
5.2. Justificación	8
5.3. Planificación	8
5.4. Metodología	8
5.5. Análisis del Sistema	9

5.6.	Diseño del Sistema	11
5.7.	Arquitectura del Sistema	14
5.8.	Desarrollo del Sistema	15
5.8.1.	Bootstrap	15
5.8.2.	AngularJS	16
5.8.3.	NodeJS y ExpressJS	17
5.8.4.	Spring y Spring Boot	18
6.	Conclusiones y Recomendaciones	19
6.1.	Conclusiones	19
6.2.	Recomendaciones	19
7.	Anexos	20
7.1.	Capturas de Pantalla de las contribuciones al código	20
7.2.	Capturas de Pantalla del Sistema	23
7.3.	Capturas de Pantalla del Código	26

Presentación

1.1. Objetivo del Informe

Aplicar los conocimientos adquiridos durante los cinco años de formación profesional dentro de la carrera académico profesional de **Ingeniería de Sistemas** de la “**Universidad Andina Néstor Cáceres Velásquez**” de la ciudad de Juliaca.

1.2. Periodo de Prácticas Pre Profesionales

Periodo de prácticas realizadas: **1 de septiembre del 2015 - 29 de febrero del 2016.**

1.3. Institución y Área de Trabajo

Las prácticas se realizaron en la empresa “**Well Done Solutions SAC**” en el área de **Desarrollo de Software**.

1.4. Funciones del Área de Trabajo

El Área de trabajo se dedica netamente al desarrollo de software a medida.

1.4.1. Planificar, organizar, diseñar e implementar software a medida

La mayoría de las empresas aún usan sistemas de información manuales, siendo una limitante de cara a aquellas empresas que hacen uso intensivo de sistemas automatizados; a esto se suma la necesidad de agilizar procesos de negocios y entregar información precisa a usuarios y clientes. Brindamos soluciones de software a medida, haciendo uso de tecnologías emergentes en el mundo del desarrollo de software.

Datos Generales del Practicante

APELLIDOS Y NOMBRES	: Mamani Paucar, wilder
DOMICILIO ACTUAL	: Jr. Mariano Melgar #657
DNI	: 73529058
CELULAR	: 951-503-009
ÁREA DE TRABAJO	: Desarrollo de Software
CARGO	: Analista Programador
EMAIL	: w11ld33r@gmail.com
GITHUB	: https://www.github.com/w11ld33r

Aspectos Generales de la Empresa

3.1. Descripción

- **WELL DONE SOLUTIONS SAC** es una empresa privada ubicada en la ciudad de Juliaca, Departamento de Puno, Perú
- Inició su funcionamiento el 5 de Octubre del año 2014

3.2. Ubicación

Pasaje 1^{ero} de Mayo 112 - Oficina 301

3.3. Teléfono

051 33-6975

3.4. Portal Web

<http://www.wdsolutions.pe>

3.5. Actividades que Realiza

La empresa “**Well Done Solutions SAC**” se dedica a brindar asesoría, desarrollo de software, cursos talleres, mantenimiento de servidores, entre otros servicios relacionados a la tecnología.

3.5.1. Desarrollo de software

Como se mencionó anteriormente, la empresa brinda soluciones de software a medida, haciendo uso de tecnologías emergentes en el mundo del desarrollo de software.

3.5.2. Cursos presenciales

La mayoría de jóvenes tanto recién egresados como autodidactas, muchos de ellos solo aprendieron a construir programas sencillos y no saben como enfrentarse a proyectos reales. La empresa **“Well Done Solutions SAC”** dicta cursos presenciales al público en general, quienes quieran aprender o reforzar conocimientos de las últimas tendencias en lenguajes de programación o arquitectura de software, entre otros.

3.5.3. Creación de portales corporativos

Tener presencia en internet es la mejor forma de darse a conocer ante mundo. Empresas privadas como públicas buscan tener dicha presencia para dar a conocer los productos o servicios que brindan. **“Well Done Solutions SAC”** brinda servicios de creación de portales corporativos a medida.

Actividades Realizadas

En el área de desarrollo de software colaboré en las distintas etapas del desarrollo del sistema que se desarrolló. Reuniones con usuarios para la obtención de requisitos, análisis de requisitos, diseño del sistema, desarrollo del sistema y por último puesta en producción del sistema.

A lo largo del periodo de prácticas utilice herramientas que ayudan a ser productivos, tales son: Sistemas de control de versiones (Git, Bitbucket, Github, Gitlab), Entornos de Desarrollo Integrado (Eclipse, Netbeans, IntelliJ IDEA, Spring Tool Suite), editores de texto (Sublime Text, Atom, Visual Studio Code, Brackets), etc.

4.1. Obtención de requisitos

La obtención de requisitos es la etapa inicial y fundamental de cualquier tipo de proyecto de software que se quiera realizar. Se enfocan sólo en la visión del sistema que tiene el usuario. La funcionalidad del sistema, la interacción entre el usuario y el sistema, los errores que el sistema puede detectar y manejar son parte de los requisitos [Bruedgge and Dutoit, 2002].

La obtención de requerimientos incluye las siguientes actividades:

- **Identificación de actores.** Durante esta actividad se identifican los diferentes tipos de usuario que el sistema soportará.
- **Identificación de escenarios.** Aquí se observan a los futuros usuarios y se desarrollan un conjunto de escenarios posibles para las distintas funcionalidades del sistema.
- **Identificación de casos de uso.** Una vez de acuerdo el usuario con el equipo de desarrollo, se abstraen los escenarios en casos de uso.

Hubo reuniones con los usuarios, escuchándolos activamente, proponiendo ideas, debatiendo, descartando casos de uso innecesarios. Para llegar a un acuerdo. Dentro de esta labor el equipo tenía un panorama general de todo el sistema y a su vez en cada reunión se tenía ideas y modelos de funcionalidades listas para implementar; y presentarlos en la próxima reunión para ser aprobado por los interesados del proyecto.

Esta labor se me fue encomendada a medida que me iba familiarizando con el equipo, ya que es una parte crítica de un proyecto de software.

4.2. Análisis de requisitos

El análisis de requisitos se enfoca en la producción de un modelo del sistema. El análisis de requisitos le proporciona al diseñador del sistema una representación de información y función [Pressman, 2006]. Aunque puede ser que el modelo de análisis no sea comprensible para los usuarios, ayuda a que los diseñadores del sistema verifiquen la especificación del sistema producida durante la obtención de requisitos.

El análisis de requisitos produce tres modelos individuales:

- **Modelo funcional.** Representado por casos de uso y escenarios.
- **Modelo de objetos de análisis.** Representado por diagramas de clase y objetos.
- **Modelo dinámico.** Representado por diagramas de estado y de secuencia.

Fui partícipe de esta actividad progresivamente, opinaba de acuerdo a los conocimientos que tenía, pero el equipo siempre estuvo dispuesto a ayudarme, con lo cual despejaba dudas e inquietudes.

4.3. Diseño de software

El Diseño de software es un proceso mediante el cual los requisitos se convierten en un plano para construir el software. Al inicio el plano representa una visión general del software. A medida que se va avanzando el proceso, se van representado partes del software a profundidad.

Parte del diseño de software se encarga de describir la descomposición en subsistemas desde el punto de vista de responsabilidades, dependencias, flujo de control, control de acceso y almacenamiento de datos.

Forme parte de esta labor activamente, ya que tenía un buen conocimiento acerca de patrones de diseño, bases de datos, descomposicion de sistemas, etc.

4.4. Desarrollo Frontend y Backend

La parte de programación, es donde utilizamos todos los diseños y diagramas para empezar a codificar en algún lenguaje de programación. El desarrollo frontend y backend requiere de habilidades y conceptos como: conocimientos de algoritmos, estructuras de datos, pruebas unitarias,

pruebas end-to-end, integración continua, etc.

Ingresé a la empresa desarrollando únicamente en el lado del frontend, conforme avanzaba en conocimientos, comencé a desarrollar frontend y backend, los cuales me permitieron conocer las tecnologías actuales que dominan el mercado del desarrollo de software empresarial.

4.5. Despliegue de sistemas en la Nube (vps)

En esta última actividad se pone en producción el sistema en su totalidad. Servicios como *Google Cloud Platform*, *Digital Ocean*, *Amazon Web Services*, etc, requieren de habilidades de administración de servidores que ofrecen estas empresas. Distino a un hosting compartido, las plataformas como *Google Cloud Platform* brindan un espacio de almacenamiento dedicado exclusivo y la libertad de configurar el servidor de acuerdo a necesidades específicas. También permite escalar a medida que el sistema crece.

Gracias a los conocimientos en servidores linux pude asumir esta tarea, la de configurar, instalar paquetes y dejar todo listo para el funcionamiento correcto en la nube.

Conforme el proyecto avanzaba, pude desenvolverme adecuadamente en todas las actividades mencionadas.

Descripción del Proyecto Realizado

Durante el periodo de prácticas se desarrolló un Sistema de Salud Ocupacional (basado en web), realizando un trabajo en equipo; por lo que el practicante estaba a cargo del desarrollo de ciertas funcionalidades del proyecto.

5.1. Objetivo

Análisis, Diseño e Implementación de un Sistema de Salud Ocupacional para la Clínica Del Valle de la ciudad de Juliaca.

5.2. Justificación

El sistema es necesario para la Clínica del Valle, porque les permitirá ahorrar tiempo y recursos, como también reducirá la tasa de errores; es necesario para los pacientes, porque se les brindará un servicio más rápido y con resultados precisos.

5.3. Planificación

La construcción del sistema tuvo una duración aproximada de seis meses. La planificación era flexible en cuanto a reuniones (cada dos semanas) con el cliente. Se llevó a cabo una primera reunión con el cliente, quién requería las funcionalidades primordiales/urgentes del sistema. Pasada las dos semanas se tenía un producto mínimo viable y funcional. Las iteraciones nos permitía ir añadiendo más funcionalidades.

5.4. Metodología

Se usó una metodología ágil apoyado en el conjunto de ideas que nos brinda “kanban” y el Análisis y Diseño Orientado a Objetos.

5.5. Análisis del Sistema

Se requiere un Sistema de Salud Ocupacional donde se puedan registrar empresas, pacientes (asociados o no a una empresa) que, deben ser sometidos a pruebas médicas de acuerdo a un perfil de exámenes previamente registrados. Terminada las pruebas, el sistema debe imprimir dichas pruebas junto con los datos calculados automáticamente.¹

A continuación se muestran algunos diagramas:

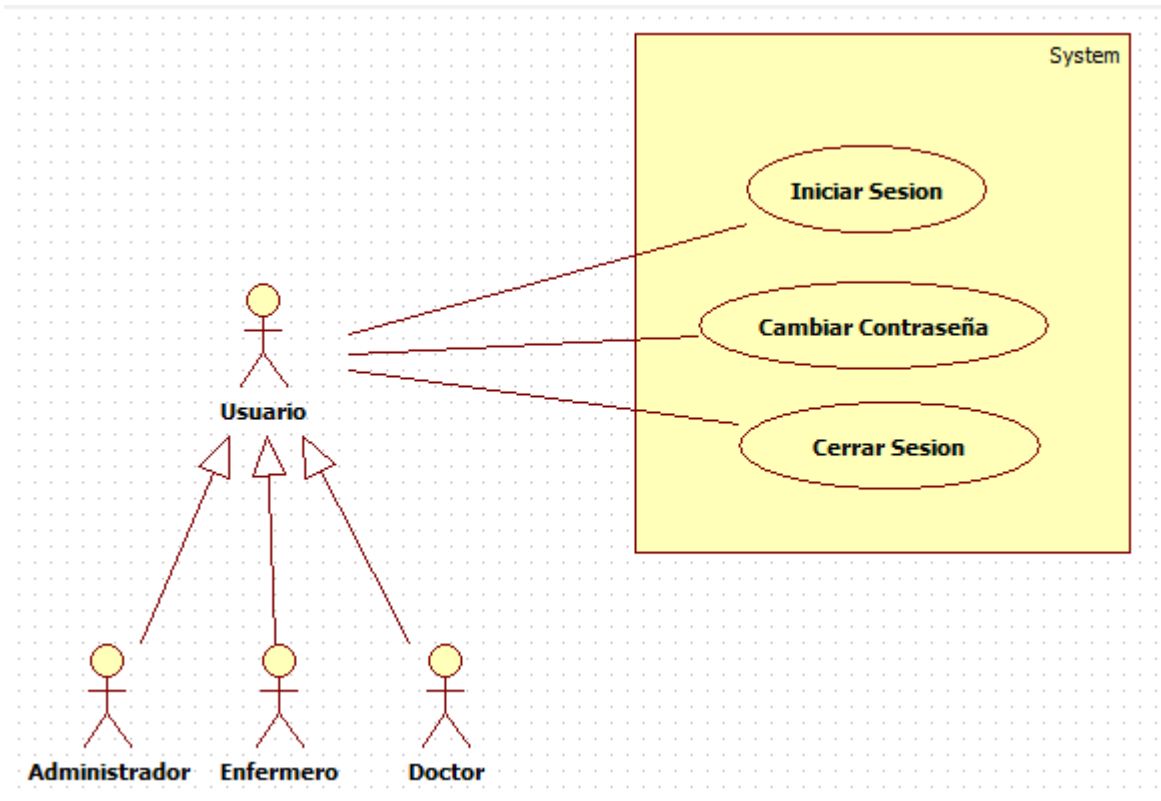


Figura 5.1: CU-1 Seguridad

¹El párrafo es un resumen del documento de requerimientos real.

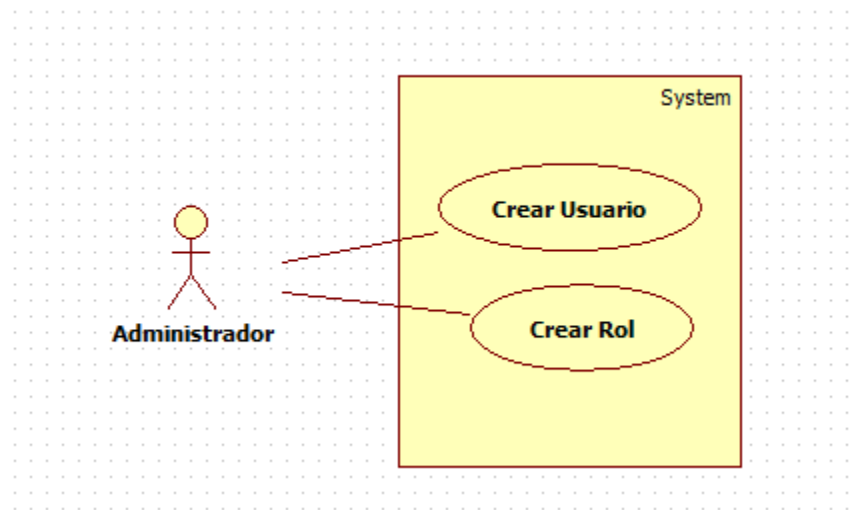


Figura 5.2: CU-2 Creación de Usuarios

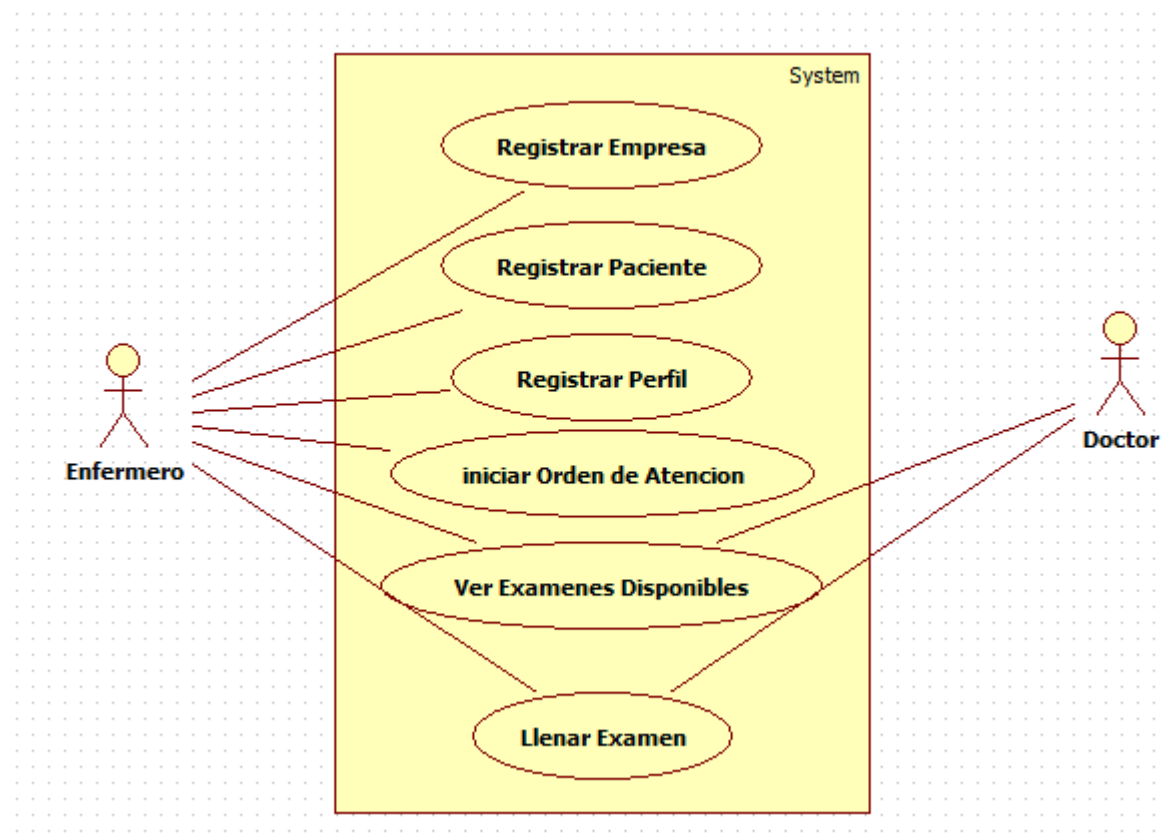


Figura 5.3: CU-3 Casos de Uso primordiales

5.6 Diseño del Sistema

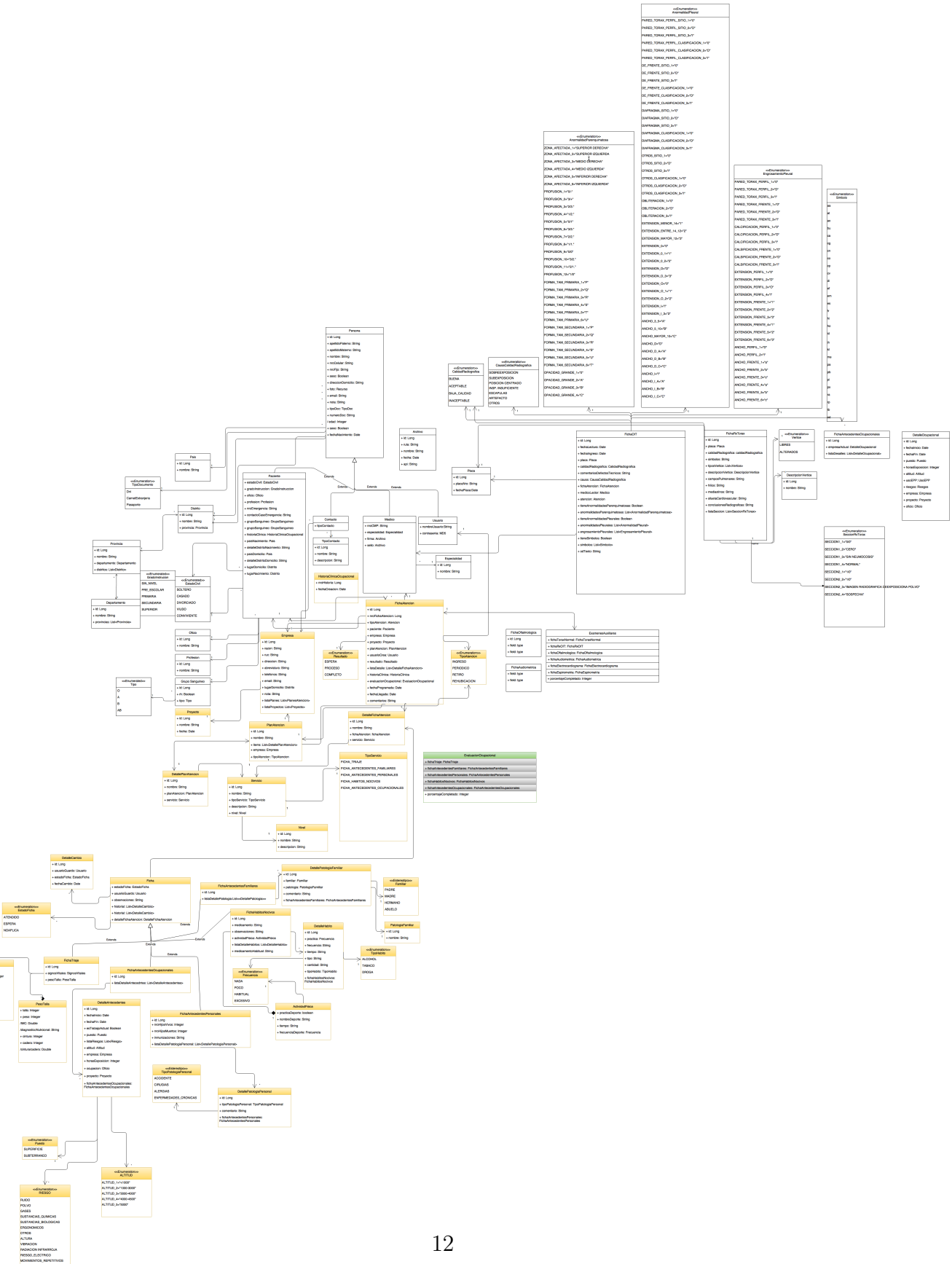


Figura 5.5: DC-B Diagrama de clases completo

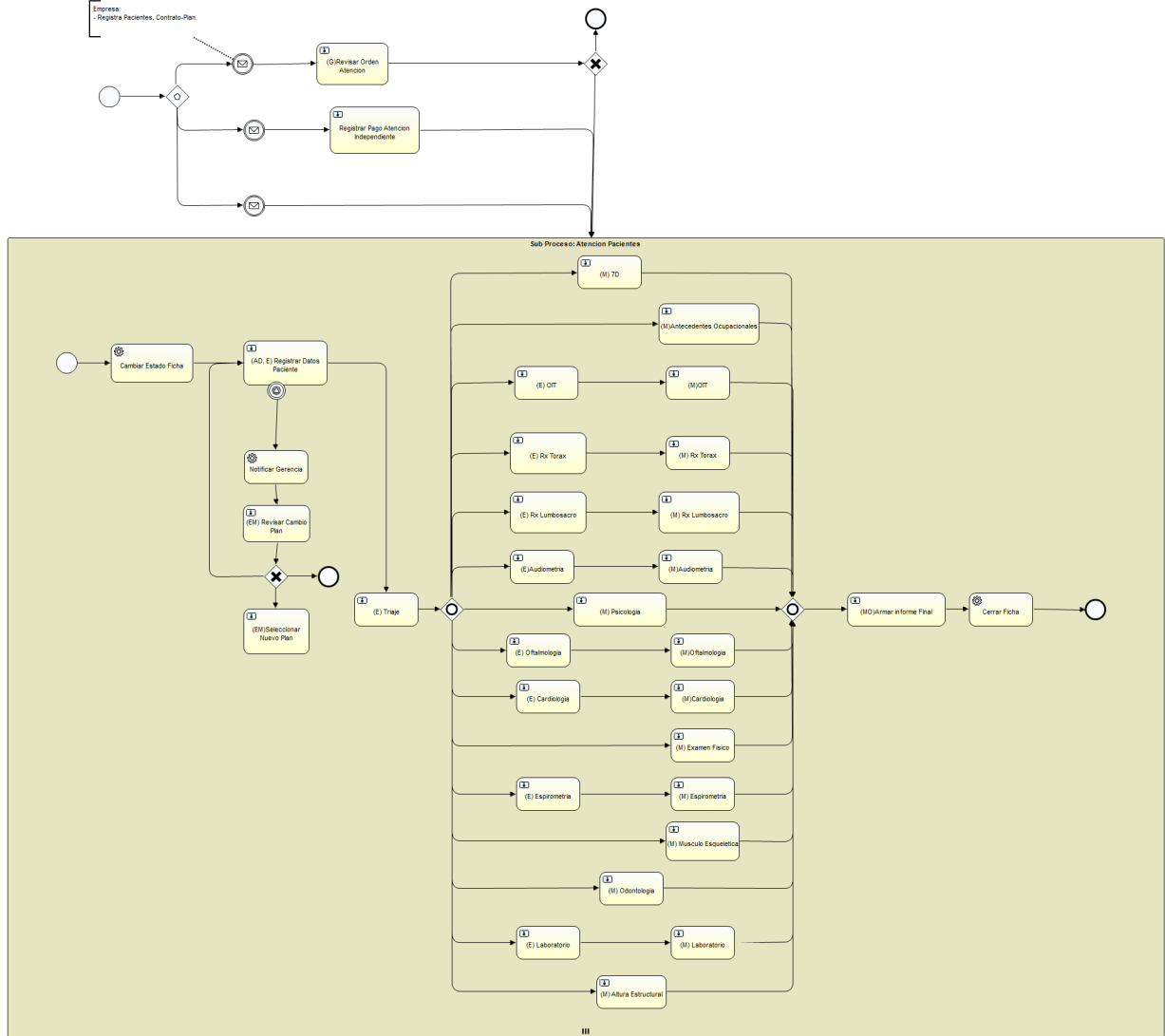


Figura 5.6: DP-1 Diagrama BPM

5.7. Arquitectura del Sistema

El Sistema sigue una arquitectura REST (Representational state transfer) ó de servicios web RESTful cliente-servidor que, funciona bajo el protocolo HTTP, figura 5.7. Básicamente el backend se encarga de proporcionar recursos (previa autorización) al frontend.

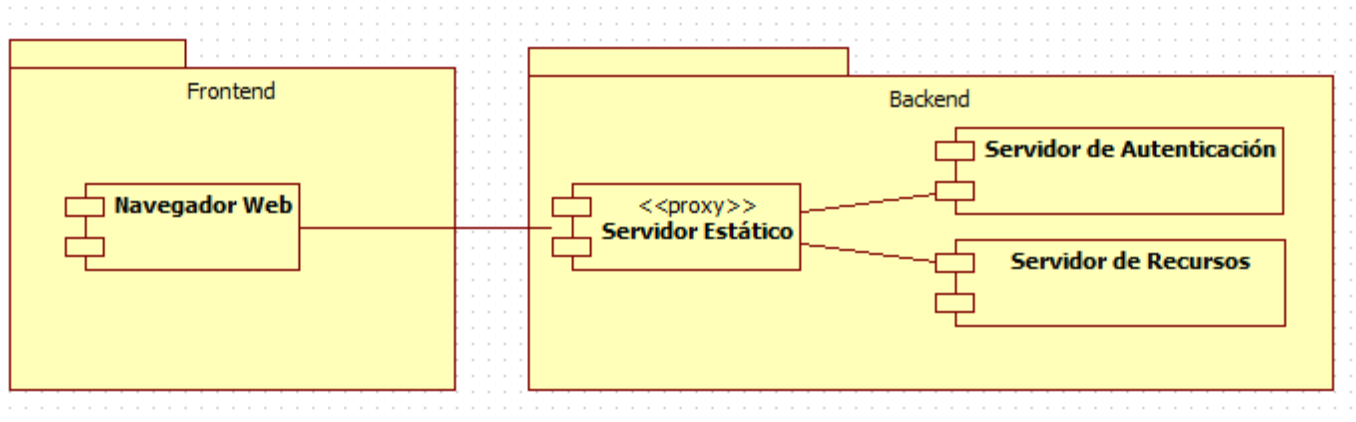


Figura 5.7: DA-1 Arquitectura del Sistema

En la figura 5.8 se muestra detalladamente la interacción de los componentes (frontend y backend).

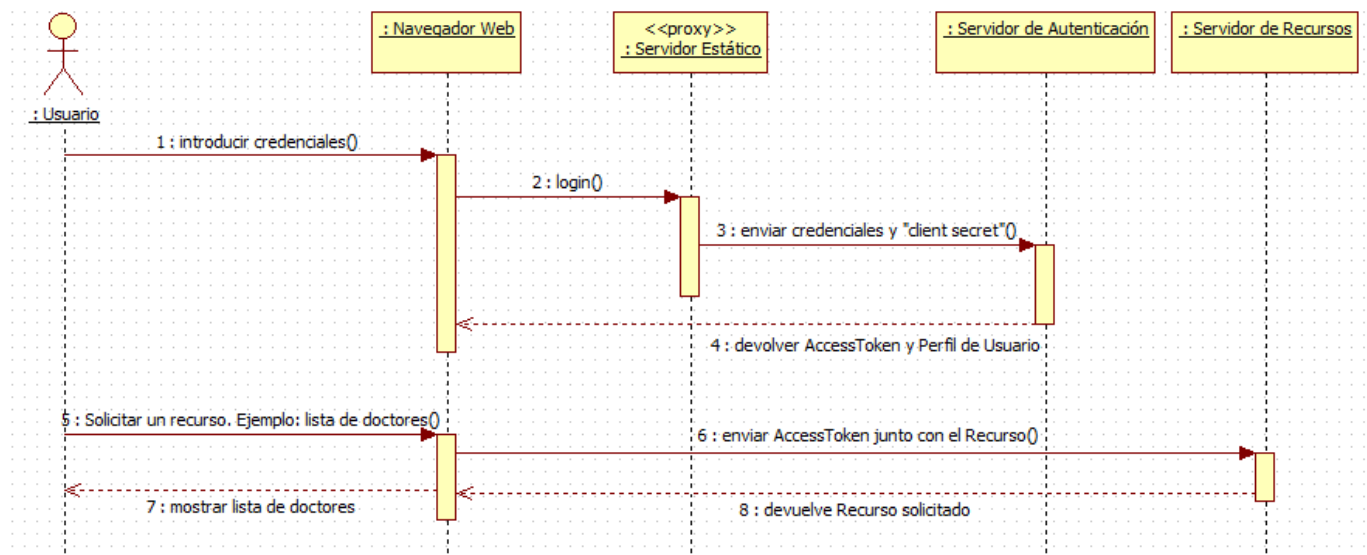


Figura 5.8: DA-2 Arquitectura del Sistema (Diagrama de secuencia)

5.8. Desarrollo del Sistema

Para el desarrollo del sistema se utilizó el siguiente stack de tecnologías (frontend y backend):

- HTML y CSS (Bootstrap)
- Javascript (AngularJS)
- NodeJS (ExpressJS)
- Java (Spring y Spring Boot)

5.8.1. Bootstrap

Bootstrap es un framework CSS que, facilita la creación de sitios web Responsive Web Design. Bootstrap viene con un conjunto de componentes de interfaz web listos para usar y una guía de estilos bastante completa, también provee grillas para el correcto maquetado de una web profesional.

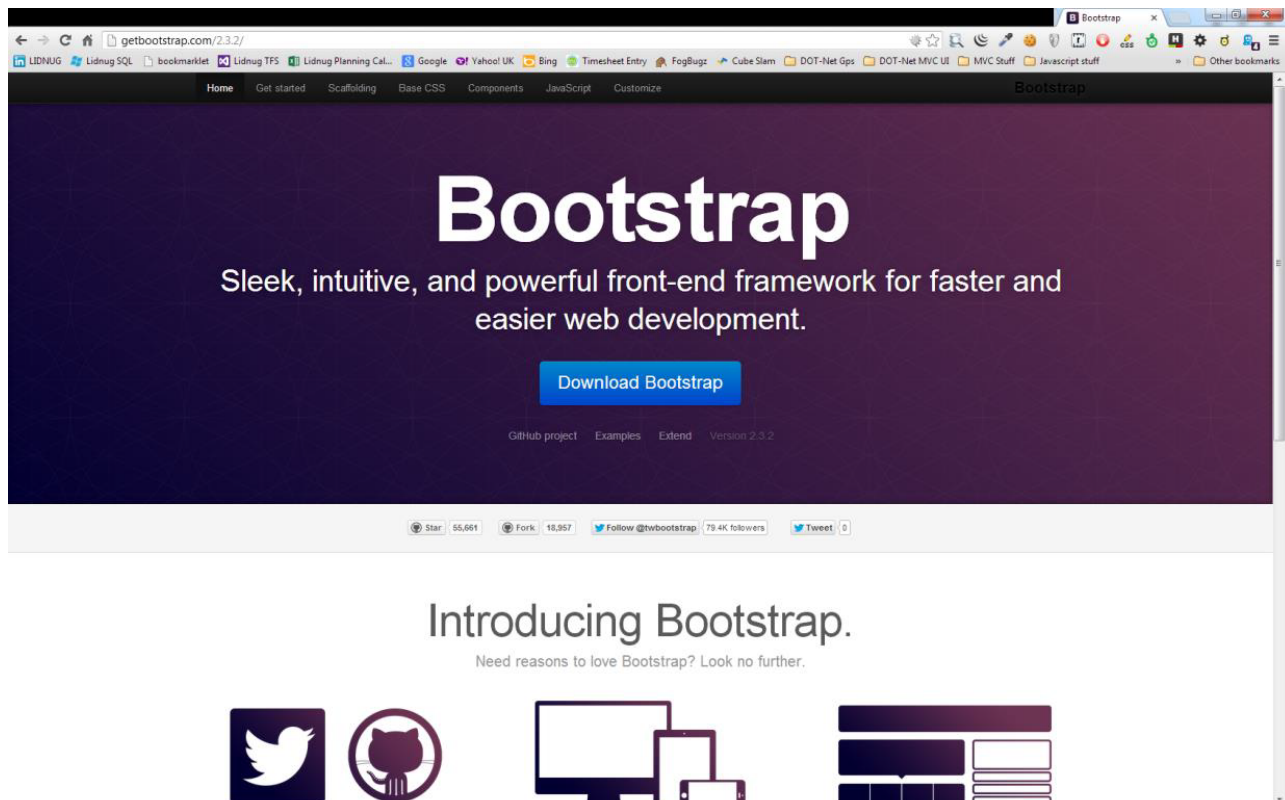


Figura 5.9: Sitio Web de Bootstrap

5.8.2. AngularJS

AngularJS es un framework de código abierto, soportado por Google. AngularJS permite a los desarrolladores crear aplicaciones web SPA (Single Page Applications), lo que facilita la experiencia de usuario, al no tener que cargar la aplicación web completa dos ó más veces.

A diferencia de las aplicaciones web SPA, en las aplicaciones web tradicionales (como se ilustra en la figura 5.10), un cliente web hace una petición al servidor y éste responde con toda la página web completa; luego, el usuario hace click en un enlace interno, el servidor vuelve a cargar la página web completa y se produce un retraso en la carga de la página, lo que produce una mala experiencia de usuario.³

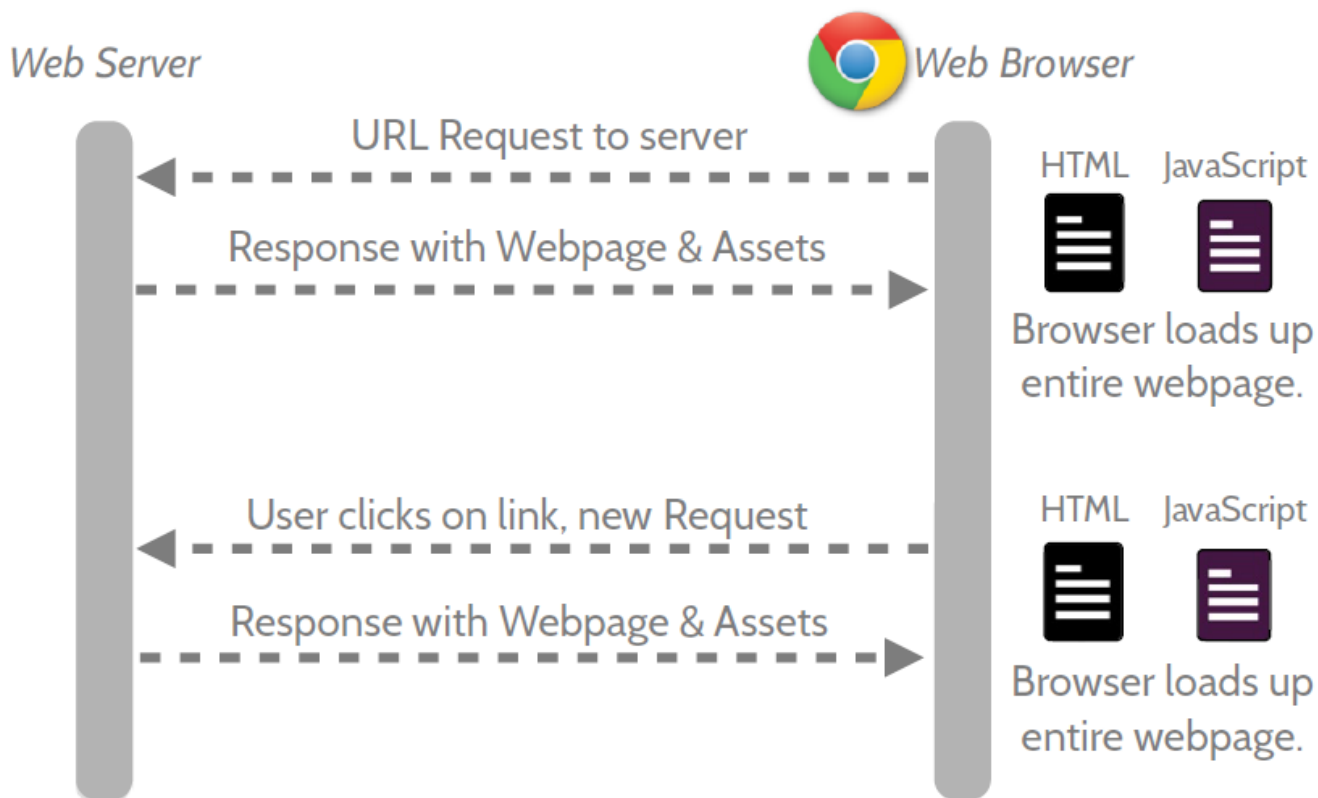


Figura 5.10: Carga de página tradicional

³Fuente de la imagen: <https://www.codeschool.com/course/shaping-up-with-angularjs>

En las aplicaciones SPA, el cliente solicita una web y el servidor inicialmente sirve la aplicación web completa (una sola vez); luego, el usuario hace click en un enlace interno, el servidor solo responde con el recurso solicitado, más no con la página web completa. En la figura 5.11 se muestra esta interacción.⁴

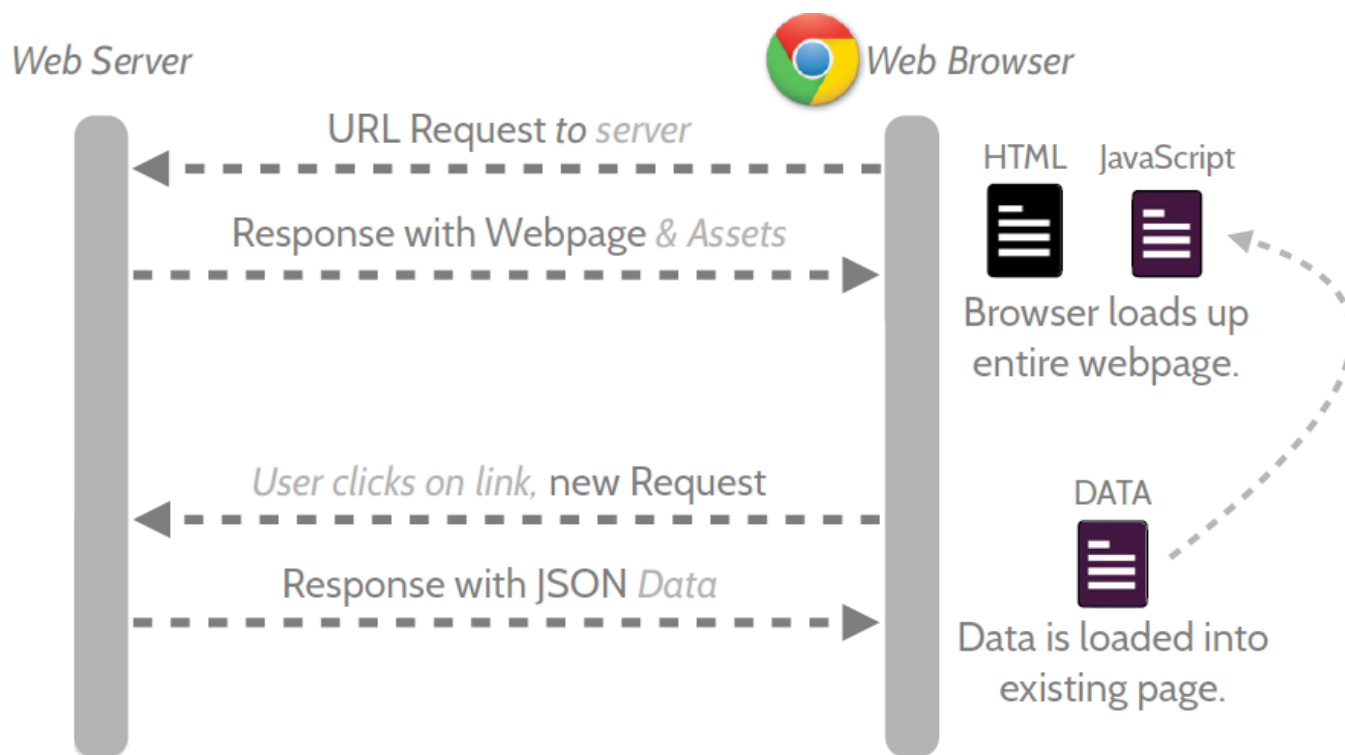


Figura 5.11: Carga de página SPA

5.8.3. NodeJS y ExpressJS

NodeJS no es un lenguaje de programación, es una plataforma capaz de ejecutar aplicaciones complejas escritas en Javascript; se ejecuta en el lado del servidor como cualquier otro lenguaje (Java, Python, Ruby, etc). Para el sistema fue usado como un proxy, y a la vez como un servidor estático.

ExpressJS es un framework web para NodeJS. Proporciona una API sencilla de utilizar, y agiliza el desarrollo de proyectos backend.

⁴Fuente de la imagen: <https://www.codeschool.com/course/shaping-up-with-angularjs>

5.8.4. Spring y Spring Boot

Spring es un framework java para el desarrollo de aplicaciones a nivel empresarial, inició como una alternativa liviana al estándar **Java Enterprise Edition (JEE)**. Spring ofrece características como inyección de dependencias y un modelo de programación orientada a aspectos.

Spring brinda muchas facilidades a la hora de programar, pero la configuración es un tanto compleja. Spring Boot simplifica toda las configuraciones iniciales para comenzar un proyecto web de manera rápida. Cabe mencionar que Spring Boot no es una herramienta ni una alternativa a Spring, es un proyecto que necesita de Spring para facilitarnos la vida.

Spring Boot trae algo de magia al desarrollo de aplicaciones con Spring. Existen cuatro características fundamentales:

- **Configuración Automática:** Spring Boot proporciona de manera automática funcionalidades a nuestro proyecto.
- **Dependencias Iniciales:** De acuerdo al tipo de proyecto, Spring Boot facilita las librerías necesarias para iniciar un proyecto.
- **Interfaz de Línea de Comandos:** Esta característica opcional permite escribir trozos de código directamente en la terminal e ir probando funcionalidades.
- **Actuador (*The Actuator*):** Muestra características de una aplicación en tiempo de ejecución.

Conclusiones y Recomendaciones

6.1. Conclusiones

- Las prácticas realizadas en todo el proceso del desarrollo de software, obtención de requisitos, análisis de requisitos, diseño de sistemas, desarrollo e implementación; son actividades que el ingeniero de sistemas debe dominar, para realizar proyectos que ayuden a las personas y por ende a la sociedad.
- Uno de los obstáculos al que se enfrenta un practicante es el de trabajar en equipo, transmitir ideas con claridad, pero que con la práctica se logra superar.
- La oportunidad que me dio la empresa **WELL DONE SOLUTIONS SAC**, de hacerme participe en la construcción de un proyecto real desde un comienzo, fue un reto y que, además me fortaleció como ingeniero de sistemas.

6.2. Recomendaciones

- Se recomienda a la empresa **Well Done Solutions SAC** abra las puertas a nuevos estudiantes quienes estén a punto de realizar sus practicas pre profesionales, para que los futuros ingenieros de sistemas estén muy bien capacitados.
- Se recomienda a la empresa **Well Done Solutions SAC** brinde talleres informativos a estudiantes de los distintos semestres de la facultad de ingeniería de sistemas de esta casa de estudios.
- Se recomienda a la Carrera Académico Profesional de **Ingeniería de Sistemas** realizar convenios con empresas de desarrollo de software, y así el estudiante podrá enfrentarse a proyectos reales.
- Se recomienda a los estudiantes de la Carrera Académico Profesional de **Ingeniería de Sistemas** descargar este documento y su código fuente (hecho en \LaTeX) en el siguiente repositorio: <https://www.github.com/w11ld33r/informe-practicas>, para que puedan tener una noción de este informe y poder adecuarlo a sus necesidades.

7.1. Capturas de Pantalla de las contribuciones al código

A continuación se muestra las capturas de pantalla de algunas de las contribuciones al código fuente en el repositorio en la nube (Bitbucket):

Commits

Find commits

Author	Commit	Message	Date	Builds
Wilder	43402cd	Examenes plan-laboratori...	2016-02-21	
Wilder	c3f8a4a	PlanLaboratorio post	2016-02-20	
Wilder	9c6925c	Examenes laboratorio	2016-02-19	
Wilder	9ec0ff1	avance laboratorio	2016-02-19	
Wilder	5230576	Merge branch 'master' of ...	2016-02-18	
Wilder	3a6cdf5	ExamenLaboratorio put	2016-02-18	
jose	7e44d73	...	2016-02-18	
jose	9c6b660	Merge branch 'master' of ...	2016-02-18	
jose	268c9cc	...	2016-02-18	
Helard	54be5a9	Merge branch 'master' of ...	2016-02-18	
Helard	16fc3a1	laboratorio	2016-02-18	
Wilder	5884cf4	Merge branch 'master' of ...	2016-02-17	
Wilder	9dc5128	Empresas paginado	2016-02-17	
jose	90da1ca	...	2016-02-17	
Wilder	d21ef82	Merge branch 'master' of ...	2016-02-17	

Navigation: Overview, Source, **Commits**, Branches, Pull requests, Pipelines, Issues, Wiki, Downloads, Boards, Settings

Figura 7.1: Pantalla contribución backend

Commits

All branches ▾

Find commits

Author	Commit	Message	Date	Builds
Helard	44eed20	trabajador	2016-01-12	
jose	f4bc039	musculo	2016-01-11	
Wilder	f9edd23	Antecedentes personales...	2016-01-11	
Wilder	1e3d47c	Directiva 'Audiometria' c...	2016-01-11	
Wilder	1d4b05e	Directiva 'rango' modifc...	2016-01-11	
Wilder	291b13b	Merge branch 'master' of...	2016-01-07	
Wilder	6302df6	Integracion Examen Fisico	2016-01-07	
jose	8a3b894	oftalmologia+	2016-01-07	
jose	5567d6b	Merge branch 'master' of...	2016-01-07	
jose	9bd7ad6	..	2016-01-07	
Helard	2f8a7c3	contactos validacion nro ...	2016-01-07	
Wilder	a84a2fc	Merge branch 'master' of...	2016-01-06	
Wilder	052665d	Directiva SoloLetras mod...	2016-01-06	
Wilder	03f2dc3	audiometria cambios	2016-01-06	
Wilder	d67603e	Examen Fisico	2016-01-06	
Helard	316d826	roles	2016-01-05	

Figura 7.2: Pantalla contribución frontend

7.2. Capturas de Pantalla del Sistema

A continuación se muestra algunas de las capturas de pantalla del sistema:

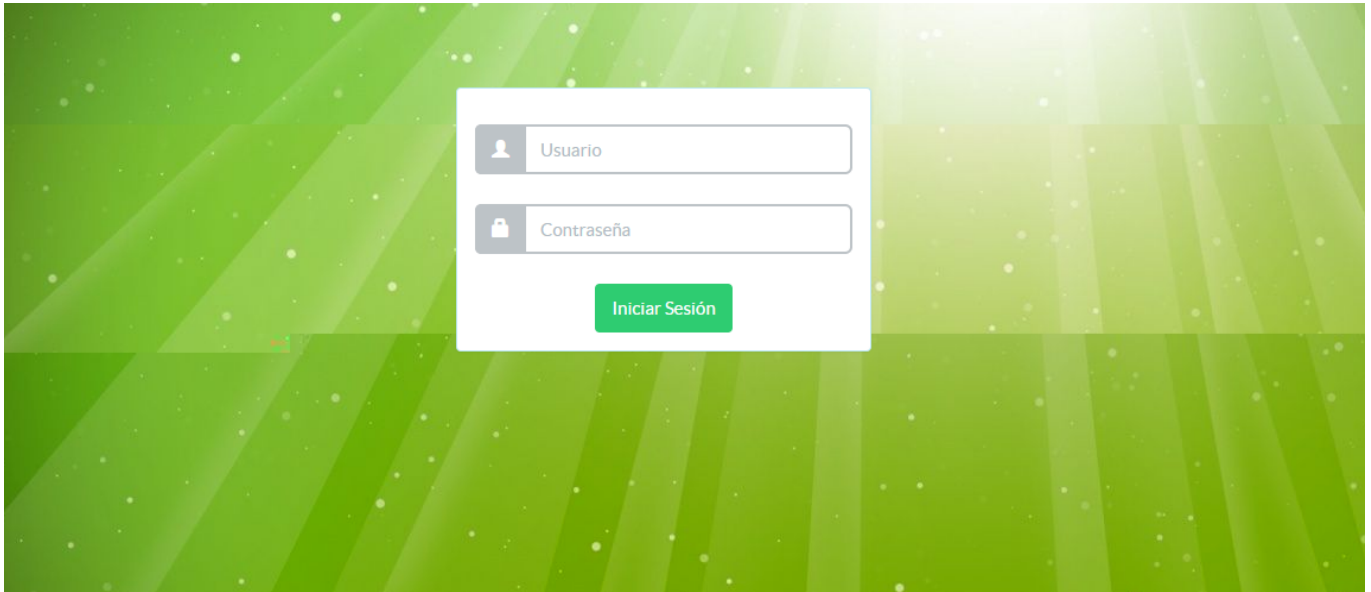


Figura 7.3: Pantalla login

Nueva Empresa

RUC <input type="text" value="65498798465"/>	Razón Social <input type="text" value="JULSA SAC"/>	Abreviatura <input type="text" value="JUL"/>
--	---	--

Actividad Económica

Tipo
☒ Empresa ☐ Contratista

Distrito

Dirección

Teléfono

Fax

Correo Electrónico

Nota

☒ Activar Campos



Seleccionar archivo | julsa.png

Figura 7.4: Pantalla nueva empresa

Trabajador: Mamani - 76595989

Razón Social	Ruc	Dirección	Email
JULSA SAC	6549879846	Terminal Terrestre Juliaca	contacto@julsa.com

Tipo Documento	Nro. de Documento
DNI	76595989

Nombres	Apellido Paterno	Apellido Materno
Wilder	Mamani	Paucar

☒ Activar Campos

Guardar **Cancelar**

Figura 7.5: Pantalla nuevo paciente

Salud Ocupacional | Del Valle

Actividades Disponibles

Mis Tareas Disponibles

N°	Paciente	Actividad	Empresa	Orden	Plan	Puesto Trabajo	Creado	Acciones
1	MAMANI PAUCAR, Wilder	(M)Antecedentes Ocupacionales	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	
2	MAMANI PAUCAR, Wilder	(M) Examen Físico	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	
3	MAMANI PAUCAR, Wilder	(M) Musculo Esqueletica	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	
4	MAMANI PAUCAR, Wilder	(M) Odontologia	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	
5	MAMANI PAUCAR, Wilder	(M) 7D	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	
6	MAMANI PAUCAR, Wilder	(M) Altura Estructural	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	
7	MAMANI PAUCAR, Wilder	(M) Psicologia	JULSA SAC	0002	PERFIL A-Ingreso	Administrador	15 noviembre 2017 02:17	

Figura 7.6: Pantalla exámenes a realizar

7.2 Capturas de Pantalla del Sistema

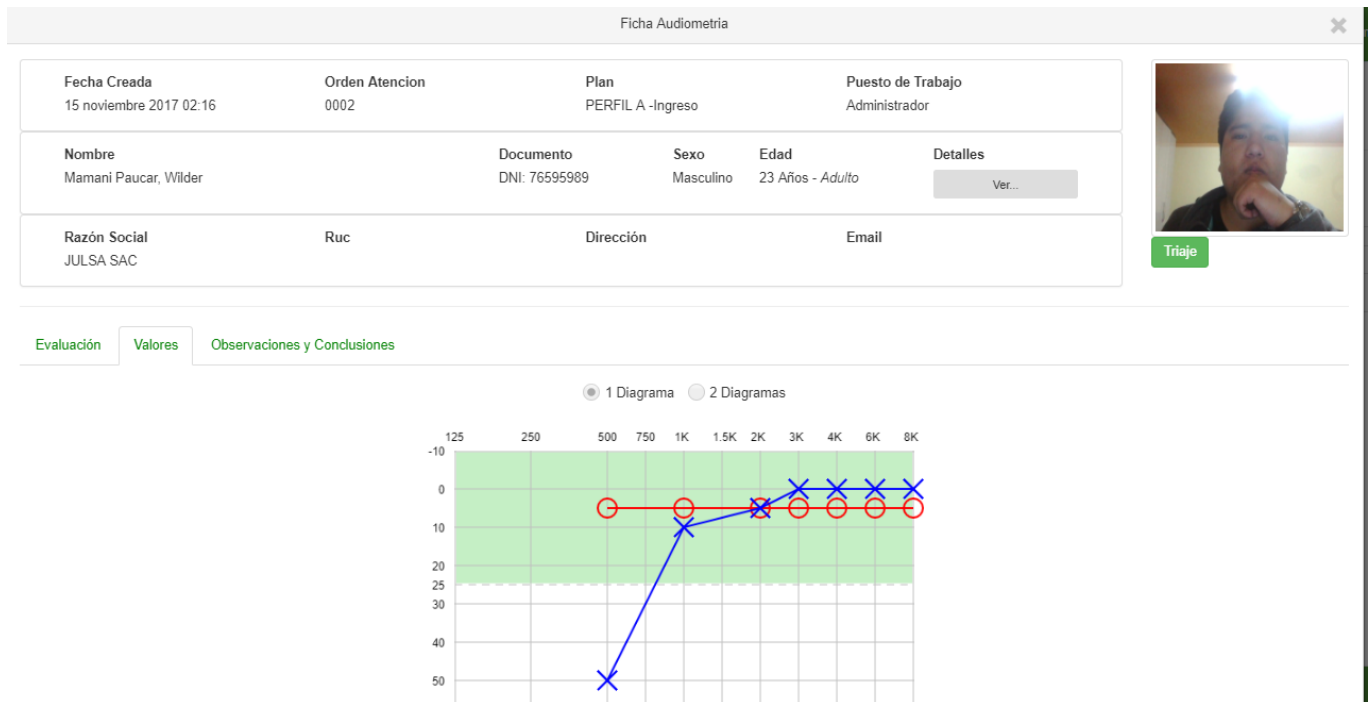


Figura 7.7: Pantalla examen audiometría

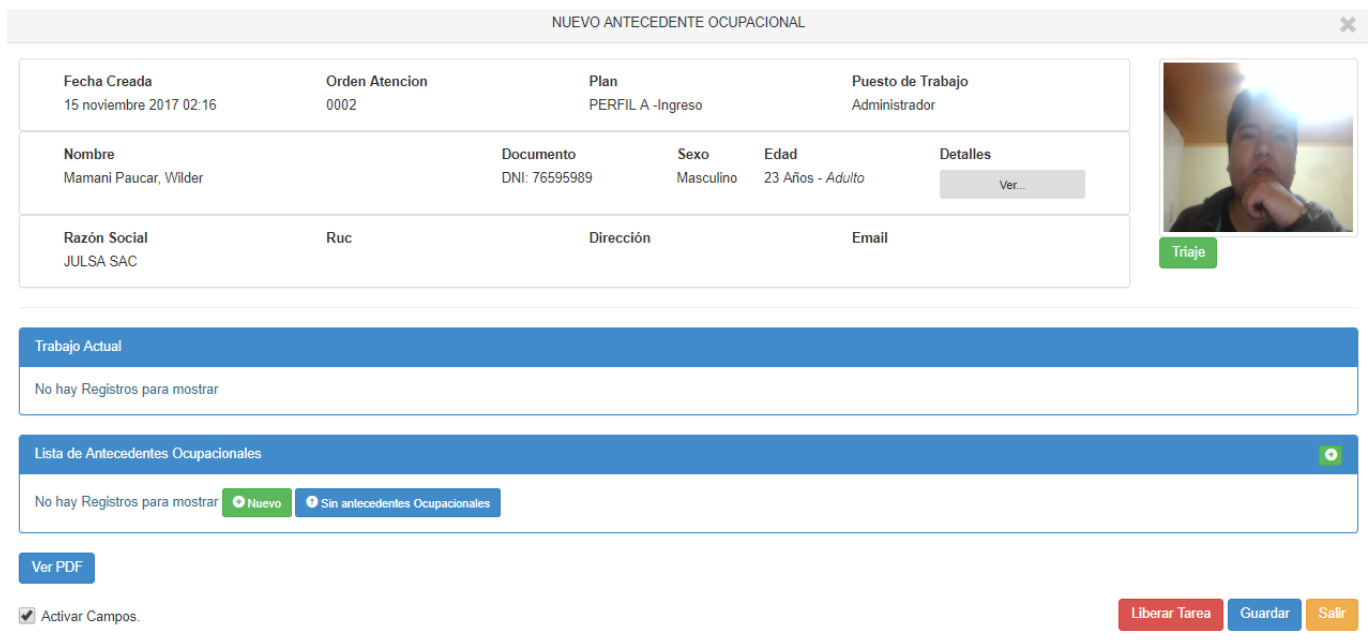


Figura 7.8: Pantalla antecedentes ocupacionales

7.3. Capturas de Pantalla del Código

A continuación se muestra algunas de las capturas de pantalla del código fuente (frontend y backend):

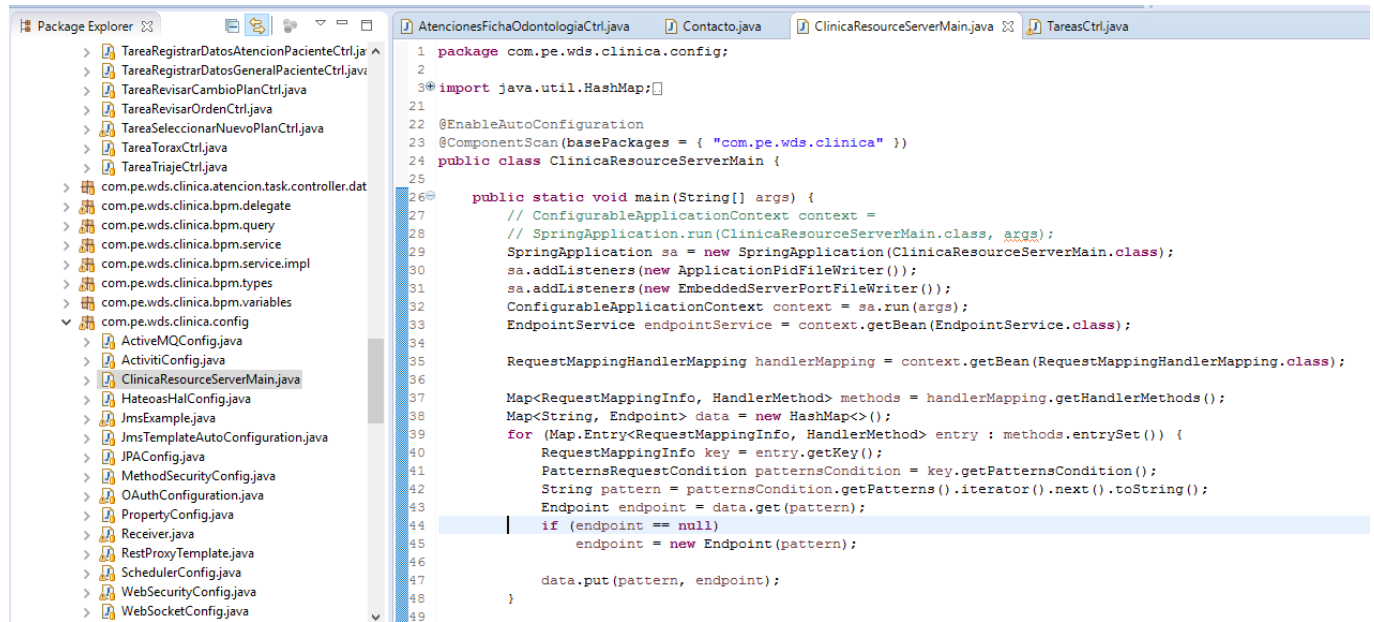


Figura 7.9: Pantalla backend código principal

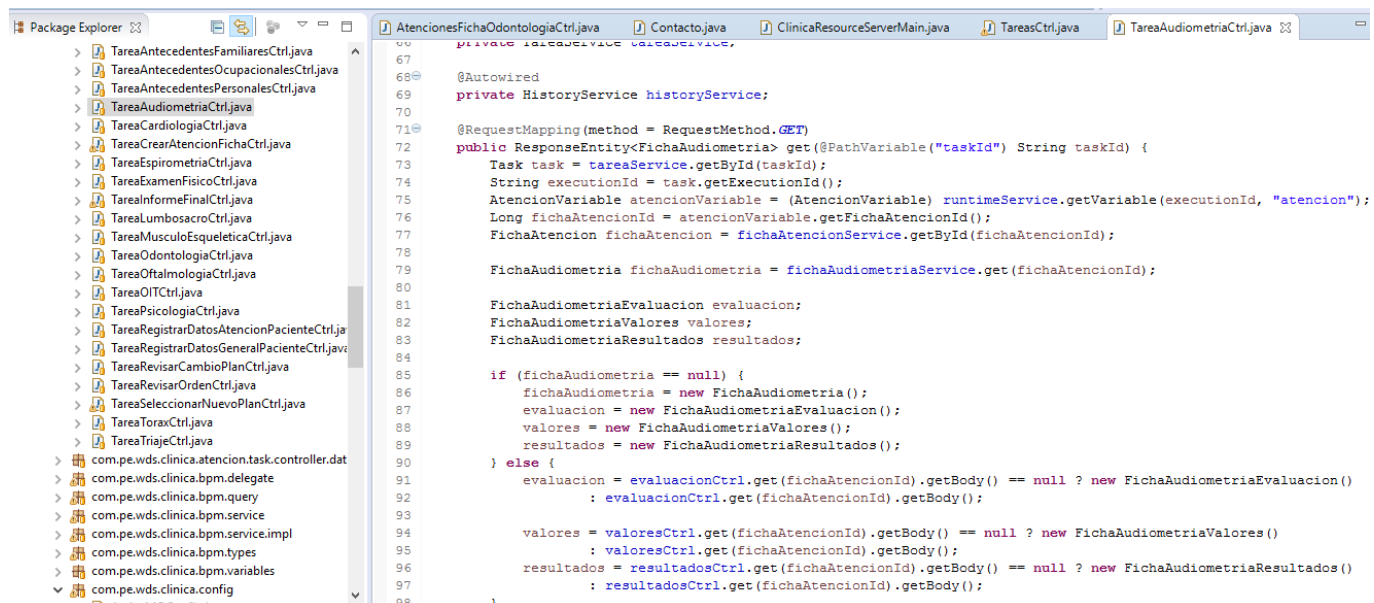


Figura 7.10: Pantalla backend código controlador audiometría

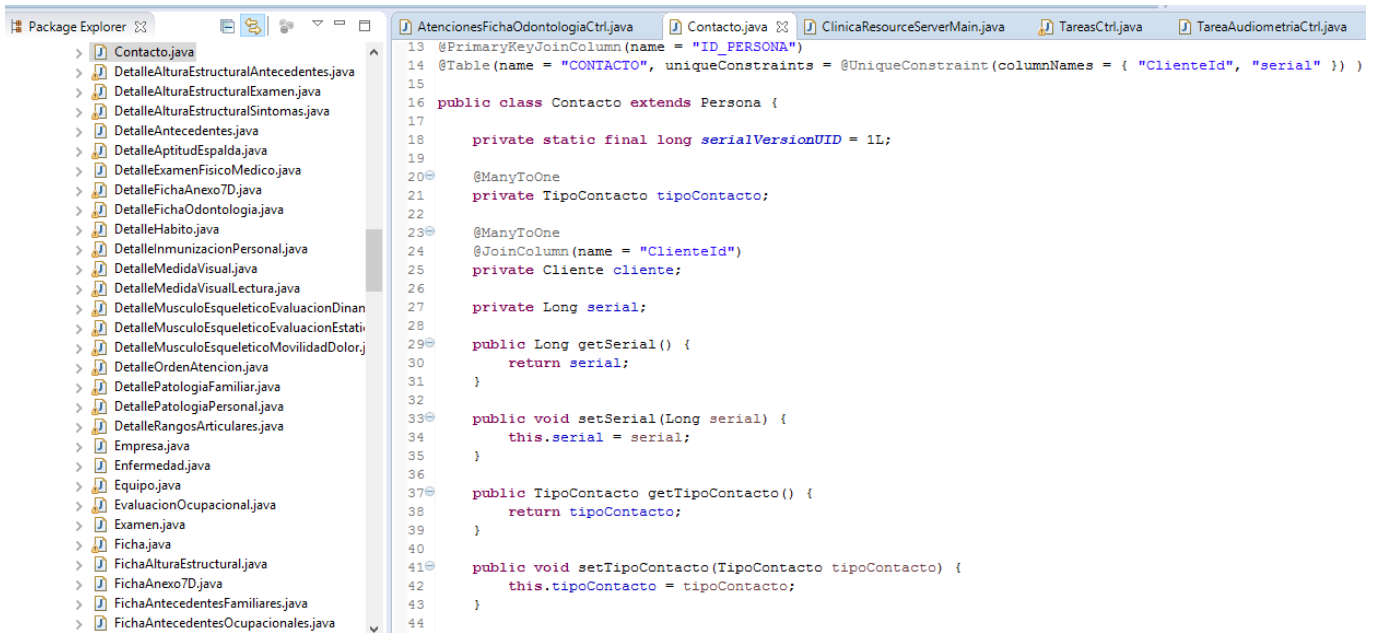


Figura 7.11: Pantalla backend código entidad contacto

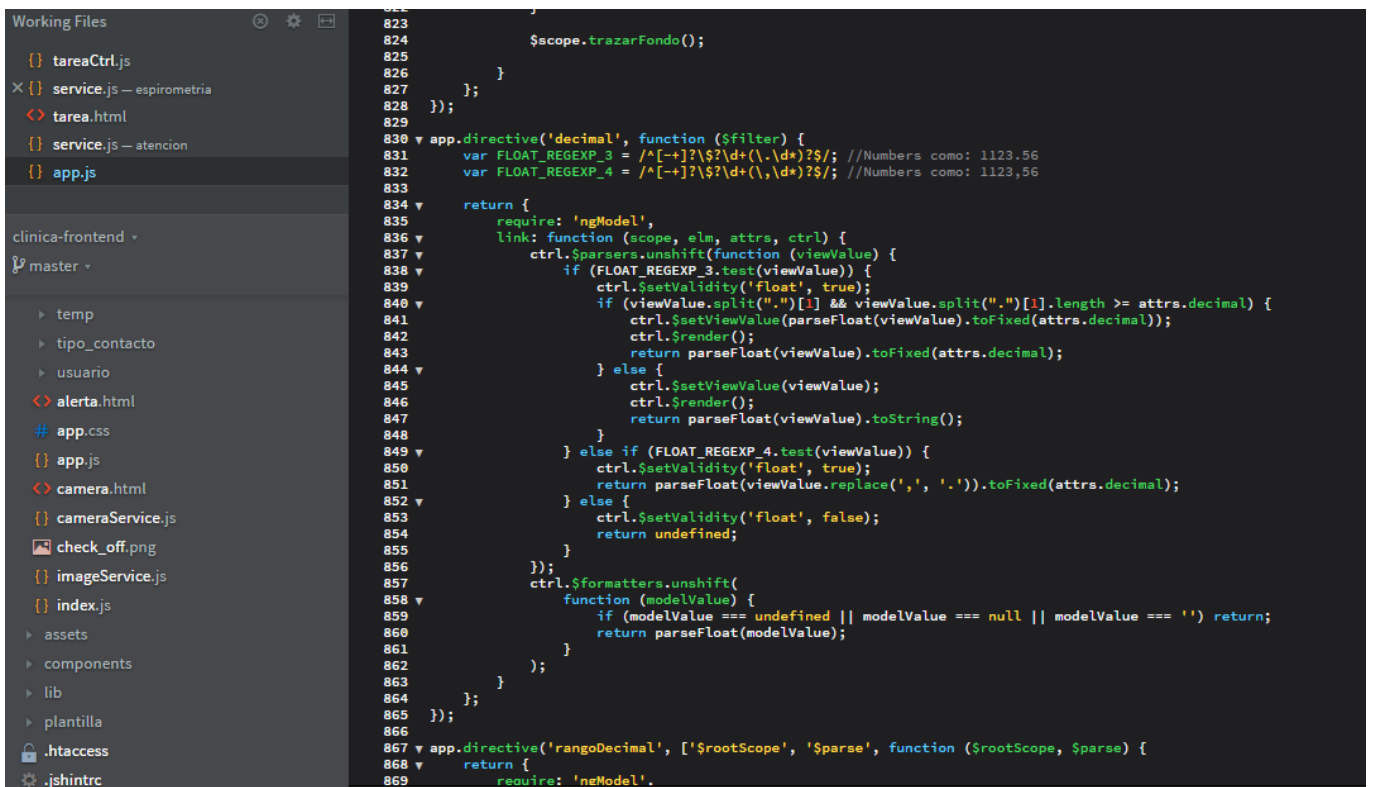


Figura 7.12: Pantalla frontend código principal

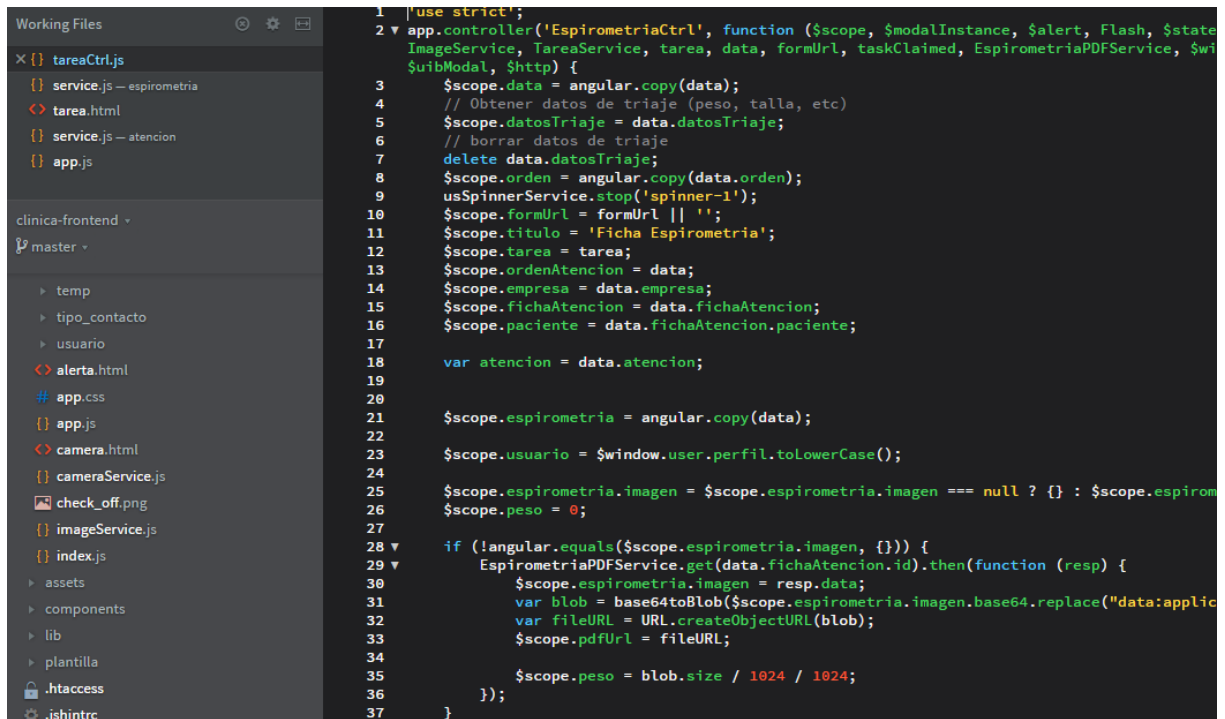


Figura 7.13: Pantalla frontend código controlador espirometría

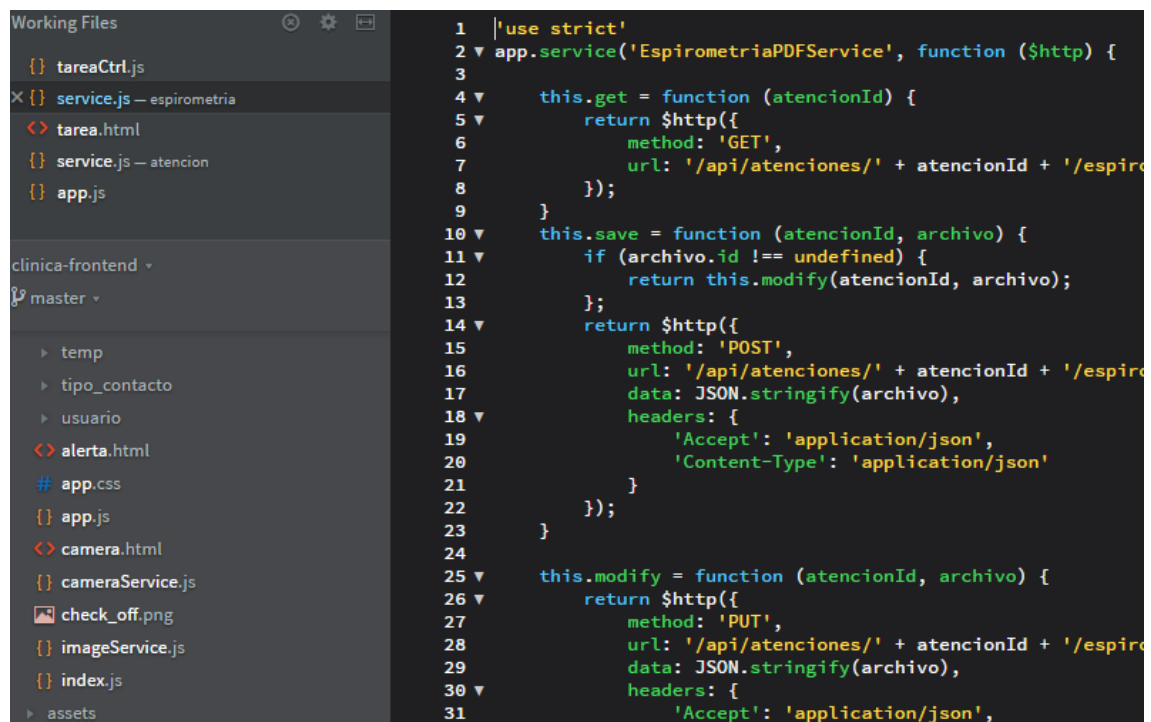


Figura 7.14: Pantalla frontend código comunicación con el backend

Bibliografía

- [Çaliskan and Sevindik, 2015] Çaliskan, M. and Sevindik, K. (2015). *Beginning Spring*. Wrox, 1 edition.
- [Bhaumik, 2015] Bhaumik, S. (2015). *Bootstrap Essentials*. Packt Publishing, 1 edition.
- [Branas, 2014] Branas, R. (2014). *AngularJS Essentials*. Packt Publishing, 1 edition.
- [Bruedgge and Dutoit, 2002] Bruedgge, B. and Dutoit, A. H. (2002). *Ingeniería del software orientado a objetos*. Prentice Hall, 1 edition.
- [Cantelon and Harter, 2014] Cantelon, M. and Harter, M. (2014). *Node.js In Action*. Manning, 1 edition.
- [Chandermani, 2015] Chandermani (2015). *AngularJS by Example*. Packt Publishing, 1 edition.
- [Freeman, 2014] Freeman, A. (2014). *Pro AngularJS*. Apress, 1 edition.
- [Gutierrez, 2014] Gutierrez, F. (2014). *Introducing Spring Framework: A Primer*. Apress, 1 edition.
- [Hahn, 2016] Hahn, E. (2016). *Express in Action*. Manning, 1 edition.
- [Haviv, 2014] Haviv, A. (2014). *MEAN Web Development*. Packt Publishing, 1 edition.
- [Pilone and Miles, 2008] Pilone, D. and Miles, R. (2008). *Head first Software development*. O'Reilly Media, 1 edition.
- [Pressman, 2006] Pressman, R. (2006). *Ingeniería del software*. Mc Graw Hill, 6 edition.
- [Scott, 2016] Scott, E. (2016). *SPA Design and Architecture*. Manning, 1 edition.
- [Shenoy and Sossou, 2014] Shenoy, A. and Sossou, U. (2014). *Learning Bootstrap*. Packt Publishing, 1 edition.
- [Somerville, 2011] Somerville, I. (2011). *Ingeniería de software*. Addison-Wesley, 9 edition.
- [Varanasi and Belida, 2015] Varanasi, B. and Belida, S. (2015). *Spring REST*. Apress, 1 edition.
- [Walls, 2016] Walls, C. (2016). *Spring Boot in Action*. Manning, 1 edition.

- [Whitten and Bentley, 2008] Whitten, J. L. and Bentley, L. D. (2008). *Análisis de sistemas, diseño y métodos*. Mc Graw Hill, 9 edition.
- [Wilson, 2013] Wilson, J. (2013). *Node.js the Right Way*. The Pragmatic Programmers, 1 edition.