



**Universidad
Europea**

LAUREATE INTERNATIONAL UNIVERSITIES

UNIVERSIDAD EUROPEA

**ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO
ÁREA DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS
TELECOMUNICACIONES**

GRADO EN INGENIERÍA INFORMÁTICA

**PROYECTO FIN DE GRADO
INFO JAPÓN**

Alumno: JORGE DE CASTRO CABELLO

Director: D. ENRIQUE PUERTAS

CURSO ACADÉMICO 2015-2016

TÍTULO: PROYECTO INFO JAPÓN

AUTOR: JORGE DE CASTRO CABELLO

DIRECTOR DEL PROYECTO: D. ENRIQUE PUERTAS

FECHA: 16 de Julio de 2016

Agradezco a mi tutor D. Enrique Puertas por su confianza y por darme la oportunidad de realizar este desafío personal. Contando con su guía y experiencia el proyecto es ahora más que nunca una realidad. Gracias a esta experiencia he podido descubrir hacia donde enfocar mi carrera profesional.

Agradezco a la profesora Dña. Asunción María Herreros por su profesionalidad, confianza y cercanía con el alumnado. Por demostrarme cuán importante es el trato personal y profesional y sobre todo por ser un ejemplo de perseverancia y buen hacer.

Agradezco a todos los compañeros, amigos, profesionales y familiares que se ha involucrado en el proyecto tanto en su desarrollo como en el apoyo que me han proporcionado.

Especialmente agradezco a mi madre por ser un ejemplo de generosidad, profesionalidad y constancia, porque el éxito que alcance en cualquier ámbito de mi vida será siempre un reflejo de su esfuerzo.

RESUMEN.

InfoJapón es una aplicación que, mediante el uso de agentes inteligentes, extrae información de páginas web de interés relacionadas con Japón para su posterior transformación y visualización en las aplicaciones web y móvil de InfoJapón. De esta manera el usuario podrá estar al tanto de los últimos artículos de los blogs hispanohablantes relacionados con Japón y tendrá acceso a módulos de búsqueda de estudios, ofertas de trabajo y viviendas en Japón.

Por tanto, InfoJapón nace con el objetivo de ser una plataforma que permita conectar al usuario con información actualizada sobre Japón de manera sencilla haciendo uso de las últimas tecnologías. De esta manera el proyecto pretende centralizar información de interés referente Japón y crecer en base a la demanda de los propios usuarios.

Palabras clave: Inteligencia artificial, Agente inteligente, Araña web, Arquitectura REST, Inteligencia de negocio, Procesos ETL.

ABSTRACT.

InfoJapón is an application that uses intelligent agents in order to gather relevant information from web pages related to Japan for transform the data and make it visible through a website and an android application accomplishing the user expectations. The user will be able to be informed about the latest articles of the Spanish-speaking blogs related to Japan and also will have access to different modules that allow to make advanced searches about study, work and live in Japan.

InfoJapón born with the objective of become a platform where to connect the user with the updated information about daily topics about Japan through an easy interface making use of the latest technologies. The project expects to centralize the information and grow with the user demand.

Key words: Artificial intelligence, intelligent agent, Web crawling, REST architecture, Business intelligence, ETL processes.

Índice

RESUMEN	5
ABSTRACT.....	5
Capítulo 1. INTRODUCCIÓN	14
1.1 PLANTEAMIENTO DEL PROBLEMA	14
1.2 OBJETIVOS DEL PROYECTO.....	14
1.3 ESTADO DEL ARTE	14
1.3.1 INTELIGENCIA ARTIFICIAL.....	15
1.3.2 WEB CRAWLING	16
1.3.3 ESTADO ACTUAL.....	17
1.4 ALCANCE Y LIMITACIONES	17
1.4.1 ARQUITECTURA SOFTWARE PROPUESTA.....	17
1.4.2 ARQUITECTURA HARDWARE PROPUESTA	20
1.4.3 ALCANCE DE LAS MÉTRICAS	20
1.5 RIESGOS.....	21
1.5.1 PLAN DE RIESGOS Y MITIGACIONES	21
1.5.2 MATRIZ DE IMPACTO	22
1.6 KEY PERFORMANCE INDICATOR (KPI)	23
1.7 DISPOSICIONES LEGALES	24
1.8 TECNOLOGÍAS Y SOFTWARE UTILIZADO	24
1.8.1 FORMATOS DE INTERCAMBIO DE DATOS	24
1.8.2 PROCESOS ETL (EXTRACCIÓN – TRANSFORMACIÓN – CARGA)	25
1.8.3 KETTLE (PENTAHO DATA INTEGRATION).....	26
1.8.4 BASES DE DATOS NOSQL.....	26
1.8.5 KIBANA	27
1.8.6 LOGSTASH.....	27
1.8.7 GIT Y BITBUCKET.....	27
1.8.8 API REST.....	28
1.8.9 APLICACIONES DE INTERNET ENRIQUECIDAS	29

1.8.10 AJAX.....	29
1.9 PLAN DE TRABAJO	30
Capítulo 2. ANÁLISIS DE REQUISITOS	33
2.1 REQUISITOS FUNCIONALES	33
2.2 REQUISITOS NO FUNCIONALES	35
Capítulo 3. DISEÑO	38
3.1 CASOS DE USO.....	38
3.1.1 RQ-01. MOSTRAR ARTÍCULOS	38
3.1.2 RQ-02. BUSCAR ARTÍCULOS	41
3.1.3 GRQ01. BUSQUEDA Y VISUALIZACIÓN DE INFORMACIÓN RELACIONADA CON ESTUDIOS.....	43
3.1.4 GRQ03. BÚSQUEDA DE OFERTAS DE EMPLEO	69
3.1.5 RQ04. BÚSQUEDA DE VIVIENDAS.....	71
3.2 DIAGRAMA DE PAQUETES SISTEMA – SERVIDOR	73
3.3 DIAGRAMA DE ACTIVIDADES SISTEMA - SERVIDOR.....	74
3.4 DIAGRAMA DE DESPLIEGUE	75
Capítulo 4. DESARROLLO	76
4.1 DESARROLLO BACKEND.....	79
4.1.1 LIBRERÍAS UTILIZADAS.....	79
4.1.2 DESARROLLO DEL MODELO.....	80
4.1.3 DESARROLLO DEL CONTROLADOR	100
4.2 DESARROLLO FRONTEND	108
4.2.1 MAPA APLICACIÓN CLIENTE	108
4.2.2 ESTRUCTURA DE APLICACIÓN CLIENTE	109
4.2.3 DESARROLLO APLICACIÓN.....	109
4.2.4 CAMBIOS PARA ADAPTAR A LA APLICACIÓN MÓVIL	133
Capítulo 5. PRUEBAS Y EVALUACIÓN.....	134
5.1 PRUEBAS MODELO	134
5.2 PRUEBAS DE CONTROLADOR Y SERVIDOR	134
5.2.1 TEST 1: /artículos.....	134
5.2.2 TEST 2: /artículos/buscar.....	136
5.2.3 TEST 3: /universidades/{prefecture}/{typeStudies}	137
5.2.4 TEST 4: /universidades/id/{id}.....	138
5.2.5 TEST 5: /universidades/id/{idUniversity}/{idFaculty}/access.....	138

5.2.6	TEST 6: /universidades/id/{idUniversity}/{idFaculty}/admissions	139
5.2.7	TEST 7: /universidades/id/{idUniversity}/{idFaculty}/facilities	140
5.2.8	TEST 8:: /universidades/id/{idUniversity}/{idFaculty}/info.....	140
5.2.9	TEST 9: /universidades/id/{idUniversity}/{idFaculty}/support.....	141
5.2.10	TEST 10: /posgrado/{prefecture}/{typeStudies}	142
5.2.11	TEST 11: /posgrado/id/{id}	142
5.2.12	TEST 12: /posgrado/id/{idGraduate}/{idFaculty}/access	143
5.2.13	TEST 13: /posgrado/id/{idGraduate}/{idFaculty}/admissions	144
5.2.14	TEST 14: /posgrado/id/{idGraduate}/{idFaculty}/facilities	144
5.2.15	TEST 15: /posgrado/id/{idGraduate}/{idFaculty}/info	145
5.2.16	TEST 16: /posgrado/id/{idGraduate}/{idFaculty}/support	146
5.2.17	TEST 17: /fp/{ prefecture}	146
5.2.18	TEST 18: /fp/id/{id}.....	147
5.2.19	TEST 20: /fp/id/{idTechSchool}/{idFaculty}/admissions	148
5.2.20	TEST 22: /fp/id/{idTechSchool}/{idFaculty}/info.....	148
5.2.21	TEST 23: /fp/id/{idTechSchool}/{idFaculty}/support	149
5.2.22	TEST 24: /fp/{ prefecture}	150
5.2.23	TEST 25: /escuelaldomas/id/{idSchool}	150
5.2.24	TEST 26: /escuelaldomas/{area}.....	151
5.2.25	TEST 27: /trabajo/{prefecture}/{specialty}.....	152
5.2.26	TEST 28: /vivir/{prefecture}.....	152
5.3	CONCLUSIÓN DE PRUEBAS.....	153
Capítulo 6.	DESPLIEGUE.....	154
6.1	DESPLIEGUE EN SERVIDOR.....	154
6.1.1	SECURIZACIÓN DEL SERVIDOR	155
6.1.2	INSTALACIÓN DE LA JVM (JAVA VIRTUAL MACHINE).....	159
6.1.3	INSTALACIÓN DE GIT (CONTROL DE VERSIONES).....	159
6.1.4	INSTALACIÓN DE APACHE MAVEN	159
6.1.5	CONFIGURACIONES PREVIAS A LA EJECUCIÓN DEL SERVICIO TOMCAT DE LA API REST	159
6.1.6	INSTALACIÓN DE APACHE2 + PHP 7.0	160
6.2	DESPLIEGUE BACKEND	160
6.2.1	CONFIGURACIÓN PARA LA CORRECTA ACTUALIZACIÓN DEL REPOSITORIO DE ARTÍCULOS	161

6.3 DESPLIEGUE FRONTEND (Cliente web)	161
6.4 GOOGLE PLAY (Cliente app)	162
Capítulo 7. ANÁLISIS DE MÉTRICAS	163
7.1.1 MÉTRICAS DE APLICACIÓN	163
7.1.2 GOOGLE ANALYTICS	164
Capítulo 8. CONCLUSIONES	165
Capítulo 9. FUTURAS LÍNEAS DE TRABAJO	166
PRESUPUESTO	167
BIBLIOGRAFÍA.....	168

Índice de Figuras

Figura 1 - Diagrama de casos de uso	38
Figura 2 - Diagrama de secuencia UC-01.....	40
Figura 3 - Diagrama de secuencia UC-02.....	42
Figura 4 - Diagrama de secuencia UC-RQ001-01-01 - 09	45
Figura 5 - Diagrama de secuencia UC-RQ01-01-10	47
Figura 6 - Diagrama de secuencia UC-RQ01-01-11	50
Figura 7 - Diagrama de secuencia UC-RQ01-02-01 - 05	52
Figura 8 - Diagrama de secuencia UC-RQ01-02-06	54
Figura 9 - Diagrama de secuencia UC-RQ01-02-07	57
Figura 10 - Diagrama de secuencia UC-RQ01-03-01 - 02	59
Figura 11 - Diagrama de secuencia UC-RQ01-03-03	61
Figura 12 - Diagrama de secuencia UC-RQ01-03-04	64
Figura 13 - Diagrama de secuencia UC-RQ01-04-01 - 02	66
Figura 14 - Diagrama de secuencia UC-RQ01-04-03	68
Figura 15 - Diagrama de secuencia UC-GR03	70
Figura 16 - Diagrama de secuencia UC-GR04	73
Figura 17 - Diagrama de paquetes	73
Figura 18 - Diagrama de actividades	74
Figura 19 - Diagrama de despliegue	75
Figura 20 - Arquitectura de la solución	76
Figura 21 - Proceso ETL de artículos.....	81
Figura 22 - Transformación Kettle para la extracción de archivos.....	81
Figura 23 - Diagrama de carga de artículos en Elastic Search.....	82
Figura 24 - Fichero para la indexación de ficheros JSON en Elastic Search	83
Figura 25 - Configuración para la actualización del repositorio de artículos.....	83
Figura 26 - Fichero JSON con los artículos de un blog específico	84
Figura 27 - JSON para la creación de un índice en ElasticSearch	84
Figura 28 - JSON de consulta sobre Elastic Search.....	85
Figura 29 - JSON respuesta a consulta en Elastic Search	85
Figura 30 - Construcción de consulta a Elastic Search en java.....	86
Figura 31 - JSON ejemplo sobre las respuestas en JSON de la solución	86
Figura 32 - Proceso de búsqueda de datos para su extracción mediante consultas del árbol DOM	87
Figura 33 - Proceso de crawling	87
Figura 34 - Estructura de clases de crawler	88
Figura 35 - Implementación de algoritmo divide y vencerás en el proceso crawling.....	88
Figura 36 - Respuesta JSON con lista de universidades	89
Figura 37 - JSON respuesta con información de universidad.....	90
Figura 38 - Respuesta JSON con lista de colegios de posgrado.....	91
Figura 39 - Respuesta JSON con información del colegio de posgrado	92

Figura 40 - Respuesta JSON con lista de colegios de formación profesional.....	94
Figura 41 - Respuesta JSON con información de colegio profesional.....	95
Figura 42 - Respuesta JSON con lista de colegios de japonés.....	96
Figura 43 - Respuesta JSON con información de colegio de japonés	97
Figura 44 - Respuesta JSON con lista de viviendas.....	99
Figura 45 - Implementación etiqueta @RestController de Spring.....	100
Figura 46 - Implementación de función como recurso de la API REST utilizando Spring	100
Figura 47 - Implementación de función para respuesta de errores 404	101
Figura 48 - Conjunto de funciones del controlador	107
Figura 49 - Mapa aplicación cliente	108
Figura 50 - Estructura aplicación cliente	109
Figura 51 - Página índice aplicación cliente	110
Figura 52 - Menu responsive desplegado aplicación cliente	110
Figura 53 - Menú responsive aplicación cliente.....	110
Figura 54 - Pantalla de búsqueda de artículos	111
Figura 55 - Código javascript para la carga de artículos al hacer scroll.....	112
Figura 56 - Petición AJAX artículos.....	112
Figura 57 - Pantalla de visualización de artículos cargados del JSON	113
Figura 58 - Pantalla de artículos buscados.....	113
Figura 59 - Pantalla de error si no se encuentran los artículos buscados.....	114
Figura 60 - Petición AJAX búsqueda de artículos	114
Figura 61 - Pantalla de visualización de artículos encontrados cargados del JSON	115
Figura 62 - Pantalla formulario paso 1 de estudios.....	115
Figura 63 - Pantalla formulario paso 2 de estudios.....	116
Figura 64 - Pantalla paso 3 estudios universitarios.....	117
Figura 65 - Pantalla paso 3 estudios de posgrado.....	117
Figura 66 - Pantalla paso 3 estudios de formación profesional	118
Figura 67 - Pantalla paso 3 estudios de lengua japonesa	118
Figura 68 - Pantalla paso 4 con lista de elementos devueltos	119
Figura 69 - Petición Wikipedia para la carga de información de prefectura.....	119
Figura 70 - Petición AJAX búsqueda de universidades.....	120
Figura 71 - Petición AJAX búsqueda de colegios de posgrado	120
Figura 72 - Petición AJAX búsqueda de colegios de formación profesional	121
Figura 73 - Petición AJAX búsqueda de colegios de japonés	121
Figura 74 - Sistema de paginación de resultados.....	121
Figura 75 - Pantalla de universidad	122
Figura 76 - Pantalla de colegio de posgrado	123
Figura 77 - Pantalla de colegio de formación profesional.....	124
Figura 78 - Pantalla 1 de colegio de japonés.....	125
Figura 79 - Pantalla 2 de colegio de japonés.....	125
Figura 80 - Botón de acceso a departamentos	126
Figura 81 - Pantalla departamentos 1.....	126
Figura 82 - Pantalla departamentos 2.....	127
Figura 83 - Peticiones AJAX información departamental.....	127
Figura 84 - Paso 1 formulario de trabajo	128

Figura 85 - Paso 2 formulario de trabajo	129
Figura 86 - Paso 1 búsqueda de viviendas	130
Figura 87 - Paso 2 búsqueda de viviendas	131
Figura 88 - Paso 3 listado de viviendas	132
Figura 89 - Pantalla de contacto.....	133
Figura 90 - Cambio de contraseña en servidor	155
Figura 91 - Fichero de configuración sudoers	155
Figura 92 - Fichero de configuración SSHD	156
Figura 93 - Fichero de configuración UFW	156
Figura 94 - Reglas Firewall servidor deploy.....	157
Figura 95 - Extracto de correo logwatch	158
Figura 96 - Fichero de configuración logStash	163
Figura 97 - Gráfica de peticiones a la API REST	164
Figura 98 - Ejemplo de registros almacenados por el sistema.....	164
Figura 99 - Panel de control Google Analytics	164
Figura 100 - Presupuesto del proyecto	167

Índice de Tablas

Tabla 1 - Plan de riesgos y mitigaciones.....	21
Tabla 2 - Matriz de impacto	22
Tabla 3 - Key Performance Indicators	23
Tabla 4 - Plan de trabajo	32
Tabla 5 - Requisitos funcionales.....	35
Tabla 6 - Requisitos no funcionales.....	37
Tabla 7 - UC-RQ001-01-01 - 09.....	44
Tabla 8 - UC-RQ01-01-10.....	47
Tabla 9 - UC-RQ01-01-11.....	49
Tabla 10 - UC-RQ01-02-01 - 05.....	52
Tabla 11 - UC-RQ01-02-06.....	54
Tabla 12 - UC-RQ01-02-07.....	56
Tabla 13 - UC-RQ01-03-01 - 02.....	59
Tabla 14 - UC-RQ01-03-03.....	61
Tabla 15 - UC-RQ01-03-04.....	63
Tabla 16 - UC-RQ01-04-01 - 02.....	66
Tabla 17 - UC-RQ01-04-03.....	68
Tabla 18 - UC-GR03	70
Tabla 19 - UC-GR04	72
Tabla 20 - Estructura proyecto.....	78
Tabla 21 - Librerías utilizadas en el modelo y controlador	80

Capítulo 1. INTRODUCCIÓN.

La creciente inmigración en Japón ha permitido acercar al resto del mundo la cultura nipona de la mano de emigrantes que hacen uso de la web 2.0 y las redes sociales. Esto ha generado los últimos años el surgimiento de comunidades en la red que comparten las mismas inquietudes entorno a un común denominador: Japón.

Con la apertura de Japón al mundo occidental cada vez más en alza, las instituciones japonesas proveen de información y recursos de carácter laboral, escolar, social y cotidiano en múltiples idiomas.

Tanto el contenido creado por inmigrantes como la información de las instituciones japonesas son de especial interés y se encuentran dispersas por la red. [1]

1.1 PLANTEAMIENTO DEL PROBLEMA

La gran cantidad de recursos que hay con referencia a Japón en la red son accesibles mediante los buscadores pero, pese a que existen comunidades angloparlantes que centralizan la información y generan contenido nuevo, no existen comunidades hispanohablantes de temática general que centralicen información relevante sobre vivir, trabajar y estudiar en Japón así como el contenido generado por bloggers hispanohablantes.

1.2 OBJETIVOS DEL PROYECTO

Punto Japón surge con el objetivo de suplir esta carencia en la red. Servir como directorio de contenido generado por los bloggers hispanohablantes y proveer recursos referentes a los puntos establecidos. Además, como su nombre indica, servirá como punto de encuentro donde se conecten las personas interesadas en el país nipón.

La solución generada será accesible mediante la web o mediante la aplicación de Punto Japón. De esta manera se pretende llegar al máximo número de usuarios dándoles siempre la mejor experiencia de uso.

La aplicación posee entonces dos objetivos definidos:

- OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón
- OBJ-02. Servir de directorio de artículos de blogs de emigrantes hispanohablantes en Japón

1.3 ESTADO DEL ARTE

Actualmente el siglo XXI se caracteriza como el siglo del Big Bang Data. El crecimiento de la red en los últimos 20 años ha traído consigo el universo digital que se expande a velocidades de vértigo debido al crecimiento de los datos que han venido motivados por la web 2.0 y el internet de las cosas. Se prevé que para el año 2020 más del 35% de todos los datos podrían considerarse útiles para las empresas pero luego tendrán que ser las mismas las que se encarguen de extraer valor a esos datos. [2]

En este panorama la rama de inteligencia artificial adquiere una notable relevancia gracias a sus aplicaciones como la minería de datos o el procesamiento del lenguaje natural pero

también nacen nuevos conceptos como Big Data o Datos masivos que hace referencia al almacenamiento de grandes cantidades de datos y sus procedimientos.

Debido a ello hay un crecimiento positivo de aplicaciones basadas en la recopilación de datos que hay en la red, la transformación, análisis y visualización de los mismos. Gracias a este sector en auge la tecnología nos brinda la posibilidad de trabajar con esa información y llevar al mundo real aplicaciones de uso relevante para el público general como es el caso de esta aplicación.

1.3.1 INTELIGENCIA ARTIFICIAL

La inteligencia artificial es un área multidisciplinaria moderna que a través de diversas ciencias como la informática, la lógica y la matemática estudia el desarrollo de sistemas capaces de realizar tareas humanas emulando el pensamiento y las actuaciones racionales.

- En el 384-322 a.C Aristóteles descubrió un conjunto de reglas que describían una parte del funcionamiento de la mente para obtener una serie de conclusiones racionales y en el 250 a.C fue Ctesibio (inventor y matemático griego en Alejandría) el primero en construir una máquina autocontrolada sin razonamiento (un regulador de secuencia de agua).
- En el 1315 Ramón Llull (teólogo y literario) en su libro Ars Magna, tuvo un conjunto de aproximaciones a que el razonamiento podía ser efectuado de manera artificial
- En 1936 Alan Turing, uno de los padres de la ciencia de la computación y precursor de la informática moderna diseño de manera formal la llamada máquina universal de Turing capaz de emular a cualquier otra máquina de Turing. Este dispositivo podía implementar cualquier cálculo formalmente definido.
- En 1955 Herbert Simon, Allen Newell y J. C. Shaw, desarrollan el IPL-11, primer lenguaje de programación orientado a la resolución de problemas de la IA. Un año más tarde desarrollan el “Logic Theorist”, el cual era capaz de demostrar teoremas matemáticos. Este programa demostró 38 de los 52 teoremas del segundo capítulo de "Principia Mathematica" de Russel y Whitehead, uno de ellos incluso de manera más elegante a la propuesta por los autores.
- En el 1955 surge también la Inteligencia Artificial como disciplina de la Informática en la Conferencia de Computación de Dartmouth. La conclusión de la conferencia fue que una máquina podía llegar a simular inteligencia humana. Empieza por tanto el periodo del entusiasmo con respecto a esta disciplina.
- A partir de 1955 se desarrollaron diversos sistemas para la resolución de problemas de manera artificial.
- Del 1966 al 1973 se dio una etapa de crisis motivada por limitaciones que se fueron encontrando los diferentes investigadores en el desarrollo de los sistemas.
- Del 1969 al 1979 se da una etapa de resurgimiento gracias a la llegada de los sistemas expertos que predicen la probabilidad de una solución bajo un conjunto de condiciones (DENDRAL), de la creación de nuevos lenguajes de programación como PROLOG y la llegada de los scripts (guiones para la ejecución de acciones, base de muchas técnicas actuales en informática). [3]

1.3.2 WEB CRAWLING

El Crawling es un proceso automático de exploración de aplicaciones web. El objetivo del web crawler es descubrir las páginas web de la aplicación web mediante la navegación de la misma. Esto se lograba normalmente mediante la simulación de interacciones del usuario considerando únicamente lado cliente de la aplicación.

Como la información en la web ha ido incrementando de manera drástica, los usuarios confían cada vez más en los sistemas de búsqueda para encontrar la información deseada. Para que estos sistemas de búsqueda puedan saber acerca de la nueva información disponible en la red, el web crawler se ejecuta constantemente actualizando la base de datos del sistema de búsqueda.

- En 1993 el CERN presentó la World Wide Web de forma pública. En este año nacieron los primeros cuatro web crawlers: World Wide Web Wanderer, Jump Station, World Wide Web Worm y el RBSE spider. El objetivo de estos cuatro web crawlers era la recolección de información y estadísticas sobre la web mediante un conjunto de URLs (enlaces web). Los primeros web crawlers descargaban iterativamente las URLs y las actualizaban en su repositorio de URLs mediante la descarga de las páginas web.
- En 1994 aparecieron WebCrawler y MOMspider que, además de recolectar estadísticas y datos sobre el estado de la web, introdujeron conceptos como *politeness* (educación) y *black-lists* (listas negras) a los crawlers tradicionales. WebCrawler fue el primer web crawler que descargó en paralelo 15 enlaces de manera simultánea. Desde World Wide Web Worm hasta el WebCrawler el número de páginas indexadas creció de 111000 a 2 millones. En los años venideros un par de web crawlers comerciales empezaron a estar disponibles (*Lycos*, *Infoseek*, *Excite*, *AltaVista* y *HotBot*).
- En 1998 Brin y Page intentaron solventar el problema de la escalabilidad de los web crawlers introduciendo un web crawler de gran escala llamado *Google*. Google solventó el problema de escalabilidad de varias maneras:
 - Redujeron el acceso a disco mediante técnicas de compresión e indexación (optimizaciones a bajo nivel).
 - Google calculaba la probabilidad de un usuario de visitar una página mediante un algoritmo denominado *PageRank*. De esta manera el crawler de Google podía reducir en base a demanda su acceso a páginas que no eran atractivas para los usuarios y de esta manera enfocar recursos a páginas que si lo eran mediante el uso de un usuario arbitrario simulado. Con esta técnica, la primera implementación de google logró descargar 100 páginas por segundo.
- En 1999 los problemas de escalabilidad fueron solventados de manera más amplia por Allan Heydon y Marc Najork con una aplicación denominada *Mercator*. Mediante el uso de componentes de terceros en un framework modular basado en Java Mercator llegó a crawlear 891 millones de páginas y fue integrado en *AltaVista* en 2001.
- En el año 2000 BrightPlanet sugirió que el contenido web de la deep web (páginas no indexadas por los sistemas de búsqueda) era 500 veces más grande que la web de la superficie. Las aplicaciones web de la deep web necesitaban muchas veces de interacciones del usuario por lo que supuso un nuevo problema.

- En el año 2007 He et al, de manera aleatoria accedió a un millón de IPs y las recorrió buscando deep webs mediante elementos de los formularios HTML. El experimento fue un éxito ya que 126 webs fueron encontradas y se llegaron a acceder hasta 190 bases de datos dando los resultados un 99% en el intervalo de confianza. Mediante esta técnica se estima que Google y Yahoo han visitado sólo el 32% de la deep web. En la segunda generación de web crawlers esto se tuvo en cuenta. Para obtener información oculta en la deep web el web crawler programó para llenar con diferentes conjuntos de datos los formularios y enviarlos varias veces. El problema de crawlear la deep web se redujo entonces a asignar los valores adecuados a los campos de los formularios.
- Para crawlear las RIA (Rich Internet Applications), fueron necesarias técnicas que tenían en cuenta la estructura web DOM (Modelo de Objetos del Documento) para determinar si era la primera vez que un crawler visitaba una página. Mediante un extractor de eventos JavaScript se simulaban diversas estrategias para abarcar todos los estados posibles.
- A partir del 2007 y hasta la actualidad se han introducido diversas técnicas y estrategias para abarcar el máximo conjunto de estados de cada web mediante el uso de eventos entre cliente-servidor. Debido a que la tecnología web evoluciona rápidamente las técnicas van cambiando al ritmo de la tecnología web. [4]

1.3.3 ESTADO ACTUAL

Info Japón hace uso de la tecnología web crawling. Bebiendo de los principios de inteligencia artificial, el proyecto presentado se basa en la programación de agentes inteligentes supervisados que, mediante técnicas de web crawling como la exploración del DOM, las consultas XPATH utilizadas en las RIA (Rich Internet Applications) y el autocompletado de formularios permiten la obtención de información relevante para el usuario final.

1.4 ALCANCE Y LIMITACIONES

1.4.1 ARQUITECTURA SOFTWARE PROPUESTA

El sistema propuesto se basa en el modelo de arquitectura Model-View-Controller (MVC).

- La lógica del sistema, las operaciones y las funciones que hacen referencia al modelo del sistema está alojada en el servidor y se encargará de llevar a cargo los diferentes procesos de obtención, transformación y carga de la información (procesos ETL) llevados a cabo por web crawlers. En ningún momento el sistema guardará ni transformará los datos obtenidos excepto en el caso del proceso ETL referente a los artículos.
- El motor de comunicación o controlador que se encargará del secuencia de datos entre los procesos del cliente y los procesos del servidor será una API propia de carácter privado.
- Los clientes que se comunicaran con el controlador serán tanto la aplicación web como la aplicación móvil donde se alojaran las vistas.

En ningún momento el sistema almacenará información obtenida mediante los bots exceptuando la información referente a los artículos y blogs necesaria para el proceso ETL relacionado con el mismo.

1.4.1.1 MODELO – PROCESOS DE LOS WEB CRAWLERS

El sistema obtendrá información de la red mediante el uso de procesos web crawlers (bots) que se encargaran de recopilar información de enlaces previamente especificados por el cliente y mostrarla en formato de intercambio de datos JSON (JavaScript Object Notation - Notación de Objetos de JavaScript).

Los procesos de crawling son los siguientes:

- Crawling de estudios

Se encargará de recopilar información referente a universidades, colegios de posgrado, colegios de formación profesional y escuelas de idioma japonés. Se identifican los siguientes subprocesos asociados al crawling de estudios:

- Búsqueda de universidades
 - Obtención de lista de universidades
 - Obtención de información de universidad específica
 - Obtención de información de departamento de universidad
 - Información de acceso al departamento
 - Información de admisiones del departamento
 - Información de instalaciones del departamento
 - Información de ayuda al estudiante del departamento
 - Información básica del departamento
- Búsqueda de escuelas de posgrado
 - Obtención de lista de colegios de posgrado
 - Obtención de información de colegio de posgrado específico
 - Obtención de información de departamento de colegio de posgrado
 - Información de acceso al departamento
 - Información de admisiones del departamento
 - Información de instalaciones del departamento
 - Información de ayuda al estudiante del departamento
 - Información básica del departamento
- Búsqueda de escuelas de formación profesional
 - Obtención de lista de colegios de formación profesional
 - Obtención de información de colegio de formación profesional específico
 - Obtención de información de departamento de colegio de formación profesional
 - Información de acceso al departamento
 - Información de admisiones del departamento
 - Información de instalaciones del departamento
 - Información de ayuda al estudiante del departamento

- Información básica del departamento
- Búsqueda de escuelas de idioma Japonés
 - Obtención de lista de escuelas de idioma Japonés
 - Obtención de información de escuela de idioma Japonés
- Crawling de ofertas laborales
 - Obtención de lista con ofertas de trabajo
- Crawling de viviendas
 - Obtención de lista de viviendas

Cada proceso de crawling será llamado por el cliente dados unos parámetros estáticos gestionados por la API. La información referente a los parámetros estáticos aceptados por la API (controlador) se encuentra en la documentación de la misma accesible desde el anexo.

1.4.1.2 MODELO - PROCESO DE OBTENCIÓN DE ARTÍCULOS

El sistema proveerá de un directorio de artículos ordenado cronológicamente. El proceso ETL contemplará los siguientes subprocessos:

- Proceso de extracción de artículos, transformación y carga generando como salida un fichero contenedor JSON por cada blog establecido por el cliente y un fichero contenedor maestro JSON que contendrá todos los artículos de todos los blogs establecidos por el cliente ordenados por orden cronológico en orden descendente.
- Proceso de indexación en base de datos de documentos. Se encargará de indexar los ficheros contenedores JSON generados en una base de datos de documentos.
- Proceso de consultas. Se encargará de dotar a la API de la capacidad de buscar en los documentos de la base de datos de documentos dados los parámetros estáticos definidos en la documentación de la API.

1.4.1.3 MODELO - TESTS UNITARIOS

El sistema contendrá una batería de test desarrollados en el plan de pruebas que se encargarán de comprobar el correcto funcionamiento de los diferentes crawlers.

Como consideración al alcance es importante remarcar que no se cubrirán los test de los procesos de crawling relacionados con viviendas y trabajo dejándolo para un desarrollo futuro.

1.4.1.4 CONTROLADOR - API REST

El controlador del sistema como bien se ha especificado en la arquitectura propuesta será el encargado de la comunicación entre la parte lógica del sistema (Back-end o modelo) y la parte relacionada a las vistas del sistema (Front-end o vistas). Este secuencia de comunicación se llevará a cabo mediante una API REST de carácter privado.

La API REST de Info Japón se encargará de recibir peticiones HTTP por parte de los clientes y, de manera concurrente, gestionará las mismas enlazándolas con las funciones definidas en la petición. El conjunto de peticiones disponibles, así como los path de las funciones y los parámetros estáticos necesarios para generar la petición correctamente se encuentran explicados en la documentación de la API REST de Punto Japón accesible desde el anexo.

Los clientes que accederán a la API serán tanto el cliente web como el cliente móvil desarrollados para Punto Japón. Si bien la API quedará abierta a terceros será necesario solicitar acceso a la misma a los administradores de Punto Japón.

La estructura del controlador, así como los test referentes al mismo se encuentran explicados en el desarrollo del problema.

1.4.1.5 API CONTROLADOR – PLAN DE PRUEBAS

Se desarrollará un plan de pruebas de petición – respuesta contra la API REST mediante el uso de Apache JMeter.

1.4.1.6 VISTA – APPLICACIÓN WEB Y APPLICACIÓN MÓVIL

Para la visualización de los datos producidos por los diferentes procesos lógicos y funcionales del sistema se dará uso de un subsistema desarrollado con este objetivo. Este subsistema programado en HTML5 y Javascript de manera responsive (multidispositivo) permitirá a los usuarios interactuar con las diferentes funciones del sistema principal desde la web.

La aplicación web cubrirá por tanto los siguientes puntos:

- Buscador de estudios
 - Llamadas a funciones definidas en la API REST referentes a estudios
- Buscador de vivienda
 - Llamadas a funciones definidas en la API REST referentes a viviendas
- Buscador de trabajo
 - Llamadas a funciones definidas en la API REST referentes a ofertas de trabajo
- Listado y búsqueda en artículos
 - Llamadas a funciones definidas en la API REST referentes a los artículos.

Como consideración al alcance no habrá cobertura de pruebas para los módulos referentes a la aplicación web y a la aplicación móvil.

1.4.2 ARQUITECTURA HARDWARE PROPUESTA

El sistema se dividirá en dos entornos: entorno de pruebas y entorno de producción. Ambos entornos serán independientes y alojarán tanto la API REST de Punto Japón como la aplicación web. El entorno de pruebas será el entorno dedicado a la modificación de código y testing donde fluirán las diferentes versiones del sistema. El entorno de producción contendrá solo la última versión reléase pertinente. Ambos entornos estarán conectados al repositorio de versión del código pero solo el entorno de producción implementará la rama release una vez haya pasado las pruebas de manera satisfactoria.

Ambos entornos estarán alojados en un SO Linux que contará con los paquetes necesarios para el correcto funcionamiento de la API. No se implementará ningún sistema de monitorización del servidor más allá de el del propio proveedor.

1.4.3 ALCANCE DE LAS MÉTRICAS

El alcance de las métricas será el reflejado en el plan de métricas. Las mismas serán usadas para mejorar la experiencia del usuario en el sistema en desarrollos futuros.

1.5 RIESGOS

En todo proyecto software es de vital importancia la prevención y reacción frente a riesgos que la aplicación va a contemplar. De esta manera se pretende ofrecer una experiencia final de calidad. Para ello se genera un plan de riesgos que contempla los riesgos a los que se enfrenta el proyecto debido a su naturaleza, desarrollo y gestión.

1.5.1 PLAN DE RIESGOS Y MITIGACIONES

El plan de riesgos y mitigaciones abarca todo evento que ponga en peligro el cumplimiento de los objetivos marcados dentro del proyecto. La probabilidad refleja, en una escala del 1 al 5 la frecuencia de aparición del riesgo mientras que el impacto refleja, también en una escala del 1 al 5, el impacto que el riesgo tendrá sobre el proyecto.

Todo riesgo tiene que tener un plan de mitigación, sino es un hecho que no se puede evitar por lo que daría lugar a la cancelación del proyecto.

#	Riesgo	Plan de mitigación o contingencia	Probabilidad	Impacto
1	Web origen para la extracción de información inaccesible	Proporcionar fuentes diferentes para la obtención de datos	1	5
2	Caída del servidor por problemas técnicos	Proporcionar acceso al servicio redireccionando el tráfico al servicio desplegado en el servidor de pruebas	1	4
3	Incumplimiento de políticas de privacidad por la extracción de información de una web específica	Proporcionar fuentes diferentes para la obtención de datos	1	5
4	La infraestructura hardware no tiene potencia para atender todas las peticiones	Escalar el servidor virtualizado sin necesidad de volver a hacer ningún despliegue	2	2
5	Fallo de algún módulo del servicio	Deshabilitar temporalmente el módulo. Comprobarlo y rehacer batería de pruebas	2	3

Tabla 1 - Plan de riesgos y mitigaciones

1.5.2 MATRIZ DE IMPACTO

La matriz de impacto refleja de manera visual el impacto que cada riesgo tendría sobre el sistema en base a su probabilidad de aparición y a la gravedad del mismo. Se pueden diferenciar diferentes colores en base al impacto final:

- Impacto blanco: El impacto que tendrá el riesgo sobre el sistema es completamente asumible y no provocará que el sistema deje de funcionar. El sistema seguirá funcionando.
- Impacto amarillo: El impacto que tendrá el riesgo sobre el sistema es medianamente asumible pero puede generar fallos o errores en algún módulo del sistema. El sistema seguirá funcionando.
- Impacto naranja: El impacto que tendrá el riesgo sobre el sistema es difícilmente asumible pero generará fallos o errores en el sistema. El sistema podría no seguir funcionando hasta cubrir el daño.
- Impacto rojo: El impacto que tendrá el riesgo sobre el sistema no es asumible ya que los fallos o errores que generaría en el sistema lo dejarían inutilizado. Se valoraría la continuación del proyecto.

		GRAVEDAD (IMPACTO)				
		MUY BAJO 1	BAJO 2	MEDIO 3	ALTO 4	MUY ALTO 5
APARICIÓN (probabilidad)	MUY ALTA 5	#1, #3				
	ALTA 4	#2				
	MEDIA 3		#5			
	BAJA 2		#4			
	MUY BAJA 1					

Tabla 2 - Matriz de impacto

1.6 KEY PERFORMANCE INDICATOR (KPI)

Los KPI son valores medibles que permiten determinar el rendimiento, eficiencia y calidad de una aplicación software mediante las métricas de la misma.

Los KPI por tanto evaluarán como de satisfactoriamente se cumplirán los objetivos de la aplicación.

Objetivo	KPI	Métricas de apoyo
a) Acceso sobre artículos	Grado de interés y eficiencia del módulo de artículos	Número de llamadas a mostrar artículos, tiempo promedio en página principal, número de búsquedas sobre artículos
b) Búsquedas de estudios	Grado de interés y eficiencia del módulo de estudios	Número de llamadas a la función de búsqueda de estudios de la API
c) Búsqueda de trabajo	Grado de interés y eficiencia del módulo de trabajo	Número de llamadas a la función de búsqueda de empleo de la API
d) Búsqueda de viviendas	Grado de interés y eficiencia del módulo de viviendas	Número de llamadas a la función de búsqueda de viviendas de la API
e) Satisfacción del cliente móvil	Grado de satisfacción del cliente móvil	Rating de la aplicación en Google Play
f) Satisfacción del cliente web	Grado de satisfacción del cliente web	Número de usuarios nuevos, número de usuario de rebote, número de usuarios de retorno
g) Eficiencia global de la aplicación	Grado de eficiencia de la aplicación	Número de registros con errores

Tabla 3 - Key Performance Indicators

1.7 DISPOSICIONES LEGALES

Antes de extraer información de una página web es necesario determinar si va o no en contra de las políticas y condiciones de uso de la misma. Como ya se ha visto, mediante estrategias de web crawling es posible acceder a datos ocultos de las páginas web que muy seguramente el propietario de la página web no desea que se accedan.

Por ello las páginas web poseen términos y condiciones que deben entenderse y aceptarse antes de extraer información de las mismas. En el caso de que no se especificara ningún término prohibiendo expresamente la extracción de información, se referenciará la fuente de la cual se ha extraído la información en la visualización de la misma como norma ética.

En la actualidad se puede extraer información de las páginas utilizando diversas estrategias ya que no hay un grado de protección estándar establecido. Por ello el desarrollador debe actuar con ética respetando las condiciones de uso. En caso de duda de que se puede o que no se puede crawlear, las páginas poseen un fichero denominado robots.txt donde define expresamente las páginas del directorio web a las que pueden acceder los robots y las páginas a las que está prohibido el acceso además de definir una lista negra de bots que tienen prohibido el crawling en esa web. Si este fichero permite los robots para obtener información de su sitio (siempre que no sea una web de cualquier gobierno), se entiende entonces que implícitamente se acepta la extracción de la información en los directorios especificados. [5]

Ejemplo: www.jpss.jp/robots.txt

User-Agent: *

Allow: /

Sitemap: http://www.jpss.jp/sitemap_index.xml

1.8 TECNOLOGÍAS Y SOFTWARE UTILIZADO

InfoJapón utiliza un conjunto de tecnologías y herramientas software de diversa naturaleza en el proyecto ya que abarca tanto la parte de desarrollo en el lado del servidor, como en el lado del cliente como el entorno de sistemas del proyecto.

1.8.1 FORMATOS DE INTERCAMBIO DE DATOS

En la actualidad hay muchísimos servicios y aplicaciones que comparten información. La representación de los datos que se comparten entre aplicaciones es un punto crítico en la funcionalidad de las mismas. Por ello la definición de estándares de representación de datos permiten envolver la información en formatos robustos que permiten transmitir información compleja entre diferentes sistemas. [6]

1.8.1.1 XML

Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Nació como estándar para el intercambio de información estructurada entre diferentes plataformas.

Posee un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir información de una manera segura, fiable y fácil.

Para la correcta estructura de un documento XML se cuenta con la Document Type Definition o DTD que define los tipos de elementos, atributos y entidades permitidas y que puede expresar algunas limitaciones para combinarlos. Los documentos XML que se ajustan a su DTD son denominados válidos.

Una de las fortalezas de XML es el soporte Unicode que permite escribir la información en cualquier idioma del mundo.

Como defecto, para documentos muy grandes suele requerir un uso más intensivo de memoria y procesador.

1.8.1.2 JSON

JSON, acrónimo de JavaScript Object Notation, es un estándar abierto que utiliza texto plano para codificar información en la forma de atributo:valor.

La ventaja del uso de JSON es que es un objeto válido JavaScript por lo que es el formato perfecto para ese lenguaje aunque es empleado con mucha frecuencia en otros lenguajes como Python. La mayoría de navegadores incluyen funciones nativas para codificar y decodificar JSON lo que le aumenta su desempeño y seguridad.

Como desventajas , dependiendo del entorno a veces se hace necesario que se incorporen funciones u objetos analizadores de JSON. También un tema a considerar es el Unicode ya que representa variaciones en las distintas implementaciones y a veces puede representar un riesgo de seguridad por la facilidad de ejecución en entornos basados en javascript.

1.8.1.3 YAML

YAML, "YAML Ain't Another Markup Language (en castellano, "YAML no es otro lenguaje de marcado"), surge con el objetivo de facilitar el mapeo de estructuras de datos complejas en un documento de texto plano legible.

Es más estricto que XML y JSON pero a su vez más simple. La simplicidad le otorga velocidad pero, a diferencia de JSON, no es usado para servicios web o APIs REST sino para archivos de configuración, depuración u otros fines.

Es ampliamente utilizado por el framework Ruby On Rails basado en ruby.

1.8.2 PROCESOS ETL (EXTRACCIÓN – TRANSFORMACIÓN – CARGA)

Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en un sistema de análisis u otro sistema operacional para apoyar un proceso de negocio. [7]

1.8.2.1 EXTRACCIÓN

La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. Una parte intrínseca del proceso de extracción es la de analizar los datos extraídos para verificar si cumplen la pauta o la estructura que se esperaba. De no ser así los datos son rechazados.

Un requerimiento importante exigible a la tarea de extracción es que cause un impacto mínimo en el sistema de origen debido a los problemas que podría ocasionar al mismo si los datos a extraer son muchos.

1.8.2.2 TRANSFORMACIÓN

La fase de transformación aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados en otro sistema. Para que los datos se ajusten a los procesos de negocio es necesario aplicar un conjunto de transformaciones a los mismos para que el sistema que los reciba pueda interpretarlos de manera correcta y así optimizar las salidas. Algunas transformaciones pueden ser las siguientes:

- Descartar datos para su carga (como por ejemplo datos nulos)
- Traducción de códigos para que encajen con el proceso de negocio
- Codificación de valores libres u obtención de cálculos en base a datos
- Unión de datos de múltiples fuentes (búsquedas, combinaciones...)
- Generación de campos clave en el destino
- División o modificación de datos para que se ajusten al formato deseado
- Validación de datos tras transformaciones

1.8.2.3 CARGA

La fase de carga es el momento en el cual los datos de la fase anterior son cargados en el sistema destino. Dependiendo de los requerimientos este proceso puede variar.

1.8.3 KETTLE (PENTAHO DATA INTEGRATION)

Kettle es una herramienta de integración de datos con licencia opensource que permite generar procesos ETL de manera gráfica y de una forma intuitiva, escalable y con una arquitectura basada en estándares.

Kettle funciona mediante la implementación de librerías de terceros por lo que eventualmente aumenta su funcionalidad. Posee una capa gráfica que permite procesos ETL de manera gráfica arrastrando y soltando componentes. Por debajo cada componente se corresponde a funciones específicas programadas en Java por lo que permite exportar los procesos a cualquier sistema. Está orientado a metadatos y puede utilizarse como aplicación independiente. Soporta grandes cantidades de formatos de entrada y salida. [8]

1.8.4 BASES DE DATOS NOSQL

En informática, NoSQL es una amplia clase de sistemas de gestión de bases de datos con características diferentes al modelo tradicional, la más destacada es que no usan SQL como el principal lenguaje de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN ni garantizan completamente la atomicidad, consistencia, aislamiento y durabilidad de los datos. Habitualmente escalan horizontalmente y presentan mejoras con respecto a las bases de datos relacionales (SQL) en la obtención de un mejor rendimiento en las operaciones de recuperación y agregación de registros (clave – valor). La pérdida de flexibilidad en tiempo de ejecución se ve compensada por ganancias en escalabilidad y rendimiento al tratar ciertos modelos de datos. [9]

1.8.4.1 ELASTIC SEARCH

ElasticSearch es un gestor de bases de datos NoSQL que permite realizar búsquedas y análisis sobre documentos en tiempo real. Entre las diferentes ventajas destacan las siguientes:

- **Datos en tiempo real:** Todos los datos están disponibles en tiempo real para realizar búsquedas y análisis
- **Análisis avanzados en tiempo real**
- **Distribución masiva:** Permite una escalabilidad horizontal de manera sencilla simplemente añadiendo nodos al cluster. El propio cluster toma gestión automáticamente la replicación entre los nodos.
- **Alta disponibilidad:** El cluster detecta tanto los nuevos nodos como los nodos defectuosos y se encarga de reorganizar y re-balancear los datos de manera automática para asegurar que los datos están a salvo y accesibles en todo momento.
- **Orientado a documentos**
- **Libre de esquema:** Elasticsearch detecta automáticamente la estructura de datos y tipos creando así un índice y haciendo que los datos de los documentos consten en las búsquedas.
- **RESTful API:** Cualquier acción puede ser llevada a cabo utilizando una simple RESTful API utilizando JSON sobre peticiones HTTP.
- Está disponible bajo **licencia Apache2 Open Source** por lo que puede ser descargado, utilizado y modificado de manera gratuita. [10]

1.8.5 KIBANA

Kibana es una herramienta software de los creadores de Elastic Search que permite la exportación de datos de diferentes fuentes (Logstash, Hadoop, Beats, Apache Flume...) para un posterior análisis inteligente incluso mediante la implementación de transformaciones matemáticas y extracción de datos específicos del conjunto de datos para posteriormente visualizarlo en gráficas. [11]

1.8.6 LOGSTASH

Logstash es una herramienta software de los creadores de Elastic Search que permite procesar ficheros de registro (logs) y otros datos de una gran cantidad de sistemas. De esta manera permite, en base a un esquema definido, enviar los datos en tiempo real de los diferentes logs a sistemas centralizados como Elastic Search. [12]

1.8.7 GIT Y BITBUCKET

GIT es una herramienta software de control de versiones que permite gestionar las diferentes versiones y etapas del desarrollo de un software en repositorios distribuidos. De esta manera existe un repositorio central donde el código está alojado y luego un repositorio local para todo aquel que quiera una copia del código. Una vez validado el código en el repositorio local es posible subirlo al centralizado para que todos los miembros del proyecto tengan acceso al mismo.

Por tanto GIT es una herramienta imprescindible en el desarrollo de proyectos software ya que permite llevar un seguimiento de los diferentes desarrollos de la aplicación derivando en un aumento de productividad, eficiencia y rapidez en la gestión del código.

Bitbucket es un servicio de alojamiento web que ofrece planes SaaS (Software as a Service) basados en Mercurial y Git, ambas herramientas de control de versiones. De esta manera, el plan gratuito de Bitbucket permite tener ilimitados repositorios privados en la nube que pueden integrarse con herramientas de gestión de proyectos de Bitbucket como Jira (de pago).

1.8.8 API REST

Una API REST es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otros sistemas como una capa de abstracción cumpliendo el estilo de arquitectura software para sistemas hipermedia distribuidos REST (Transferencia de Estado Representacional). [13]

El término REST se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP. Se podría considerar como un framework para construir aplicaciones web respetando el protocolo HTTP por tanto es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a internet.

Existen tres niveles de calidad a la hora de aplicar REST en el desarrollo de APIs:

1. **Uso correcto de URLs:** Las URL, Uniform Resource Locator , son un tipo de URI, Uniform Resource Identifier, que además de permitir identificar de forma única el recurso, nos permite localizarlo para poder acceder a él o compartir su ubicación. Existen varias reglas básicas para nombrar la URI de un recurso:
 - a. El nombre de URI no debe implicar una acción por tanto debe evitar utilizar verbos en ellos
 - b. Deben ser únicas, no se debe tener más de una URI para identificar un mismo recurso
 - c. Deben ser independientes de formato
 - d. Deben mantener una jerarquía lógica
 - e. Los filtrados de información de un recurso no se hacen en la URI
2. **HTTP:** Para desarrollar APIs REST los aspectos clave son los siguientes:
 - a. Métodos HTTP: Para manipular recursos, HTTP dota de los siguientes métodos sobre los que se debe operar:
 - i. GET: Para consultar y leer recursos
 - ii. POST: Para crear recursos
 - iii. PUT: Para editar recursos
 - iv. DELETE: Para eliminar recursos
 - v. PATCH: Para editar partes concretas de un recurso
 - b. Códigos de estado
 - c. Aceptación de tipos de contenido
3. **Hypermedia:** tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional. [14]

1.8.9 APPLICACIONES DE INTERNET ENRIQUECIDAS

Una rich Internet application (RIA), "aplicación de Internet enriquecida" o "aplicación rica de internet" (ARI), es una aplicación web que tiene la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

Las RIA surgen como una combinación de las ventajas que ofrecen las aplicaciones web y las aplicaciones tradicionales. Buscan mejorar la experiencia y productividad del usuario.

Normalmente en las aplicaciones web, hay una recarga continua de páginas web cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces a recargar la misma página con un cambio mínimo.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y solo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una base de datos o de otros ficheros externos. [15]

1.8.10 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

JavaScript es el Lenguaje de programación (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante peticiones HTTP para obtener contenido formateado en formatos como XML o JSON. [16]

1.9 PLAN DE TRABAJO

Fase	Horas	Días
Estudio Previo	16h	
Definición de necesidades	2h	1d
Definición tecnologías a utilizar	3h	1d
Fase de aprendizaje del previo del conjunto de tecnologías y herramientas elegidas	5h	2d
Análisis preliminar de requisitos	2h	1d
Análisis en profundidad de requisitos funcionales	2h	1d
Análisis en profundidad de requisitos no funcionales	2h	1d
DISEÑO FUNCIONAL	90h	
Diseño de casos de uso	18h	3d
Diseño de diagramas asociados a los casos de uso	20h	3d
Diseño preliminar del diagrama de clases	8h	2d
Diseño preliminar del diagrama de despliegue	8h	2d
Análisis de páginas para la extracción de información	36h	6d
DESARROLLO	264h	
Creación de entorno de trabajo para el desarrollo	22h	4d
Desarrollo del proceso ETL de artículos en Kettle	8h	2d
Desarrollo de clases de apoyo	4h	1d
Desarrollo del modelo: Clases abstractas e interfaces de los crawlers de estudios	9h	1d
Desarrollo del modelo: Clases de los crawlers de estudios	40h	6d
Desarrollo del modelo: Desarrollo de los test de los crawlers de estudios	10h	1d

Desarrollo del modelo: Clases del crawler de colegios de japonés	34h	5d
Desarrollo del modelo: Clases del crawler de trabajo	25h	4d
Desarrollo del modelo: Clases del crawler de viviendas	26h	4d
Desarrollo del modelo: Clases asociadas al proceso ETL de artículos en Kettle	16h	2d
Desarrollo del controlador: Funciones asociadas a los recursos y API REST	20h	2d
Desarrollo del frontend: Desarrollo de la parte de diseño	12h	2d
Desarrollo del frontend: Desarrollo de funcionalidades javascript	34h	6d
Desarrollo del frontend: Desarrollo funcionalidades contra errores	4h	1d
Pruebas de usuario	10h	
Batería de pruebas de rendimiento y API REST en servidor local	6h	1d
Batería de pruebas del módulo de estudios	4h	1d
Producción	59h	
Contratación servidor virtual	1h	1d
Securización del servidor	2h	1d
Instalación y configuración de paquetes esenciales	8h	1d
Configuración de Spring-boot y maven	23h	3d
Configuración de ElasticSearch	5h	1d
Configuración de logstash y Kibana	4h	1d
Despliegue de aplicación backend	8h	1d
Despliegue de aplicación frontend	8h	1d
Mantenimiento y documentación	30h	

Documentación de memoria	22h	3d
Mantenimiento de servidor y monitorización de métricas	8h	1d
Total:	469h	80d

Tabla 4 - Plan de trabajo

Capítulo 2. ANÁLISIS DE REQUISITOS

2.1 REQUISITOS FUNCIONALES

Los requisitos funcionales son el conjunto de servicios que el sistema deberá proveer dadas un conjunto de entradas para generar unas salidas determinadas.

ID	DESCRIPCIÓN	PRIORIDAD	IMPL.
RQ01	La solución permitirá la visualización de artículos en orden cronológico.	1	SÍ
RQ02	La solución permitirá la búsqueda sobre artículos en base a una entrada determinada.	1	SÍ
GR01-01	<p>La solución permitirá la búsqueda y visualización de información referente a universidades teniendo en cuenta los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • RQ01-01-01. Búsqueda por prefectura. • RQ01-01-02. Búsqueda por nombre. • RQ01-01-03. Búsqueda por especialidad. • RQ01-01-04. Búsqueda por mes de admisión. • RQ01-01-05. Búsqueda por periodo de admisión de aplicaciones. • RQ01-01-06. Búsqueda en referencia a examen de entrada (EJU). • RQ01-01-07. Búsqueda por tipo de universidad. • RQ01-01-08. Búsqueda por fecha límite de aplicación. • RQ01-01-09. Búsqueda en referencia a certificación de inglés necesaria. • RQ01-01-10. Visualizar información de la universidad • RQ01-01-11. Visualizar información del departamento universitario 	1	SÍ
GR01-02	<p>La solución permitirá la búsqueda y visualización de información referente a colegios de posgrado teniendo en cuenta los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • RQ01-02-01. Búsqueda por prefectura. • RQ01-02-02. Búsqueda por nombre. • RQ01-02-03. Búsqueda por especialidad. • RQ01-02-04. Búsqueda por tipo de escuela. • RQ01-02-05. Búsqueda en referencia al idioma hablado en la clase. • RQ01-02-06. Visualizar información de colegio de posgrado • RQ01-02-07. Visualizar información del 	1	SÍ

	departamento del colegio de posgrado		
GR01-03	<p>La solución permitirá la búsqueda y visualización de información referente a colegios de formación profesional teniendo en cuenta los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • RQ01-03-01. Búsqueda por prefectura. • RQ01-03-02. Búsqueda por nombre. • RQ01-03-03. Visualizar información del colegio de formación profesional • RQ01-03-04. Visualizar información del departamento de colegio de formación profesional 	1	Sí
GR01-04	<p>La solución permitirá la búsqueda y visualización de información referente a colegios de japonés teniendo en cuenta los siguientes parámetros de búsqueda si proceden:</p> <ul style="list-style-type: none"> • RQ01-04-01. Búsqueda por prefectura • RQ01-04-02. Búsqueda por barrio • RQ01-04-03. Visualizar información de colegio de japonés 	1	Sí
RQ03	<p>La solución permitirá la búsqueda de ofertas de empleo en base a los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • RQ03-01. Búsqueda por localización • RQ03-02. Búsqueda por especialidad 	1	Sí
RQ04	<p>La solución permitirá la búsqueda de viviendas en base a los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • RQ04-01. Búsqueda por prefectura • RQ04-02. Búsqueda por alquiler mínimo • RQ04-03. Búsqueda por alquiler máximo • RQ04-04. Búsqueda por dimensión • RQ04-05. Búsqueda por nº de habitaciones • RQ04-06. Búsqueda por tipo de construcción • RQ04-07. Búsqueda en base a la distancia a la estación más cercana • RQ04-08. Búsqueda en base a la antigüedad de la vivienda. • RQ04-09. Búsqueda en base a los bajos costes iniciales. • RQ04-10. Búsqueda en base al pago de una señal inicial. • RQ04-11. Búsqueda en base al depósito • RQ04-12. Búsqueda en base a los gastos de la 	1	Sí

	<p>agencia</p> <ul style="list-style-type: none"> • RQ04-13. Búsqueda en base al servicio de internet en la vivienda • RQ04-14. Búsqueda en base a la disponibilidad de pago con tarjeta. • RQ04-15. Búsqueda en base a la existencia del aval • RQ04-16. Búsqueda en base a la admisión de mascotas en la vivienda • RQ04-17. Búsqueda en base al periodo de estancia. • RQ04-18. Búsqueda en base a si la vivienda está amueblada o no. • RQ04-19. Búsqueda en base a la disponibilidad de wifi en la vivienda 		
--	---	--	--

Tabla 5 - Requisitos funcionales

2.2 REQUISITOS NO FUNCIONALES

Los requisitos funcionales son el conjunto de indicaciones preceptivas que marcarán el cómo el cómo la solución funcionará.

ID	DESCRIPCIÓN	PRIORIDAD	IMPL.
GRNF01	<p>La solución deberá tener en cuenta los siguientes requisitos de rendimiento:</p> <ul style="list-style-type: none"> • RQNF01-01. El rendimiento en peticiones HTTP al cliente web debe ser inferior a 4/sec por cada 10 muestras de peticiones a cada página del cliente web. • RQNF01-02. Gestión de reportes mensuales en base al rendimiento de peticiones a la API REST mediante JMeter. 	2	Sí
GRNF02	<p>La solución deberá tener en cuenta los siguientes requisitos de comunicaciones:</p> <ul style="list-style-type: none"> • RQNF02-01. El secuencia de información cliente-servidor debe ser mediante ficheros JSON. • RQNF02-02. Las peticiones a la API REST deben ser mediante el protocolo HTTP cumpliendo con los requisitos establecidos en la misma API dependiendo de la función 	1	Sí
GRNF03	<p>La solución deberá tener en cuenta los siguientes requisitos relativos a la interfaz de usuario:</p> <ul style="list-style-type: none"> • RQNF03-01. Las interfaces del cliente web 	3	Sí

	<p>deben tener un diseño responsive y escalable.</p> <ul style="list-style-type: none"> • RQNF03-02. Se requerirá un diseño minimalista que permita al usuario una experiencia sencilla e intuitiva 		
GRNF04	<p>La solución deberá tener en cuenta los siguientes requisitos relativos al entorno de desarrollo:</p> <ul style="list-style-type: none"> • RQNF04-01. La información de los artículos será gestionada en formato JSON en la base de datos documental Elastic Search. • RQNF04-02. El servidor web donde que ejecutará la API REST deberá ser Tomcat mediante Spring boot (en referencia al framework Spring) • RQNF04-03. El servidor web que ejecutará el cliente web será Apache. • RQNF04-04. Los únicos puertos abiertos serán los referentes a los servicios listados anteriormente y los mínimos necesarios para la administración remota del servidor • RQNF04-05. Las actualizaciones de la solución se harán mediante el software de control de versiones git utilizando el repositorio indicado para el entorno de desarrollo. 	3	SÍ
GRNF05	<p>La solución deberá tener en cuenta los siguientes requisitos de portabilidad:</p> <ul style="list-style-type: none"> • RQNF05-01. El cliente web deberá ser visualizable en entornos desktop y móviles. • RQNF05-02. El cliente app deberá poder ser instalable mediante la Google Store y podrá ser ejecutado en clientes con Android 4.0 o superior. 	3	SÍ
GRNF06	<p>La solución deberá tener en cuenta los siguientes requisitos de concurrencia:</p> <ul style="list-style-type: none"> • RQNF06-01. Las peticiones a la API REST deberán ser concurrentes soportando varias peticiones a la vez. • RQNF06-02. Las peticiones al cliente web deberán ser concurrentes soportando varias peticiones a la vez. 	3	SÍ

GRNF07	<p>La solución deberá tener en cuenta los siguientes requisitos de seguridad:</p> <ul style="list-style-type: none"> • RQNF07-01. Tanto el entorno de pruebas como el entorno de desarrollo deberán implantar un firewall de seguridad que solo permita al mínimo número de puertos necesario para que el servicio y la administración remota funcionen. • RQNF07-02. Tanto la API REST como el cliente web deberán estar protegidos contra ataques DDoS que puedan intervenir en el correcto funcionamiento del servicio. • RQNF07-03. Será necesario establecer un plan de riesgos y contingencias que permita que el servicio continúe en caso de fallos técnicos. 	3	Sí
GRNF08	<p>La solución deberá tener en cuenta los siguientes requisitos de documentación:</p> <ul style="list-style-type: none"> • RQNF08-01. Será necesario documentar las funcionalidades de la API REST • RQNF08-02. Será necesario documentar la configuración del entorno de desarrollo 	3	Sí

Tabla 6 - Requisitos no funcionales

Capítulo 3. DISEÑO

3.1 CASOS DE USO

Los casos de uso abarcan el conjunto de actividades que cubren los requisitos especificados.

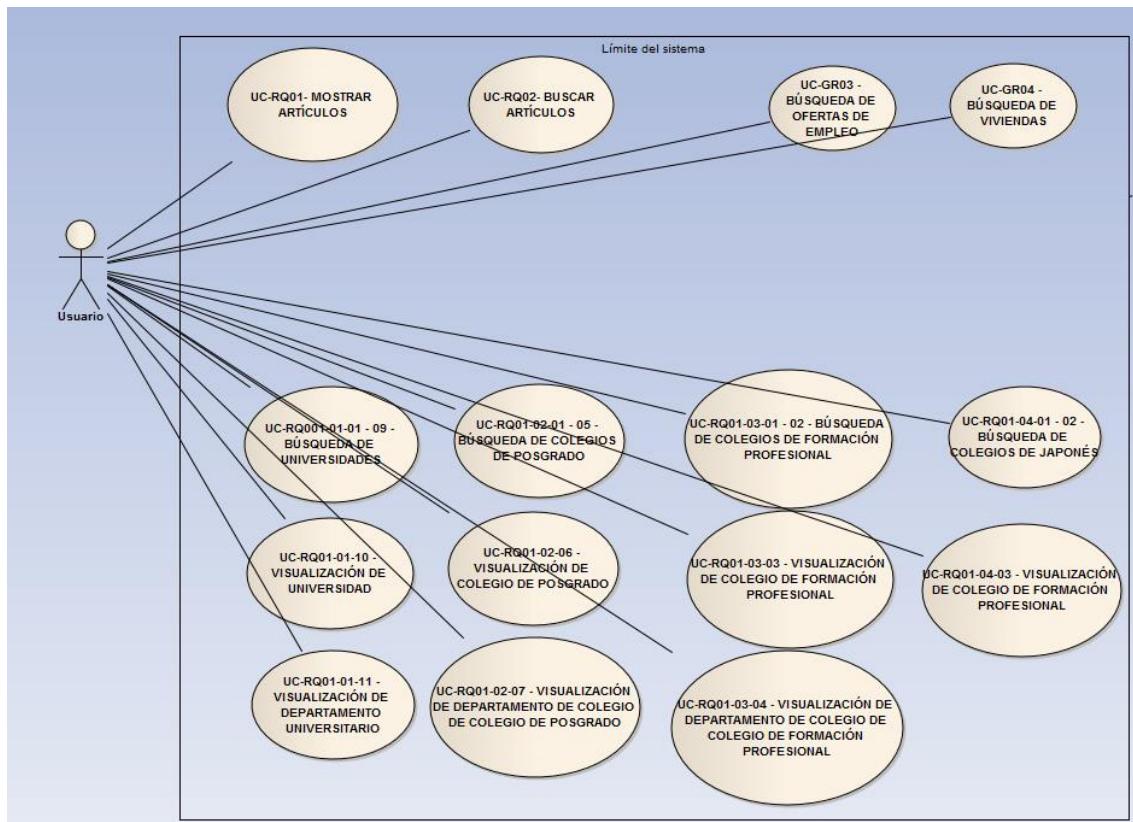


Figura 1 - Diagrama de casos de uso

3.1.1 RQ-01. MOSTRAR ARTÍCULOS

UC-RQ01	MOSTRAR ARTÍCULOS	
Objetivos asociados	OBJ-02. Servir de directorio de artículos de blogs de emigrantes hispanohablantes en Japón	
Requisitos asociados	RQ01. La solución permitirá la visualización de artículos en orden cronológico.	
Descripción	La solución provee al usuario de los artículos ordenados cronológicamente de los blogs asociados al proyecto.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El usuario accede a la página principal de la aplicación web o móvil o hace scroll en la página principal superando el offset del marco.
	2	El cliente envía una petición HTTP asíncrona al controlador

		REST para obtener el listado de artículos
3		El controlador ejecuta la función del modelo asociada al recurso accedido (mostrar artículos).
4		El modelo carga la información del fichero alojado en el servidor sort.json que contiene todos los artículos en formato json ordenados cronológicamente y devuelve la información al controlador.
5		El controlador responde a la petición del cliente incluyendo en la cabecera de la respuesta el estado 200 OK para confirmar que la petición se ha completado satisfactoriamente e incluyendo en el cuerpo de la respuesta el fichero JSON con los artículos.
6		El cliente recorre el json de la respuesta y lo transforma para su correcta visualización.
7		El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.
Postcondición		El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.
Excepciones	Paso	Acción
	2	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	4	Si el modelo no puede cargar el fichero, este devuelve un string vacío que provoca que el controlador devuelva un error HTTP 404 al cliente. El cliente devolverá error.
Frecuencia		Muy alta
Estabilidad		Media
Comentarios		Ninguno

Tabla 7 - UC-RQ01

Diagrama de secuencia:

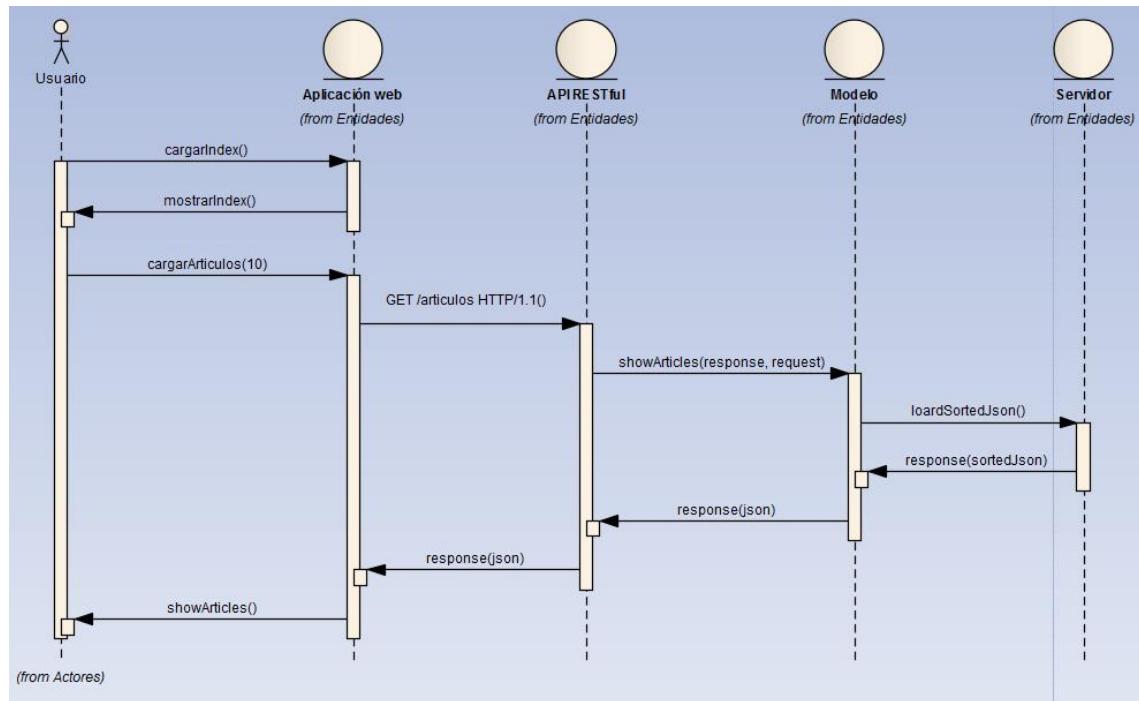


Figura 2 - Diagrama de secuencia UC-01

3.1.2 RQ-02. BUSCAR ARTÍCULOS

UC-RQ01	BUSCAR ARTÍCULOS	
Objetivos asociados	OBJ-02. Servir de directorio de artículos de blogs de emigrantes hispanohablantes en Japón	
Requisitos asociados	RQ02. La solución permitirá la búsqueda sobre artículos en base a una entrada determinada.	
Descripción	La solución provee al usuario de la opción de buscar en el repositorio de artículos según el título que el usuario desee.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El usuario accede a la página principal de la aplicación web o móvil o hace scroll en la página principal superando el offset del marco.
	2	El usuario accede al botón de la lupa situada en la parte inferior izquierda de la pantalla y carga la ventana de búsqueda de artículos.
	3	El usuario introduce el título que quiere buscar en el repositorio de artículos en el cuadro de búsqueda.
	4	El usuario presiona sobre el botón “Buscar artículos” situado en el medio de la pantalla.
	5	El cliente cambia a la página se búsqueda de artículos con los parámetros introducidos por el usuario.
	6	El cliente, en la nueva página, envía una petición HTTP asíncrona al controlador REST para obtener el listado de artículos que encajan en la búsqueda.
	7	El controlador ejecuta la función del modelo asociada al recurso accedido (buscar artículos).
	8	El modelo ejecuta una petición CURL en cuyo cuerpo se encuentra una consulta de ElasticSearch que buscara los artículos que encajen con el título provisto por el usuario utilizando la API REST de Elastic Search.
	9	El servicio de ElasticSearch devuelve una respuesta en cuyo cuerpo se encuentra el JSON con la información de la respuesta a la consulta y los artículos que han encajado.
	10	El modelo devuelve la respuesta JSON al controlador
	11	El controlador devuelve la respuesta JSON al cliente en el cuerpo de la respuesta y establece en la cabecera el estado HTTP 200 OK.

	12	El cliente carga el JSON y recoge sólo la información relacionada a los artículos que han encajado con la búsqueda transformándola para una correcta visualización.
Postcondición		El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.
Excepciones	Paso	Acción
	6	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	7	Si el modelo no puede ejecutar la consulta el controlador devuelve estado 404 ABORTADO en la cabecera de la petición. El cliente devolverá error.
	12	Si no hubiera ningún artículo que encajara con los parámetros de búsqueda el sistema informará al usuario que no se han encontrado artículos relacionados mediante un cuadro de texto.
Frecuencia		Media
Estabilidad		Media
Comentarios		Ninguno

Tabla 8 - UC-RQ01

Diagrama de secuencia

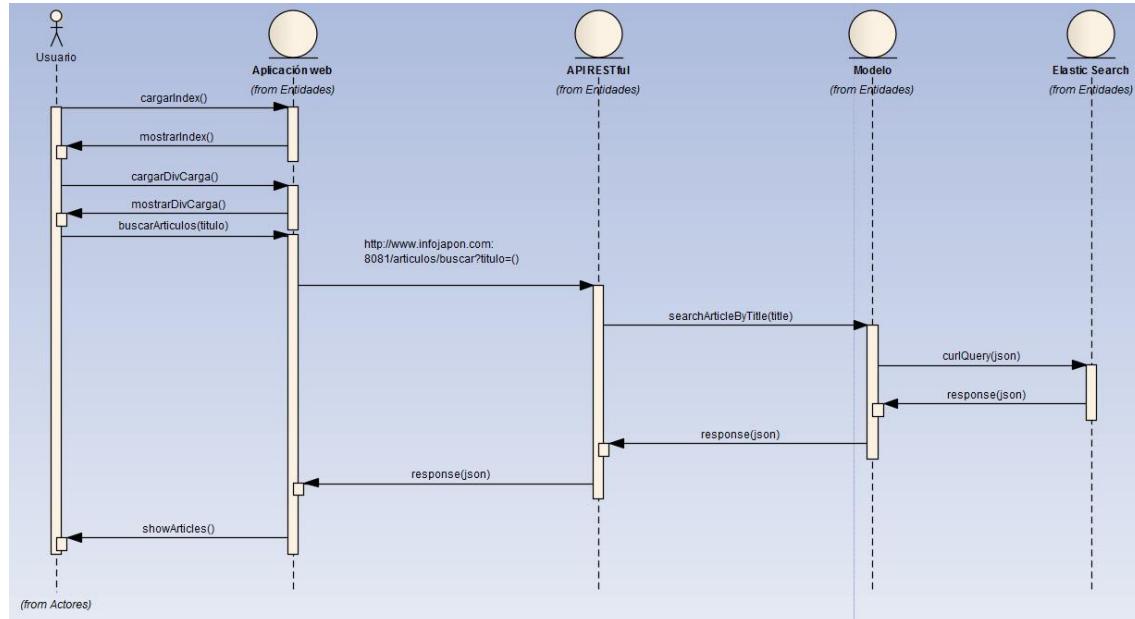


Figura 3 - Diagrama de secuencia UC-02

3.1.3 GRQ01. BUSQUEDA Y VISUALIZACIÓN DE INFORMACIÓN RELACIONADA CON ESTUDIOS

3.1.3.1 GRQ-01-01 BÚSQUEDA Y VISUALIZACIÓN DE INFORMACIÓN RELACIONADA CON UNIVERSIDADES

3.1.3.1.1 RQ01-01-01 - RQ01-01-09 BÚSQUEDA DE UNIVERSIDADES

UC-RQ001-01-01 - 09	BÚSQUEDA DE UNIVERSIDADES	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-01-01. Búsqueda de universidades por prefectura. RQ01-01-02. Búsqueda de universidades por nombre. RQ01-01-03. Búsqueda de universidades por especialidad. RQ01-01-04. Búsqueda de universidades por mes de admisión. RQ01-01-05. Búsqueda de universidades por periodo de admisión de aplicaciones. RQ01-01-06. Búsqueda de universidades en referencia a examen de entrada (EJU). RQ01-01-07. Búsqueda de universidades por tipo de universidad. RQ01-01-08. Búsqueda de universidades por fecha límite de aplicación. RQ01-01-09. Búsqueda de universidades en referencia a certificación de inglés necesaria.	
Descripción	La solución provee al usuario de una lista de universidades que encajen con el formulario llenado.	
Precondición	El usuario ha accedido a la página de estudios	
Secuencia normal	Paso	Acción
	1	En el primer paso del formulario ubicado en la página de estudios, el usuario selecciona en el desplegable “Niveles de estudio” la opción “Grado universitario”.
	2	El usuario presiona el botón “Siguiente” o el botón “2” referente al paso 2 del formulario.
	3	El usuario selecciona en el menú desplegable “Prefectura” la prefectura donde desea buscar las universidades.
	4	El usuario presiona el botón “Siguiente” o el botón “3” referente al paso 3 del formulario.
	5	El usuario selecciona en el menú desplegable “Tipos de estudios universitarios” la especialidad que quiere que tengan las universidades a buscar. Opcionalmente el usuario podrá llenar el cuadro “Nombre de universidad” y los desplegables “Mes de admisión”, “Admisión de aplicaciones”, “Examen de entrada (EJU)”, “Tipo de universidad”, “Fecha límite (mes)” y “Certificación de inglés necesaria”.

	6	El usuario presiona el botón “Iniciar búsqueda” o el botón “4” referente al paso 4 del formulario.
	7	El cliente envía una petición HTTP asíncrona al controlador REST para obtener el listado de universidades que encajen con los parámetros introducidos en el formulario por parte del usuario.
	8	El controlador ejecuta la función del modelo asociada al recurso accedido (buscar universidades).
	9	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling en profundidad de manera recursiva de la web, almacenamiento de información, carga en json).
	10	El sistema devuelve el json generado con la información del listado de universidades al controlador
	11	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	12	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	7	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	9	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 7 - UC-RQ001-01-01 - 09

Diagrama de secuencia

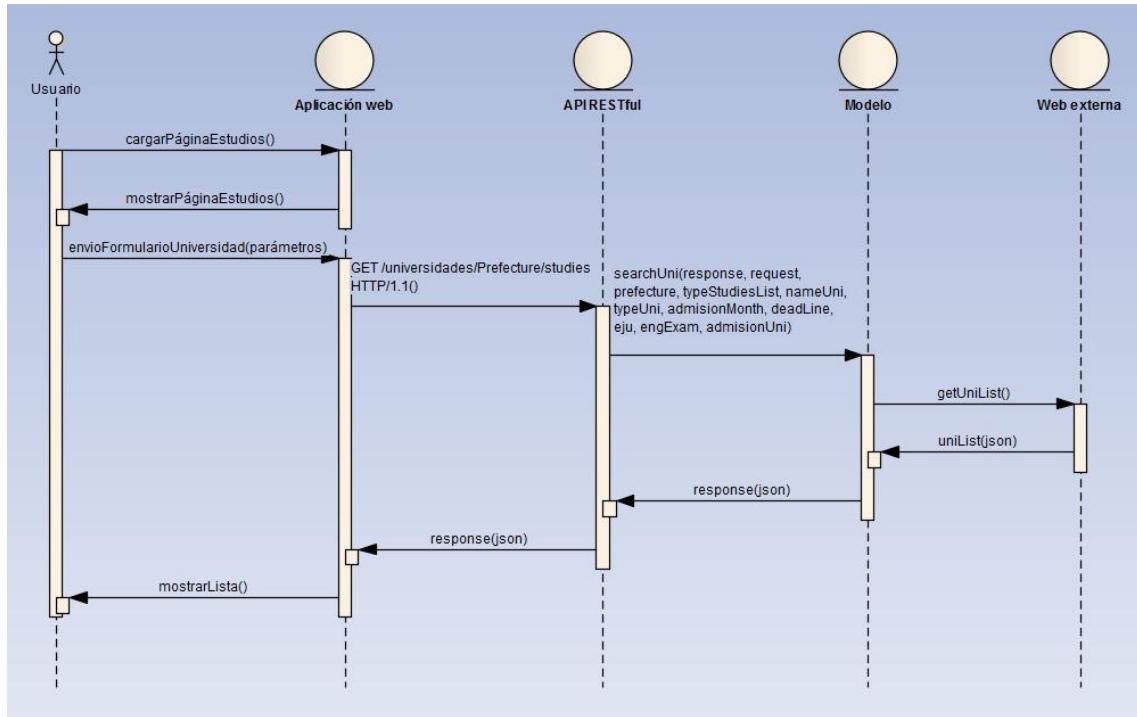


Figura 4 - Diagrama de secuencia UC-RQ001-01-01 - 09

3.1.3.1.2 RQ01-01-10 VISUALIZACIÓN DE UNIVERSIDAD

UC-RQ01-01-10	VISUALIZACIÓN DE UNIVERSIDAD	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-01-10. Visualizar información de la universidad	
Descripción	La solución provee al usuario de la información de la universidad elegida.	
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de universidades obteniendo así una lista con las mismas.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el nombre de “Ver más” de una universidad para acceder a la página de información de la universidad elegida.
	2	El cliente carga la página de escuela pasando como parámetros el id de la universidad elegida por el usuario.
	3	El cliente envía una petición HTTP asíncrona al controlador REST para obtener la información de la universidad elegida en base al id.
	4	El controlador ejecuta la función del modelo asociada al recurso accedido (obtener información de universidad).
	5	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).
	6	El sistema devuelve el JSON generado con la información de la universidad al controlador
	7	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	8	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	5	Si el sistema no puede ejecutar el proceso de crawling el

		controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 8 - UC-RQ01-01-10

Diagrama de secuencia

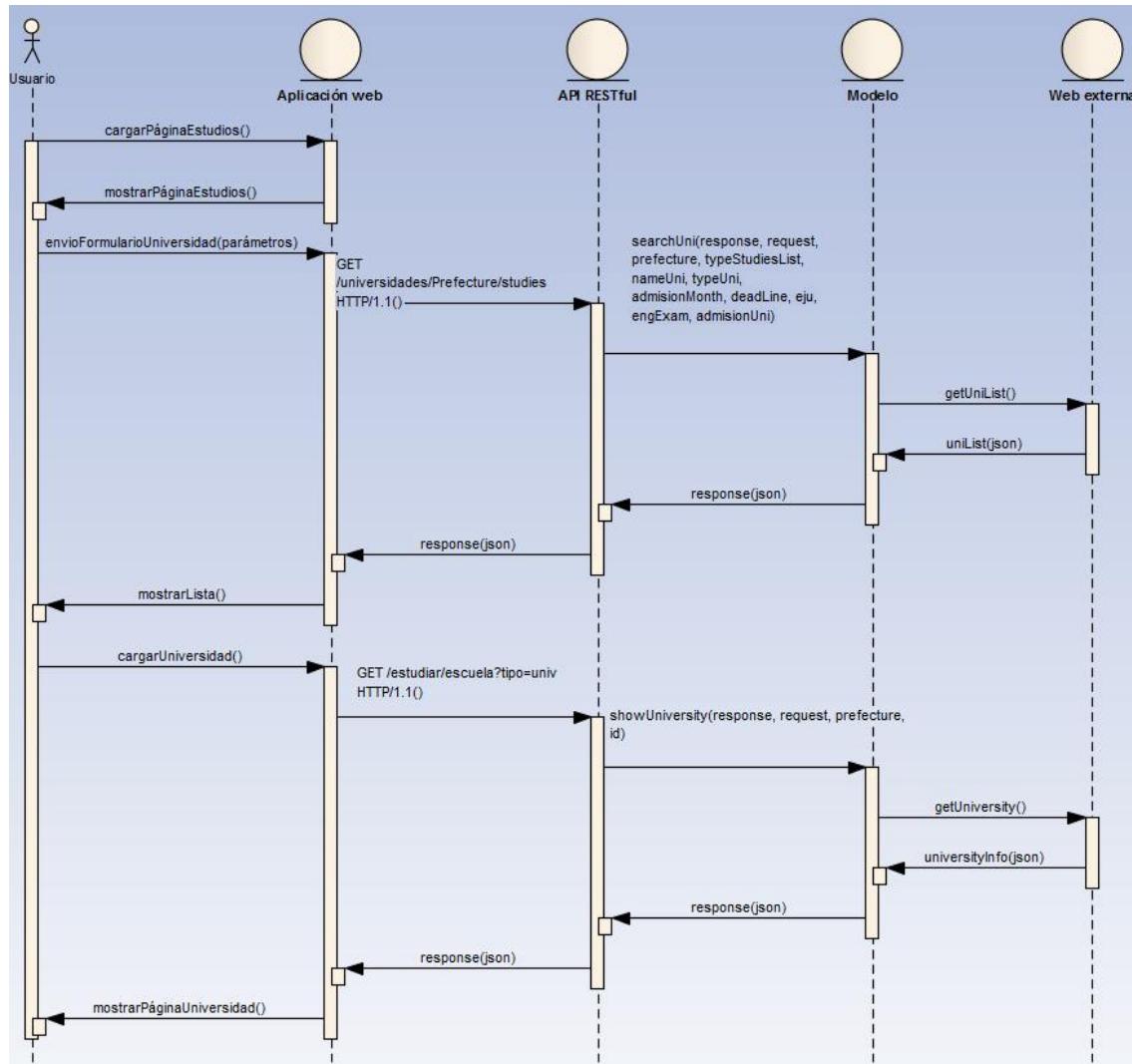


Figura 5 - Diagrama de secuencia UC-RQ01-01-10

3.1.3.1.3 RQ01-01-11 VISUALIZACIÓN DE DEPARTAMENTO UNIVERSITARIO

UC-RQ01-01-11	VISUALIZACIÓN DE DEPARTAMENTO UNIVERSITARIO																		
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón																		
Requisitos asociados	RQ01-01-11. Visualizar información del departamento universitario																		
Descripción	La solución provee al usuario de la información del departamento de la universidad elegida.																		
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de universidades obteniendo así una lista con las mismas o bien el usuario ha accedido a la página de una universidad.																		
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario presiona el botón azul con el nombre del departamento en el que está interesado para acceder a la página de información del departamento elegido.</td></tr> <tr> <td>2</td><td>El cliente carga la página de departamento pasando como parámetros el id de la universidad elegida y el id del departamento elegido por el usuario.</td></tr> <tr> <td>3</td><td>El cliente envía un conjunto de peticiones HTTP asíncronas al controlador REST para obtener la información del departamento.</td></tr> <tr> <td>4</td><td>El controlador ejecuta el conjunto de funciones del modelo asociadas a los recursos accedidos (obtener información básica del departamento, obtener información de acceso, obtener información de admisión, obtener información de instalaciones, obtener información ayuda al estudiante).</td></tr> <tr> <td>5</td><td>El sistema inicia los diferentes procesos de crawling concurrentemente. (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).</td></tr> <tr> <td>6</td><td>El sistema devuelve los JSON generados paralelamente.</td></tr> <tr> <td>7</td><td>El controlador devuelve el JSON específico generado en el cuerpo de cada respuesta y en la cabecera devuelve el estado HTTP 200 OK.</td></tr> <tr> <td>8</td><td>El cliente carga los diferentes JSON de forma asíncrona transformando la información contenida para una correcta visualización.</td></tr> </tbody> </table>	Paso	Acción	1	El usuario presiona el botón azul con el nombre del departamento en el que está interesado para acceder a la página de información del departamento elegido.	2	El cliente carga la página de departamento pasando como parámetros el id de la universidad elegida y el id del departamento elegido por el usuario.	3	El cliente envía un conjunto de peticiones HTTP asíncronas al controlador REST para obtener la información del departamento.	4	El controlador ejecuta el conjunto de funciones del modelo asociadas a los recursos accedidos (obtener información básica del departamento, obtener información de acceso, obtener información de admisión, obtener información de instalaciones, obtener información ayuda al estudiante).	5	El sistema inicia los diferentes procesos de crawling concurrentemente. (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).	6	El sistema devuelve los JSON generados paralelamente.	7	El controlador devuelve el JSON específico generado en el cuerpo de cada respuesta y en la cabecera devuelve el estado HTTP 200 OK.	8	El cliente carga los diferentes JSON de forma asíncrona transformando la información contenida para una correcta visualización.
Paso	Acción																		
1	El usuario presiona el botón azul con el nombre del departamento en el que está interesado para acceder a la página de información del departamento elegido.																		
2	El cliente carga la página de departamento pasando como parámetros el id de la universidad elegida y el id del departamento elegido por el usuario.																		
3	El cliente envía un conjunto de peticiones HTTP asíncronas al controlador REST para obtener la información del departamento.																		
4	El controlador ejecuta el conjunto de funciones del modelo asociadas a los recursos accedidos (obtener información básica del departamento, obtener información de acceso, obtener información de admisión, obtener información de instalaciones, obtener información ayuda al estudiante).																		
5	El sistema inicia los diferentes procesos de crawling concurrentemente. (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).																		
6	El sistema devuelve los JSON generados paralelamente.																		
7	El controlador devuelve el JSON específico generado en el cuerpo de cada respuesta y en la cabecera devuelve el estado HTTP 200 OK.																		
8	El cliente carga los diferentes JSON de forma asíncrona transformando la información contenida para una correcta visualización.																		
Postcondición	El cliente carga asíncronamente las transformaciones en un cuadro previamente invisible inmediatamente para la visualización del usuario.																		

Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos sólo en el cuadro de información asociado.
	5	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error en el cuadro asociado (información, soporte...).
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 9 - UC-RQ01-01-11

Diagrama de secuencia

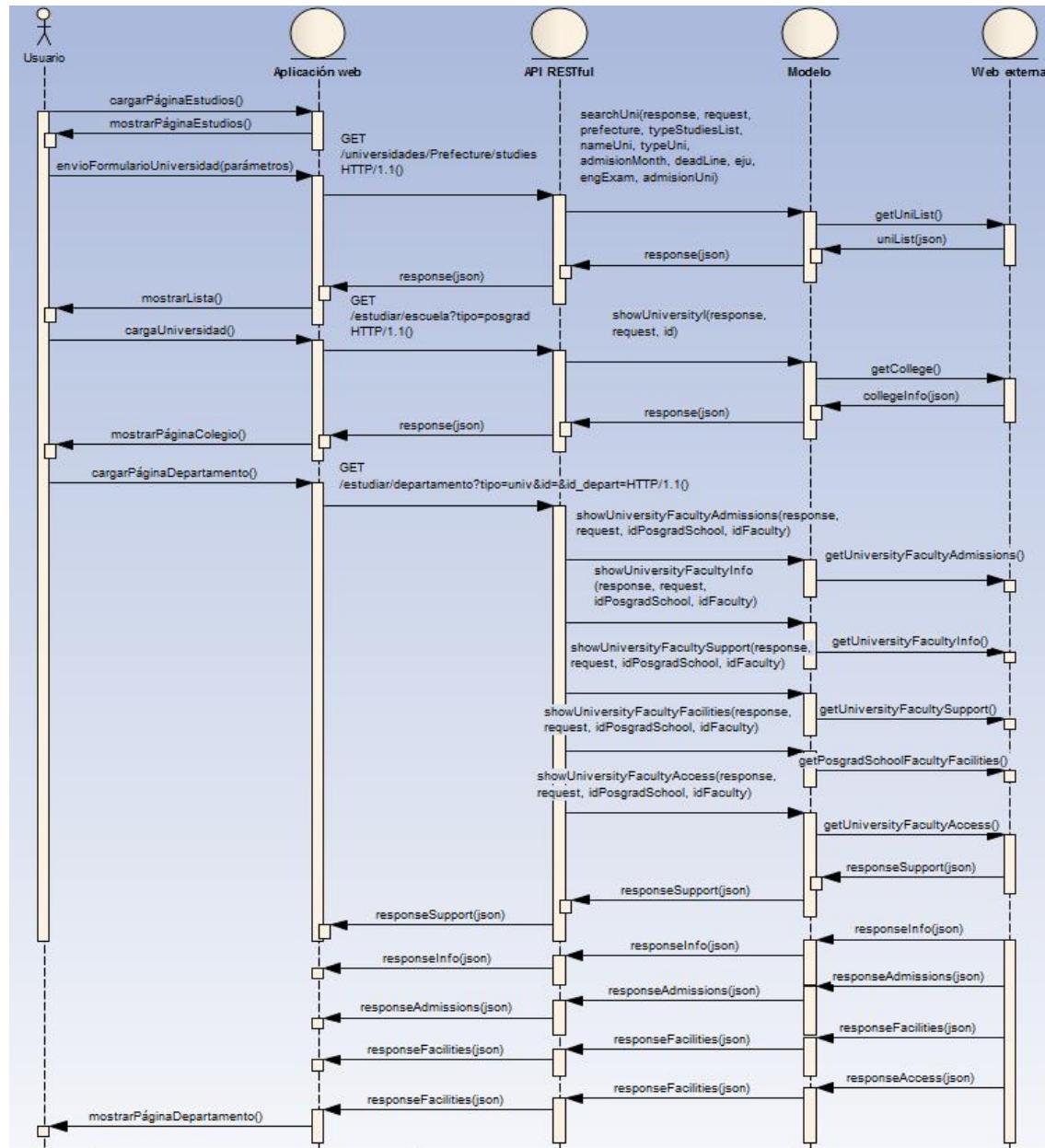


Figura 6 - Diagrama de secuencia UC-RQ01-01-11

3.1.3.2 GRQ-01-02 BÚSQUEDA Y VISUALIZACIÓN DE INFORMACIÓN RELACIONADA CON COLEGIOS DE POSGRADO

3.1.3.2.1 RQ01-02-01 – RQ01-02-05 BÚSQUEDA DE COLEGIOS DE POSGRADO

UC-RQ01-02-01 - 05	BÚSQUEDA DE COLEGIOS DE POSGRADO	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-02-01. Búsqueda de colegios de posgrado por prefectura. RQ01-02-02. Búsqueda colegios de posgrado por nombre. RQ01-02-03. Búsqueda colegios de posgrado por especialidad. RQ01-02-04. Búsqueda colegios de posgrado por tipo de escuela. RQ01-02-05. Búsqueda colegios de posgrado en referencia al idioma hablado en la clase.	
Descripción	La solución provee al usuario de una lista de colegios de posgrado que encajen con el formulario llenado.	
Precondición	El usuario ha accedido a la página de estudios	
Secuencia normal	Paso	Acción
	1	En el primer paso del formulario ubicado en la página de estudios, el usuario selecciona en el desplegable “Niveles de estudio” la opción “Master o posgrado”.
	2	El usuario presiona el botón “Siguiente” o el botón “2” referente al paso 2 del formulario.
	3	El usuario selecciona en el menú desplegable “Prefectura” la prefectura donde desea buscar los colegios de posgrado.
	4	El usuario presiona el botón “Siguiente” o el botón “3” referente al paso 3 del formulario.
	5	El usuario selecciona en el menú desplegable “Tipos de estudios de posgrado” la especialidad que quiere que tengan los colegios de posgrado a buscar. Opcionalmente el usuario podrá llenar el cuadro “Nombre de escuela” y los desplegables “Tipo de universidad” y “Idioma de las clases”.
	6	El usuario presiona el botón “Iniciar búsqueda” o el botón “4” referente al paso 4 del formulario.
	7	El cliente envía una petición HTTP asíncrona al controlador REST para obtener el listado de colegios de posgrado que encajen con los parámetros introducidos en el formulario por parte del usuario.
	8	El controlador ejecuta la función del modelo asociada al recurso accedido (buscar colegios de posgrado).

	9	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling en profundidad de manera recursiva de la web, almacenamiento de información, carga en json).
	10	El sistema devuelve el JSON generado con la información del listado de colegios de posgrado al controlador
	11	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	12	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	7	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	9	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 10 - UC-RQ01-02-01 - 05

Diagrama de secuencia

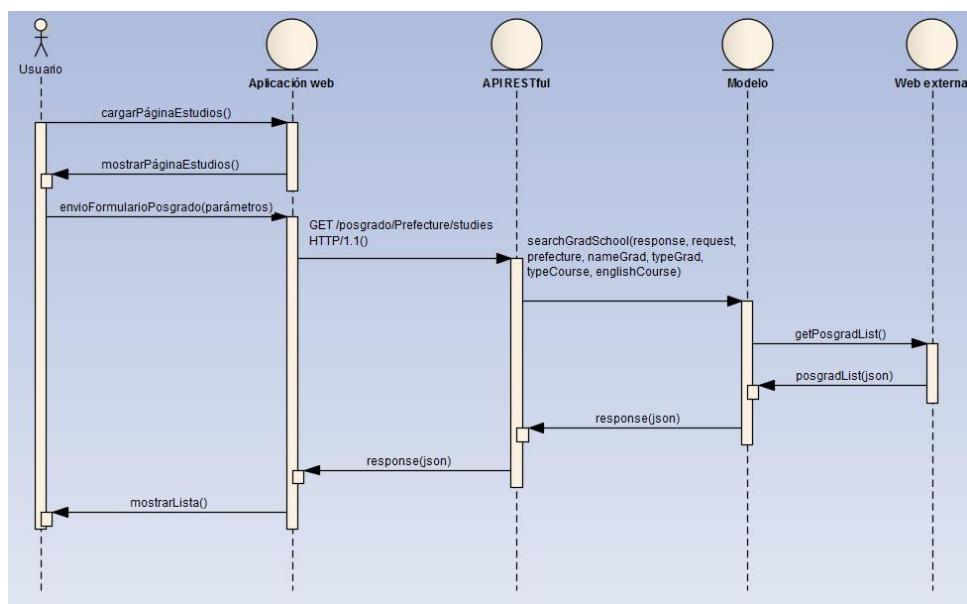


Figura 7 - Diagrama de secuencia UC-RQ01-02-01 - 05

3.1.3.2.2 RQ01-02-05 VISUALIZACIÓN DE COLEGIO DE POSGRADO

UC-RQ01-02-06	VISUALIZACIÓN DE COLEGIO DE POSGRADO	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-02-06. Visualizar información de colegio de posgrado.	
Descripción	La solución provee al usuario de la información del colegio de posgrado elegido.	
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de colegios de posgrado obteniendo así una lista con los mismos.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el nombre de “Ver más” de un colegio de posgrado para acceder a la página de información del colegio de posgrado elegido.
	2	El cliente carga la página de escuela pasando como parámetros el id del colegio de posgrado elegido por el usuario.
	3	El cliente envía una petición HTTP asíncrona al controlador REST para obtener la información del colegio de posgrado elegido en base al id.
	4	El controlador ejecuta la función del modelo asociada al recurso accedido (obtener información de colegio de posgrado).
	5	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).
	6	El sistema devuelve el JSON generado con la información del colegio de posgrado al controlador.
	7	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	8	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no

		ha sido posible conectarse con el servidor de datos.
	5	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 11 - UC-RQ01-02-06

Diagrama de secuencia

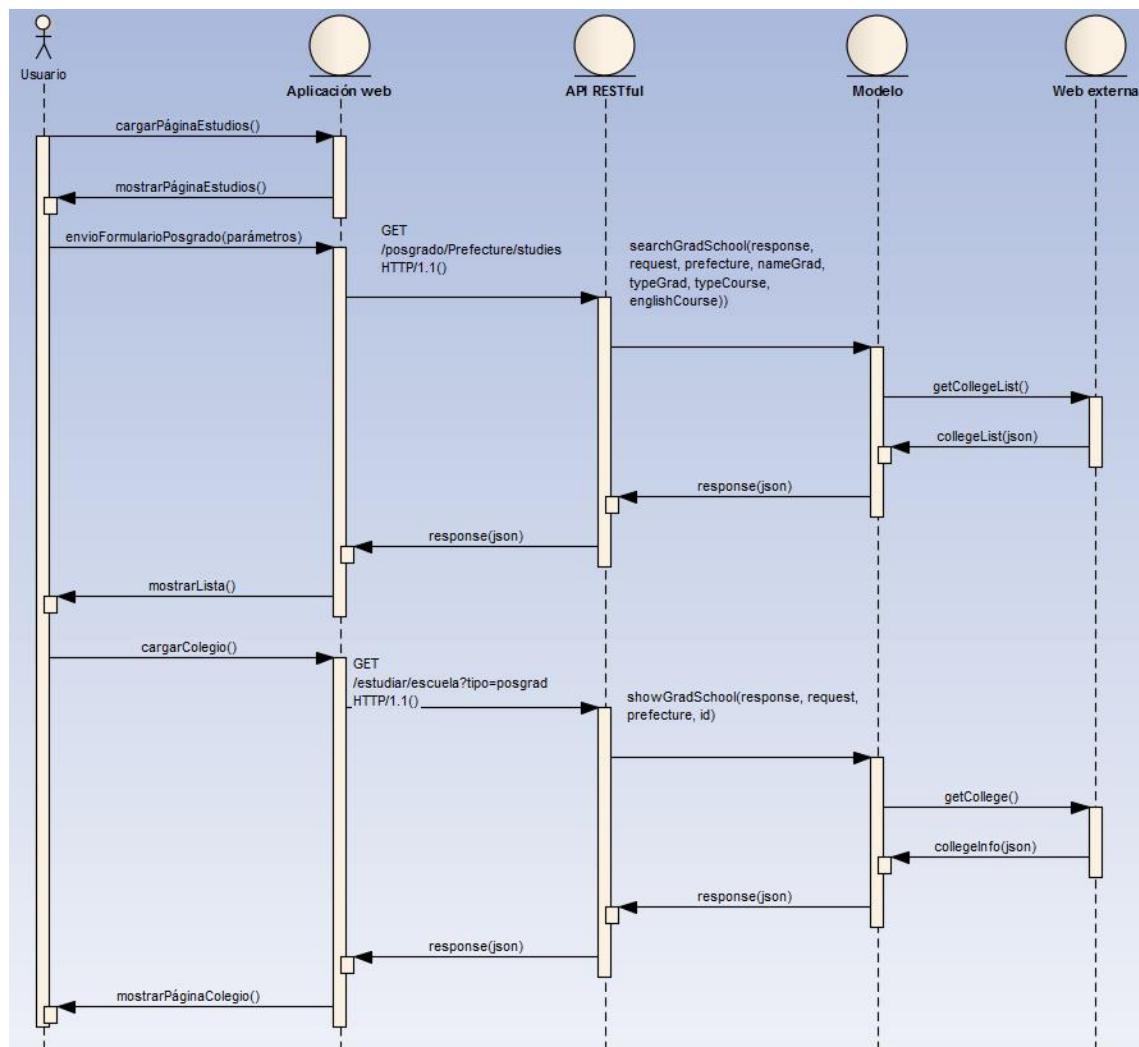


Figura 8 - Diagrama de secuencia UC-RQ01-02-06

3.1.3.2.3 RQ01-02-06 VISUALIZACIÓN DE DEPARTAMENTO DE COLEGIO DE POSGRADO

UC-RQ01-02-07	VISUALIZACIÓN DE DEPARTAMENTO DE COLEGIO DE COLEGIO DE POSGRADO
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón
Requisitos asociados	RQ01-02-07. Visualizar información del departamento del colegio de posgrado
Descripción	La solución provee al usuario de la información del departamento del colegio de posgrado elegido.
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de colegios de posgrado obteniendo así una lista con las mismas o bien el usuario ha accedido a la página de un colegio de posgrado.
Secuencia normal	Paso Acción
	1 El usuario presiona el botón azul con el nombre del departamento en el que está interesado para acceder a la página de información del departamento elegido.
	2 El cliente carga la página de departamento pasando como parámetros el id del colegio de posgrado elegido y el id del departamento elegido por el usuario.
	3 El cliente envía un conjunto de peticiones HTTP asíncronas al controlador REST para obtener la información del departamento.
	4 El controlador ejecuta el conjunto de funciones del modelo asociadas a los recursos accedidos (obtener información básica del departamento, obtener información de acceso, obtener información de admisión, obtener información de instalaciones, obtener información ayuda al estudiante).
	5 El sistema inicia los diferentes procesos de crawling concurrentemente. (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).
	6 El sistema devuelve los JSON generados paralelamente.
	7 El controlador devuelve el JSON específico generado en el cuerpo de cada respuesta y en la cabecera devuelve el estado HTTP 200 OK.
	8 El cliente carga los diferentes JSON de forma asíncrona transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga asíncronamente las transformaciones en un cuadro

	previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos sólo en el cuadro de información asociado.
	5	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error en el cuadro asociado (información, soporte...).
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 12 - UC-RQ01-02-07

Diagrama de secuencia

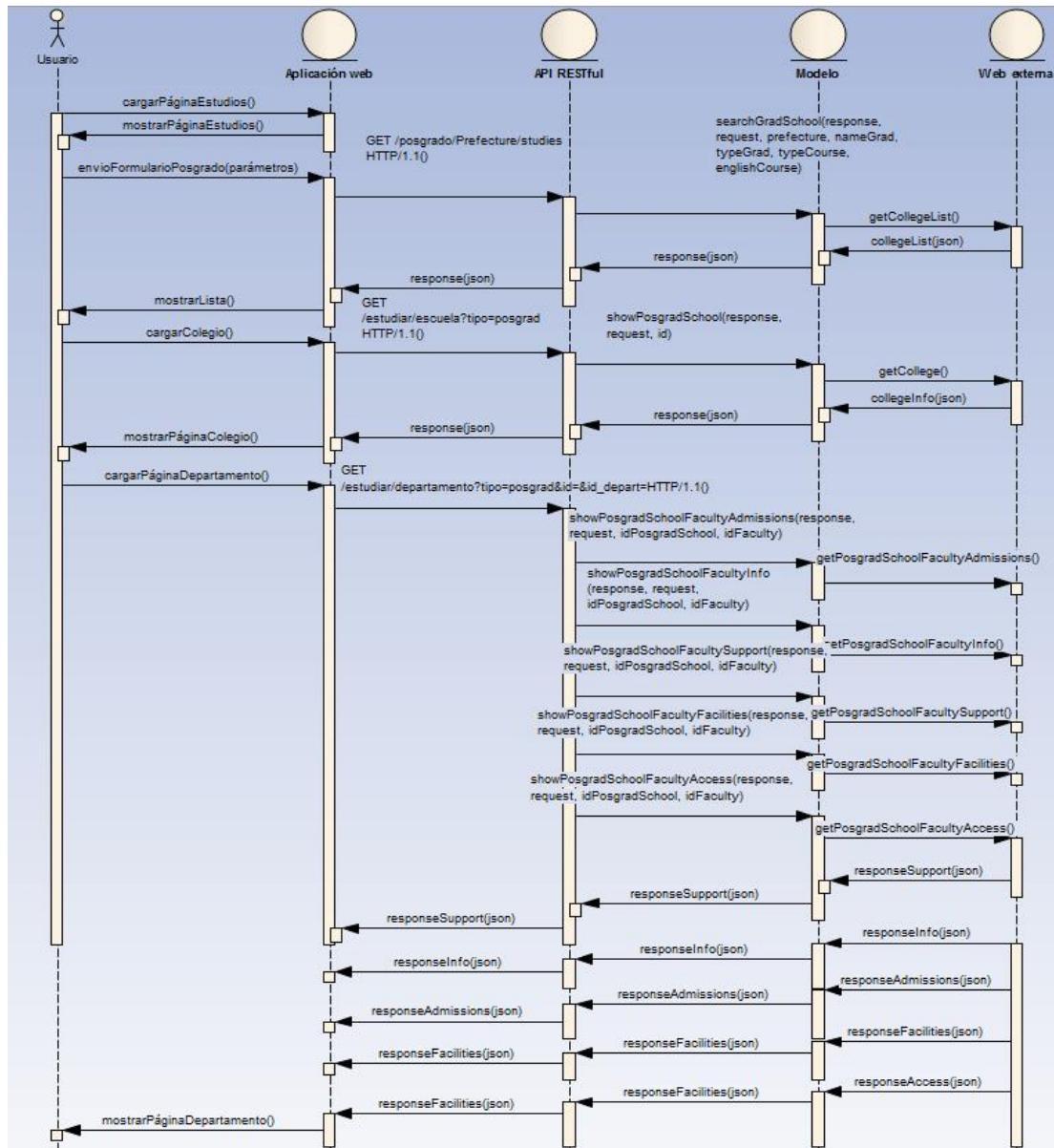


Figura 9 - Diagrama de secuencia UC-RQ01-02-07

3.1.3.3 GRQ-01-03 BÚSQUEDA Y VISUALIZACIÓN DE INFORMACIÓN RELACIONADA CON COLEGIOS DE FORMACIÓN PROFESIONAL

3.1.3.3.1 RQ01-03-01 – RQ01-03-02 BÚSQUEDA DE COLEGIOS DE FORMACIÓN PROFESIONAL

UC-RQ01-03-01 - 02	BÚSQUEDA DE COLEGIOS DE FORMACIÓN PROFESIONAL	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-03-01. Búsqueda de colegios de formación profesional por prefectura. RQ01-03-02. Búsqueda de colegios de formación profesional por nombre.	
Descripción	La solución provee al usuario de una lista de colegios de formación profesional que encajen con el formulario rellenado.	
Precondición	El usuario ha accedido a la página de estudios	
Secuencia normal	Paso	Acción
	1	En el primer paso del formulario ubicado en la página de estudios, el usuario selecciona en el desplegable “Niveles de estudio” la opción “Formación profesional”.
	2	El usuario presiona el botón “Siguiente” o el botón “2” referente al paso 2 del formulario.
	3	El usuario selecciona en el menú desplegable “Prefectura” la prefectura donde desea buscar los colegios de formación profesional.
	4	El usuario presiona el botón “Siguiente” o el botón “3” referente al paso 3 del formulario.
	5	El usuario seleccionará opcionalmente el rellenar el campo “Nombre de escuela”.
	6	El usuario presiona el botón “Iniciar búsqueda” o el botón “4” referente al paso 4 del formulario.
	7	El cliente envía una petición HTTP asíncrona al controlador REST para obtener el listado de colegios de formación profesional que encajen con los parámetros introducidos en el formulario por parte del usuario.
	8	El controlador ejecuta la función del modelo asociada al recurso accedido (buscar colegios de formación profesional).
	9	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling en profundidad de manera recursiva de la web, almacenamiento de información, carga en json).

	10	El sistema devuelve el JSON generado con la información del listado de colegios de formación profesional al controlador
	11	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	12	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	7	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	9	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 13 - UC-RQ01-03-01 - 02

Diagrama de secuencia

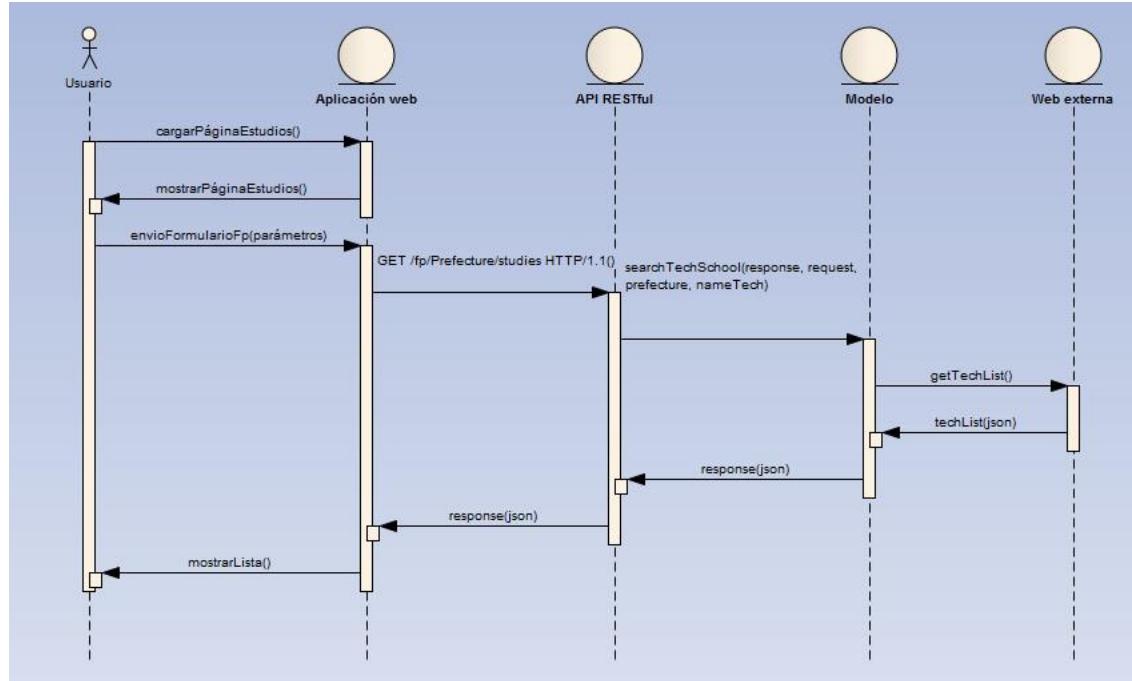


Figura 10 - Diagrama de secuencia UC-RQ01-03-01 - 02

3.1.3.3.2 RQ01-03-03 VISUALIZACIÓN DE COLEGIO DE FORMACIÓN PROFESIONAL

UC-RQ01-03-03	VISUALIZACIÓN DE COLEGIO DE FORMACIÓN PROFESIONAL	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-03-03. Visualizar información del colegio de formación profesional.	
Descripción	La solución provee al usuario de la información del colegio de formación profesional elegido.	
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de colegios de formación profesional obteniendo así una lista con los mismos.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el nombre de “Ver más” de un colegio de formación profesional para acceder a la página de información del colegio de formación profesional elegido.
	2	El cliente carga la página de escuela pasando como parámetros el id del colegio de formación profesional elegido por el usuario.
	3	El cliente envía una petición HTTP asíncrona al controlador REST para obtener la información del colegio de formación profesional elegido en base al id.
	4	El controlador ejecuta la función del modelo asociada al recurso accedido (obtener información de colegio de formación profesional).
	5	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).
	6	El sistema devuelve el JSON generado con la información del colegio de formación profesional al controlador.
	7	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	8	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no

		ha sido posible conectarse con el servidor de datos.
	5	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 14 - UC-RQ01-03-03

Diagrama de secuencia

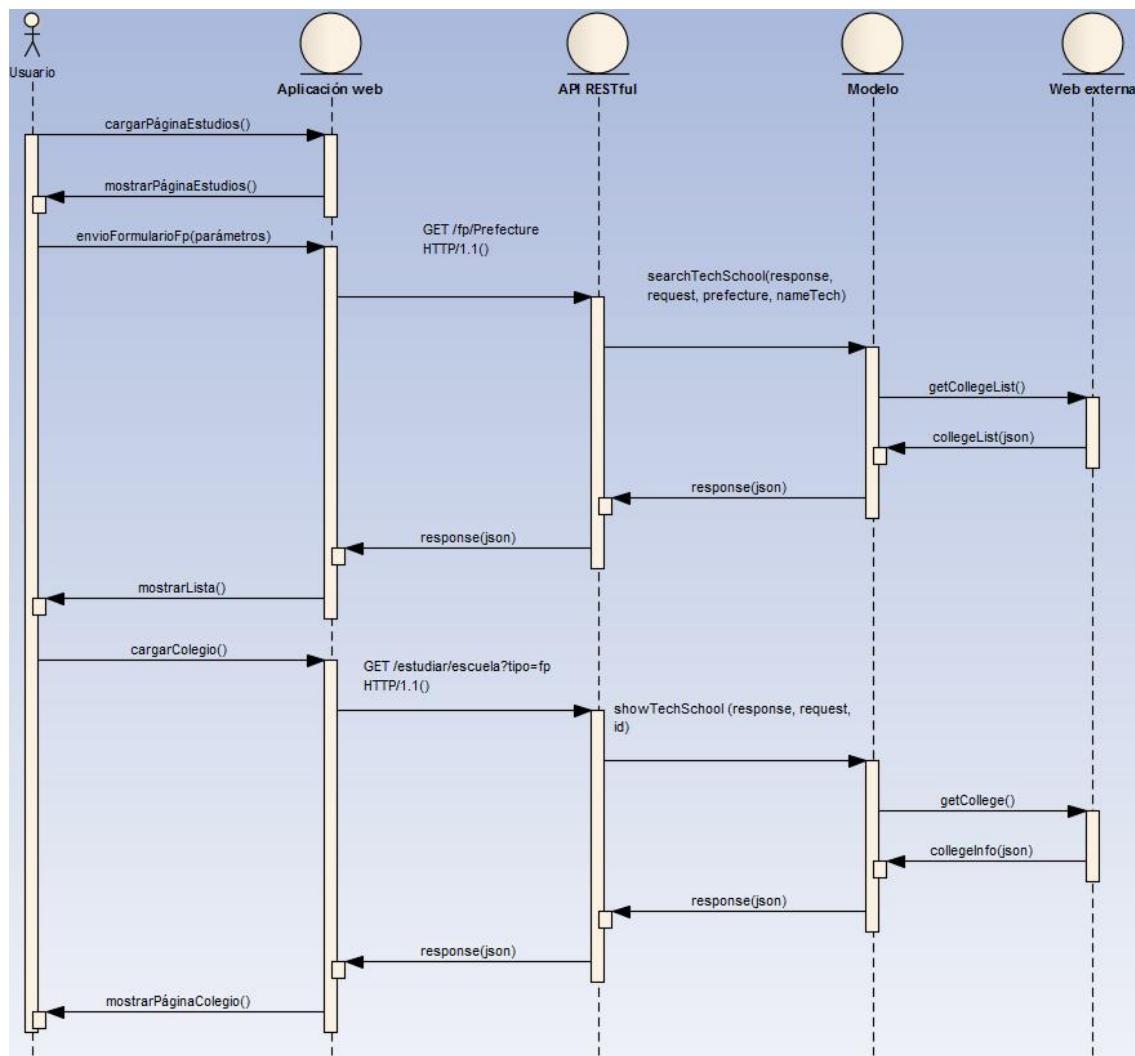


Figura 11 - Diagrama de secuencia UC-RQ01-03-03

3.1.3.3.3 RQ01-03-04 VISUALIZACIÓN DE DEPARTAMENTO DE COLEGIO DE FORMACIÓN PROFESIONAL

UC-RQ01-03-04	VISUALIZACIÓN DE DEPARTAMENTO DE COLEGIO DE COLEGIO DE FORMACIÓN PROFESIONAL	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-03-04. Visualizar información del departamento de colegio de formación profesional	
Descripción	La solución provee al usuario de la información del departamento del colegio de formación profesional elegido.	
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de colegios de formación profesional obteniendo así una lista con las mismas o bien el usuario ha accedido a la página de un colegio de formación profesional.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el botón azul con el nombre del departamento en el que está interesado para acceder a la página de información del departamento elegido.
	2	El cliente carga la página de departamento pasando como parámetros el id del colegio de formación profesional elegido y el id del departamento elegido por el usuario.
	3	El cliente envía un conjunto de peticiones HTTP asíncronas al controlador REST para obtener la información del departamento.
	4	El controlador ejecuta el conjunto de funciones del modelo asociadas a los recursos accedidos (obtener información básica del departamento, obtener información de admisión, información ayuda al estudiante).
	5	El sistema inicia los diferentes procesos de crawling concurrentemente. (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).
	6	El sistema devuelve los JSON generados paralelamente.
	7	El controlador devuelve el JSON específico generado en el cuerpo de cada respuesta y en la cabecera devuelve el estado HTTP 200 OK.
	8	El cliente carga los diferentes JSON de forma asíncrona transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga asíncronamente las transformaciones en un cuadro	

	previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos sólo en el cuadro de información asociado.
	5	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error en el cuadro asociado (información, soporte...).
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 15 - UC-RQ01-03-04

Diagrama de secuencia

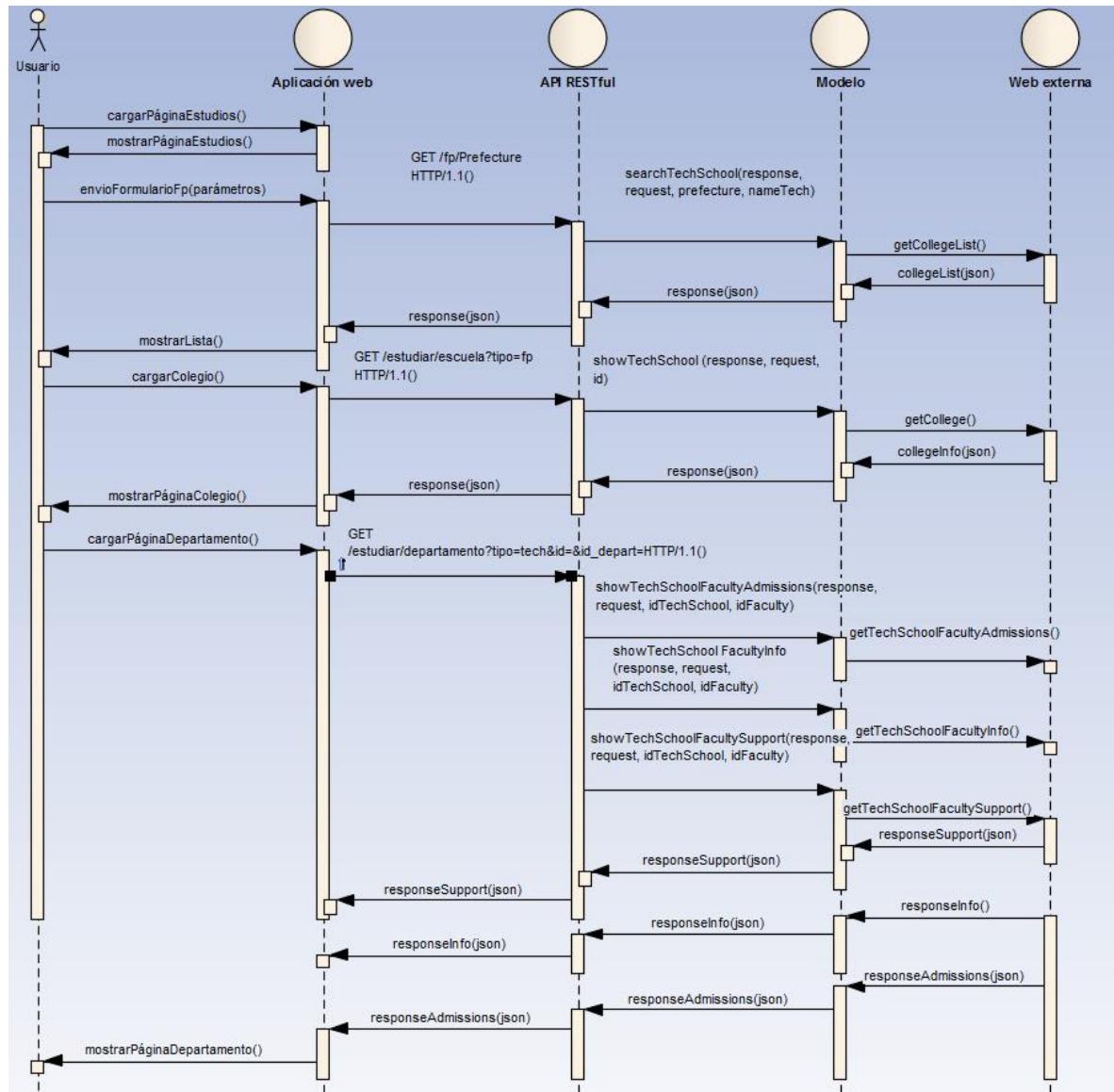


Figura 12 - Diagrama de secuencia UC-RQ01-03-04

3.1.3.4 GR-01-04 BÚSQUEDA Y VISUALIZACIÓN DE INFORMACIÓN RELACIONADA CON COLEGIOS DE JAPONÉS

3.1.3.4.1 RQ01-04-01 – RQ01-04-02 BÚSQUEDA DE COLEGIOS DE JAPONÉS

UC-RQ01-04-01 - 02	BÚSQUEDA DE COLEGIOS DE JAPONÉS	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-04-01. Búsqueda de colegios de japonés por prefectura RQ01-04-02. Búsqueda de colegios de japonés por barrio	
Descripción	La solución provee al usuario de una lista de colegios de japonés que encajen con el formulario llenado.	
Precondición	El usuario ha accedido a la página de estudios	
Secuencia normal	Paso	Acción
	1	En el primer paso del formulario ubicado en la página de estudios, el usuario selecciona en el desplegable “Niveles de estudio” la opción “Lengua japonesa”.
	2	El usuario presiona el botón “Siguiente” o el botón “2” referente al paso 2 del formulario.
	3	El usuario selecciona en el menú desplegable “Prefectura” la prefectura donde desea buscar los colegios de japonés. Opcionalmente, si la prefectura así lo dispone, el usuario visualizará un nuevo desplegable para elegir el barrio donde quiere buscar las escuelas de japonés.
	4	El usuario presiona el botón “Siguiente” o el botón “3” referente al paso 3 del formulario.
	5	El usuario presiona el botón “Iniciar búsqueda” o el botón “4” referente al paso 4 del formulario.
	6	El cliente envía una petición HTTP asíncrona al controlador REST para obtener el listado de colegios de japonés que encajen con los parámetros introducidos en el formulario por parte del usuario.
	7	El controlador ejecuta la función del modelo asociada al recurso accedido (buscar colegios de japonés).
	8	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en json).
	9	El sistema devuelve el JSON generado con la información del listado de colegios de japonés al controlador
	10	El controlador devuelve el JSON generado en el cuerpo de la

		respuesta y en la cabecera devuelve el estado HTTP 200 OK
11		El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	6	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	8	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 16 - UC-RQ01-04-01 - 02

Diagrama de secuencia

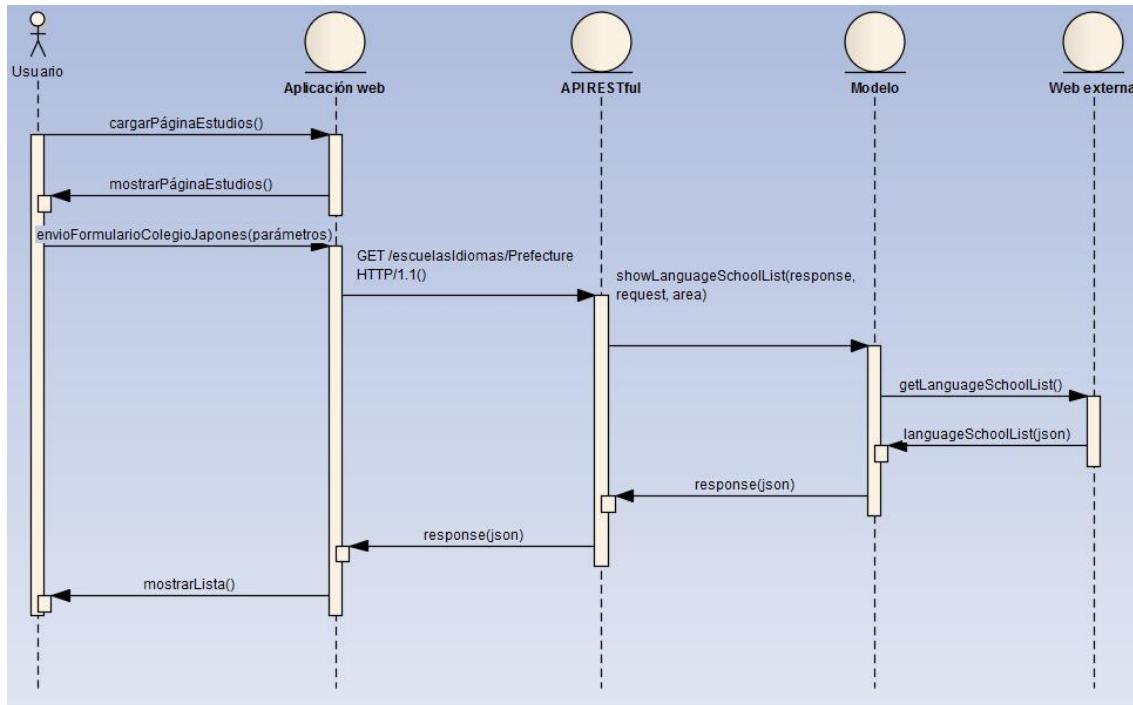


Figura 13 - Diagrama de secuencia UC-RQ01-04-01 - 02

3.1.3.4.2 RQ01-04-03 VISUALIZACIÓN DE COLEGIO DE JAPONÉS

UC-RQ01-04-03	VISUALIZACIÓN DE COLEGIO DE FORMACIÓN PROFESIONAL	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ01-04-03. Visualizar información de colegio de japonés	
Descripción	La solución provee al usuario de la información del colegio de japonés elegido.	
Precondición	El usuario ha accedido a la página de estudios y ha efectuado una búsqueda de colegios de japonés obteniendo así una lista con los mismos.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el nombre de “Más información” de un colegio de japonés para acceder a la página de información del colegio de japonés elegido.
	2	El cliente carga la página de escuela pasando como parámetros el id del colegio de japonés elegido por el usuario.
	3	El cliente envía una petición HTTP asíncrona al controlador REST para obtener la información del japonés elegido en base al id.
	4	El controlador ejecuta la función del modelo asociada al recurso accedido (obtener información de colegio de japonés).
	5	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en JSON).
	6	El sistema devuelve el JSON generado con la información del colegio de japonés al controlador.
	7	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	8	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	3	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no

		ha sido posible conectarse con el servidor de datos.
	5	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 17 - UC-RQ01-04-03

Diagrama de secuencia

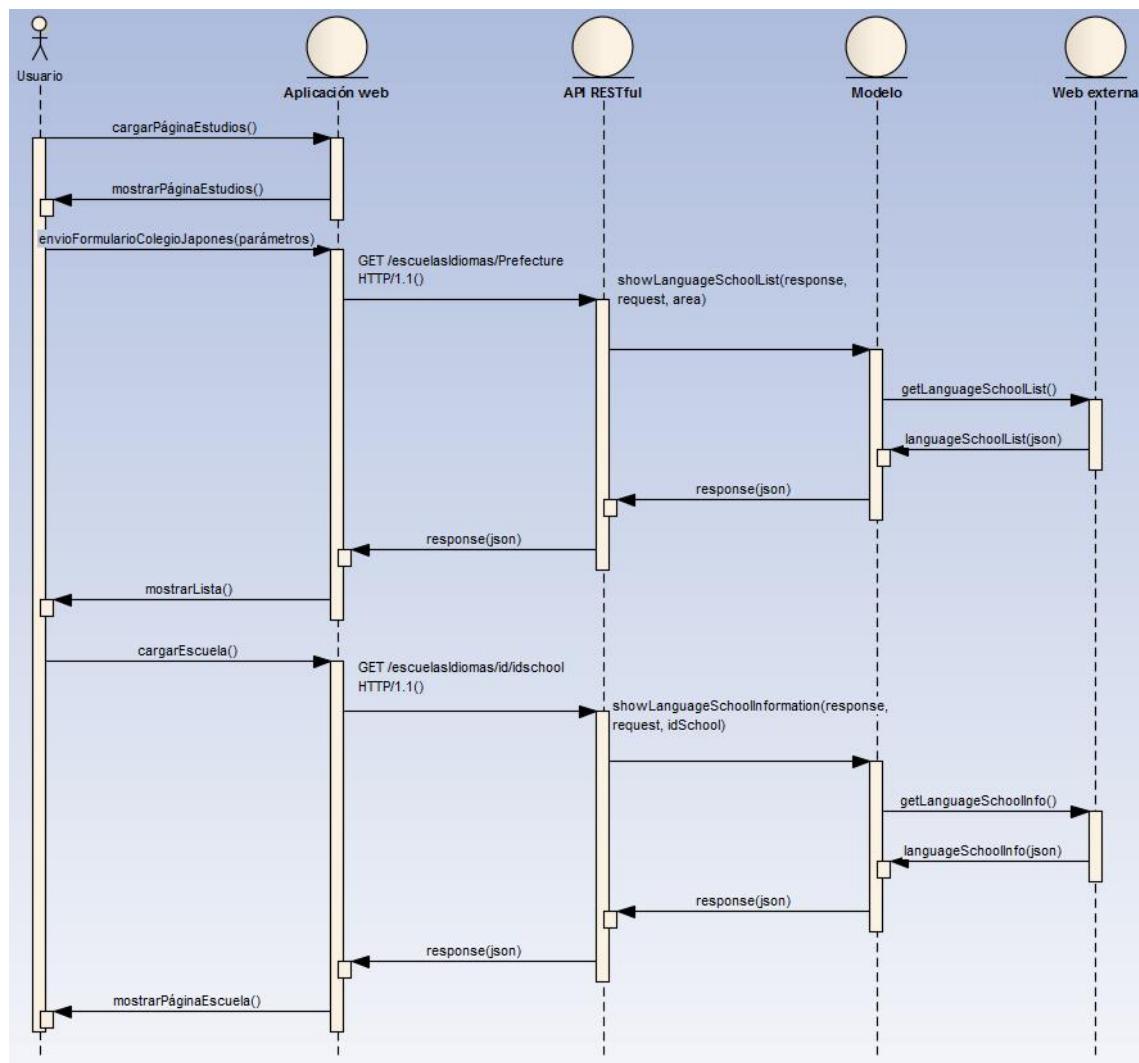


Figura 14 - Diagrama de secuencia UC-RQ01-04-03

3.1.4 GRQ03. BÚSQUEDA DE OFERTAS DE EMPLEO

UC-GR03	BÚSQUEDA DE OFERTAS DE EMPLEO	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ03-01. Búsqueda por localización RQ03-02. Búsqueda por especialidad	
Descripción	La solución provee al usuario de una lista de ofertas de trabajo que encajen con el formulario rellenado.	
Precondición	El usuario ha accedido a la página de trabajo	
Secuencia normal	Paso	Acción
	1	En el primer paso del formulario ubicado en la página de trabajo, el usuario seleccionará en el desplegable “Especialidad” el área donde se buscarán las ofertas de trabajo.
	2	El usuario presiona el botón “Siguiente” o el botón “2” referente al paso 2 del formulario.
	3	El usuario selecciona en el menú desplegable “Prefectura” la prefectura donde desea buscar las ofertas de empleo.
	4	El usuario presiona el botón “Siguiente” o el botón “3” referente al paso 3 del formulario.
	5	El cliente envía una petición HTTP asíncrona al controlador REST para obtener el listado de ofertas de empleo que encajen con los parámetros introducidos en el formulario por parte del usuario.
	6	El controlador ejecuta la función del modelo asociada al recurso accedido (buscar ofertas de trabajo).
	7	El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en json).
	8	El sistema devuelve el JSON generado con la información del listado de ofertas de trabajo al controlador
	9	El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
	10	El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	

Excepciones	Paso	Acción
	5	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	7	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 18 - UC-GR03

Diagrama de secuencia

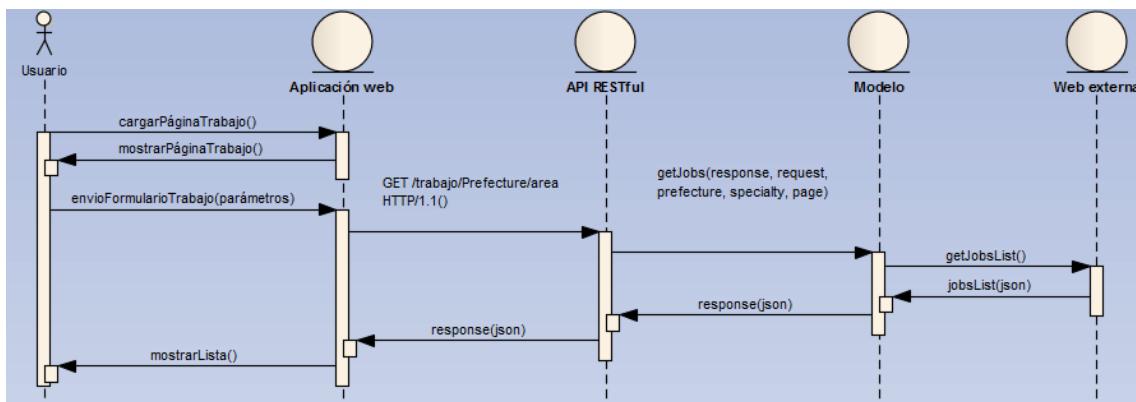


Figura 15 - Diagrama de secuencia UC-GR03

3.1.5 RQ04. BÚSQUEDA DE VIVIENDAS

UC-GR04	BÚSQUEDA DE VIVIENDAS	
Objetivos asociados	OBJ-01. Proveer información referente a estudiar, trabajar y vivir en Japón	
Requisitos asociados	RQ04-01. Búsqueda por prefectura RQ04-02. Búsqueda por alquiler mínimo RQ04-03. Búsqueda por alquiler máximo RQ04-04. Búsqueda por dimensión RQ04-05. Búsqueda por nº de habitaciones RQ04-06. Búsqueda por tipo de construcción RQ04-07. Búsqueda en base a la distancia a la estación más cercana RQ04-08. Búsqueda en base a la antigüedad de la vivienda. RQ04-09. Búsqueda en base a los bajos costes iniciales. RQ04-10. Búsqueda en base al pago de una señal inicial. RQ04-11. Búsqueda en base al deposito RQ04-12. Búsqueda en base a los gastos de la agencia RQ04-13. Búsqueda en base al servicio de internet en la vivienda RQ04-14. Búsqueda en base a la disponibilidad de pago con tarjeta. RQ04-15. Búsqueda en base a la existencia del aval RQ04-16. Búsqueda en base a la admisión de mascotas en la vivienda RQ04-17. Búsqueda en base al periodo de estancia. RQ04-18. Búsqueda en base a si la vivienda está amueblada o no. RQ04-19. Búsqueda en base a la disponibilidad de wifi en la vivienda	
Descripción	La solución provee al usuario de una lista de viviendas que encajen con el formulario rellenado.	
Precondición	El usuario ha accedido a la página de vivienda	
Secuencia normal	Paso	Acción
	1	El usuario selecciona en el menú desplegable “Prefectura” la prefectura donde desea buscar las viviendas.
	2	El usuario presiona el botón “Siguiente” o el botón “2” referente al paso 2 del formulario.
	3	Opcionalmente, el usuario podrá elegir diversas opciones de los diferentes menús desplegables disponibles tales como “Alquiler mínimo”, “Alquiler máximo”, “Metros cuadrados”, “Habitaciones”, “Tipo de construcción”, “Distancia a la estación”, “Antigüedad” así como la opciones de marcado tales como “Bajos costes iniciales”, “Sin aval”, Sin señal inicial”, “Admiten mascotas”, “Sin deposito”, “Periodo de estancia corto”, “Sin gasto de agencia”, “Amueblado”, “Internet” y “WIFI”.
	4	El usuario presiona el botón “Siguiente” o el botón “3” referente al paso 3 del formulario.
	5	El cliente envía una petición HTTP asíncrona al controlador

		REST para obtener el listado de viviendas que encajen con los parámetros introducidos en el formulario por parte del usuario.
6		El controlador ejecuta la función del modelo asociada al recurso accedido (buscar viviendas).
7		El sistema inicia el proceso de crawling (construcción de URL de búsqueda, crawling de la web, almacenamiento de información, carga en json).
8		El sistema devuelve el JSON generado con la información del listado de viviendas al controlador
9		El controlador devuelve el JSON generado en el cuerpo de la respuesta y en la cabecera devuelve el estado HTTP 200 OK
10		El cliente carga el JSON transformando la información contenida para una correcta visualización.
Postcondición	El cliente carga la transformación en un cuadro previamente invisible inmediatamente para la visualización del usuario.	
Excepciones	Paso	Acción
	5	Si la petición excede un tiempo de 5 segundos se cancela y se muestra al usuario un cuadro de texto informando que no ha sido posible conectarse con el servidor de datos.
	7	Si el sistema no puede ejecutar el proceso de crawling el controlador devuelve estado 404 ERROR en la cabecera de la petición. El cliente devolverá error.
Frecuencia	Media	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 19 - UC-GR04

Diagrama de secuencia

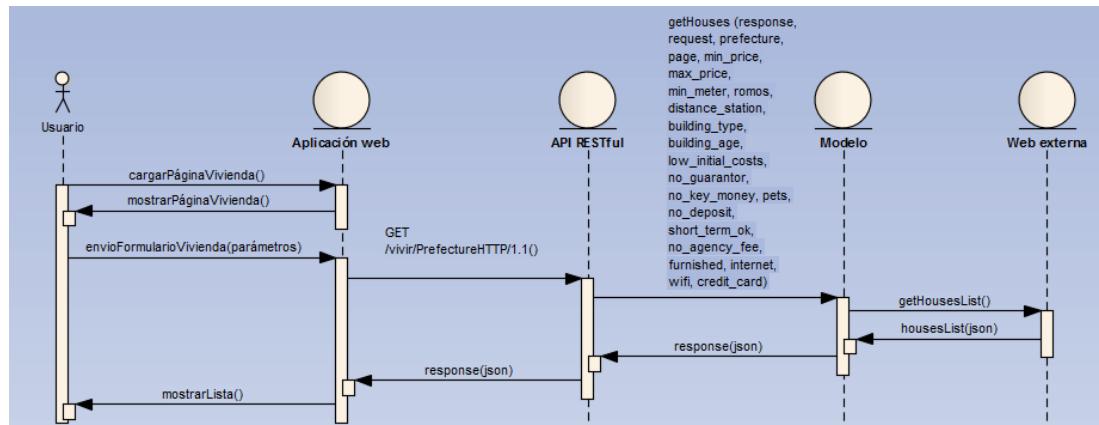


Figura 16 - Diagrama de secuencia UC-GR04

3.2 DIAGRAMA DE PAQUETES SISTEMA – SERVIDOR

A continuación se puede observar el conjunto de paquetes que comprenden la aplicación backend del proyecto.

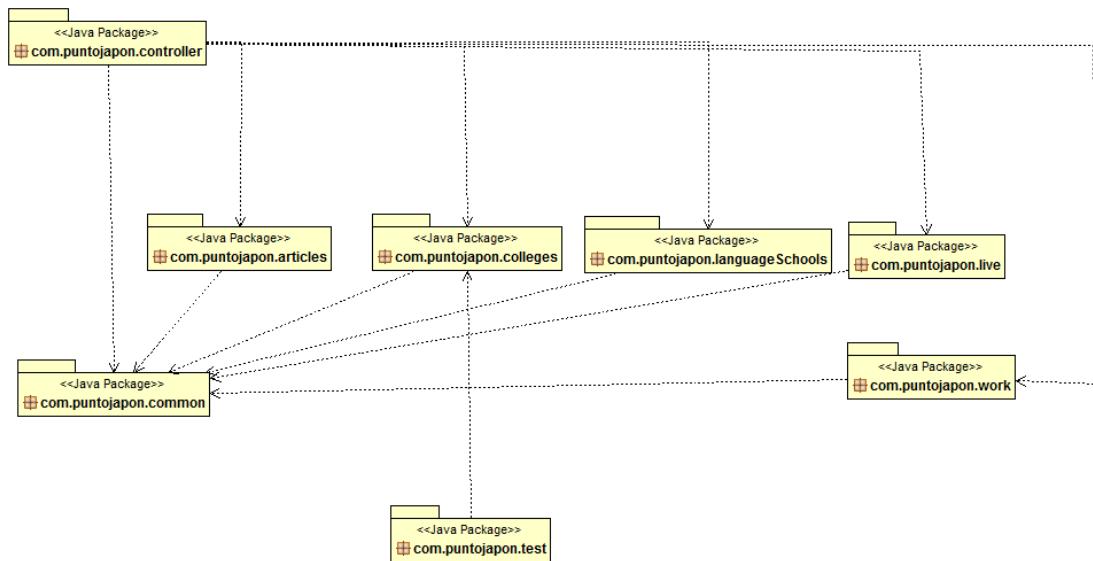


Figura 17 - Diagrama de paquetes

3.3 DIAGRAMA DE ACTIVIDADES SISTEMA - SERVIDOR

A continuación se puede observar el conjunto de clases que comprenden el conjunto de paquetes de la aplicación backend del proyecto.

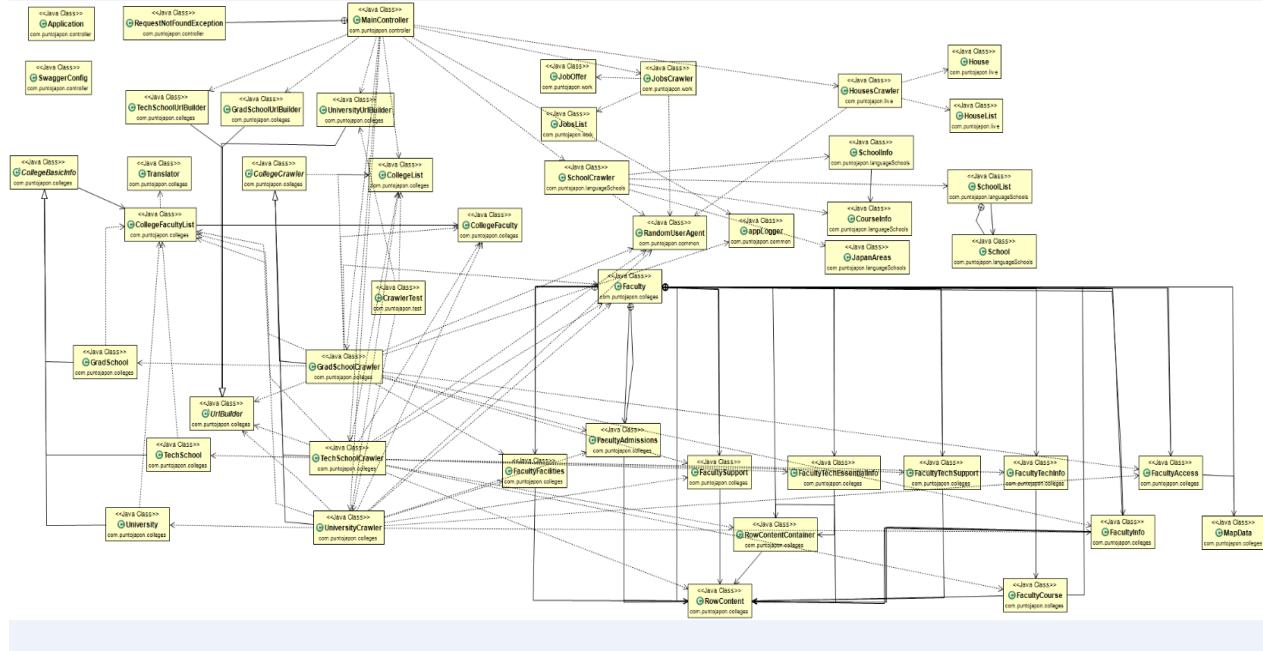


Figura 18 - Diagrama de actividades

3.4 DIAGRAMA DE DESPLIEGUE

A continuación se puede observar el despliegue del hardware necesario en el sistema y su disposición.

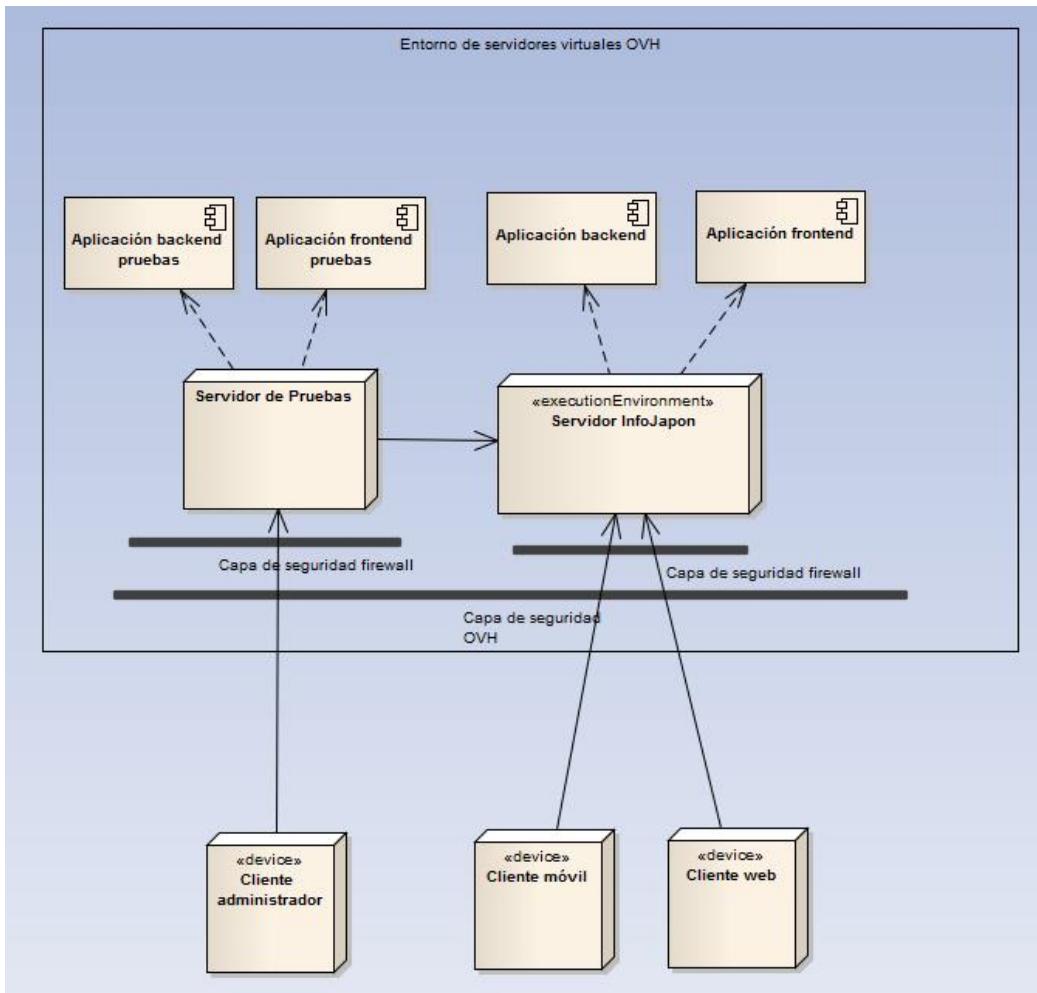


Figura 19 - Diagrama de despliegue

Capítulo 4. DESARROLLO

Para el desarrollo de la aplicación se decidió desde un principio aplicar el patrón de arquitectura software MVC (Modelo – Vista – Controlador). El patrón MVC permite separar la lógica del negocio de la interfaz de usuario y el controlador encargado de gestionar las comunicaciones y eventos entre ambos.

Se empezó por tanto desarrollando el modelo que contendría la lógica del negocio y a continuación el controlador que, en este caso, se hizo el desarrollo de una API REST. La arquitectura REST permite:

- Gestionar los eventos y comunicaciones del cliente con el servidor mediante peticiones HTTP.
- Permite la asociación de operaciones bien definidas con las funciones del modelo.
- Permite el uso de hipermedios en las comunicaciones (XML, JSON, HTML...).

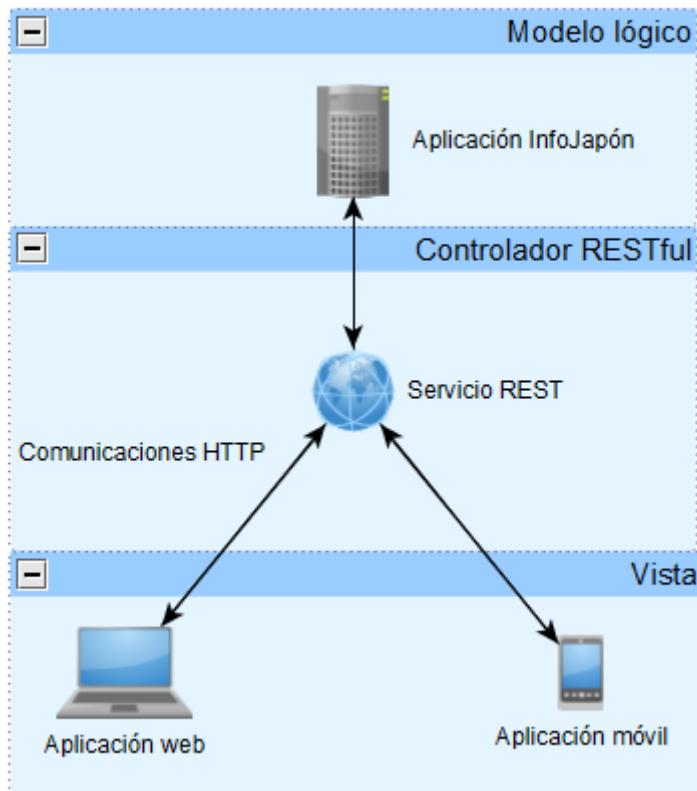


Figura 20 - Arquitectura de la solución

La estructura del proyecto es la siguiente:

Recurso	Contiene	Descripción
src/main/java	Paquetes: <ul style="list-style-type: none"> • com.puntojapon.articles • com.puntojapon.colleges • com.puntojapon.common • com.puntojapon.languageSchools • com.puntojapon.live • com.puntojapon.work 	Contiene todo el código fuente del modelo lógico de la aplicación
src/main/resources	Ficheros y carpetas: <ul style="list-style-type: none"> • /articles • indexBlogs.sh • log4j.properties • RSStojson.ktr • Sort.json 	Contiene las carpetas y ficheros referentes a las funciones de los artículos así como el fichero de configuración de log4j
src/test/java	Paquetes: <ul style="list-style-type: none"> • com.puntojapon.test 	Contiene el código fuente de la batería de test
src/controller	Paquetes: <ul style="list-style-type: none"> • com.puntojapon.controller 	Contiene el código fuente del controlador REST de la aplicación
src/web	Recursos: <ul style="list-style-type: none"> • css/ • estudiar/ • img/ • js/ • estudios.html • trabajo.html • vivienda.html • index.html 	Contiene el código fuente del cliente web.
target/		Contiene los archivos salida tras compilar la aplicación

app.log		Fichero log principal de la aplicación. Registra los eventos sucedidos.
application.properties		Fichero de propiedades de spring-boot
pom.xml		Fichero principal de configuración del proyecto gestionado por Apache Maven.

Tabla 20 - Estructura proyecto

4.1 DESARROLLO BACKEND

Para el desarrollo del backend antes de nada hubo que decidir que lenguaje de programación utilizar ya que esta decisión marcaría el devenir del proyecto. La duda estuvo entre desarrollar el backend con Java EE utilizando el framework MVC Spring o desarrollarlo con Ruby utilizando el framework web Ruby on Rails. [17]

Al final la decisión se decantó por el uso de Java con el framework Spring debido a los siguientes puntos:

- El desarrollador tenía más experiencia en java y con las librerías de terceros que se iban a utilizar a lo largo del proyecto.
- Escalabilidad más eficiente debido al rendimiento de la JVM.

4.1.1 LIBRERÍAS UTILIZADAS

Para el desarrollo del modelo y el controlador se han utilizado las siguientes librerías de terceros gestionadas con Apache Maven:

Librería	Descripción
<ul style="list-style-type: none">• Kettle-core 5.3.0.0-200• Kettle-dbdialog 5.3.0.0-200• Kettle-engine 5.3.0.0-200• Kettle-ui-swt 5.3.0.0-200• Kettle-log4j-plugin 5.3.0.0-200	Librerías de Kettle para la ejecución de la tarea asociada al proceso ETL de los artículos.
<ul style="list-style-type: none">• Feed4j	Librería para obtener feed de fuentes RSS. Utilizada en la extracción de información del proceso ETL de artículos.
<ul style="list-style-type: none">• UserAgentUtilz 1.19	Presenta un conjunto de herramientas para la manipulación de cabeceras HTTP. Utilizadas para modificar las cabeceras de respuesta y obtener información de las peticiones HTTP.
<ul style="list-style-type: none">• Springfox-swagger2 2.2.2• Springfox-swagger-ui 2.2.2	Librerías utilizadas para la generación de documentación de la API REST.
<ul style="list-style-type: none">• Gson 2.5	Librería para la gestión y manipulación de objetos JSON.
<ul style="list-style-type: none">• Spring-boot-starter-web	Librería utilizada para el despliegue del servicio backend.
<ul style="list-style-type: none">• Jsoup 1.8.2	Librería utilizada para la extracción de información de fuentes HTML.
<ul style="list-style-type: none">• Log4j 1.2.17	Librería utilizada para la manipulación

	concurrente del log en tiempo real.
<ul style="list-style-type: none">• Spring-context• Spring-aop• Spring-webmvc• Spring-web	Librerías del framework Spring utilizado por el controlador REST.

Tabla 21 - Librerías utilizadas en el modelo y controlador

4.1.2 DESARROLLO DEL MODELO

El modelo lógico de la aplicación abarca el conjunto de funciones que cumplen con los objetivos y alcance especificados. Por tanto podemos diferenciar el proceso ETL de los artículos y su gestión y los procesos crawling que permiten obtener información específica de fuentes HTML definidas.

4.1.2.1 DESARROLLO DE LOS PROCESOS RELACIONADOS CON LOS ARTÍCULOS

El objetivo principal de este proceso era mostrar en un listado los artículos de los diferentes blogs adscritos a la página de manera cronológica y resumida que permitiera al usuario acceder al artículo completo mediante un enlace.

Para abordar este problema se escogió trabajar con el conjunto de herramientas de integración de datos Pentaho, concretamente con la herramienta denominada Kettle para la extracción y transformación de los artículos y Elastic Search para la carga y almacenaje de los mismos.

4.1.2.1.1 EXTRACCIÓN Y TRANSFORMACIÓN DE DATOS

Kettle es una herramienta que permite, de manera visual y amigable, la creación de procesos de extracción, transformación y carga de datos. Posee además una API escrita en Java por lo que permitía automatizar las transformaciones. De esta manera se crea una transformación con la herramienta visual, se guardó en un fichero con formato .ktr y, en java en el backend se enlazó la transformación a un recurso de la API REST para que, cada vez que fuera llamado, se iniciara la transformación.

Esta transformación se encarga de leer las fuentes RSS de cada blog adscrito, obtener los artículos (título, resumen, fecha de publicación) y guardar en un fichero JSON todos los artículos por cada blog. A su vez, tras generar un JSON por cada blog, se utilizarán todo el pool de artículos para generar un JSON general que almacene todos los artículos de todos los blogs por orden alfabético.

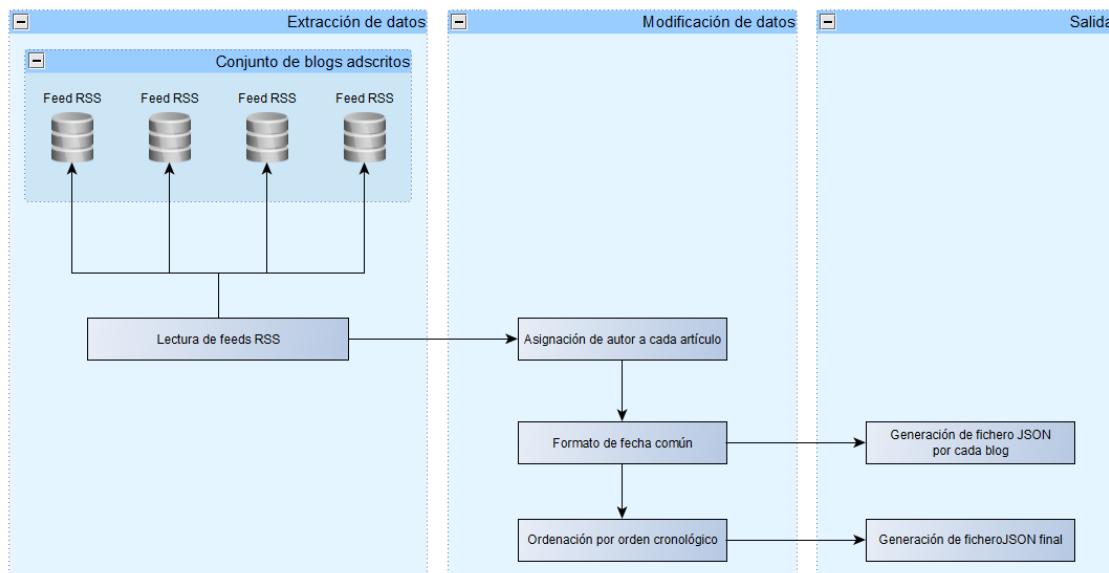


Figura 21 - Proceso ETL de artículos

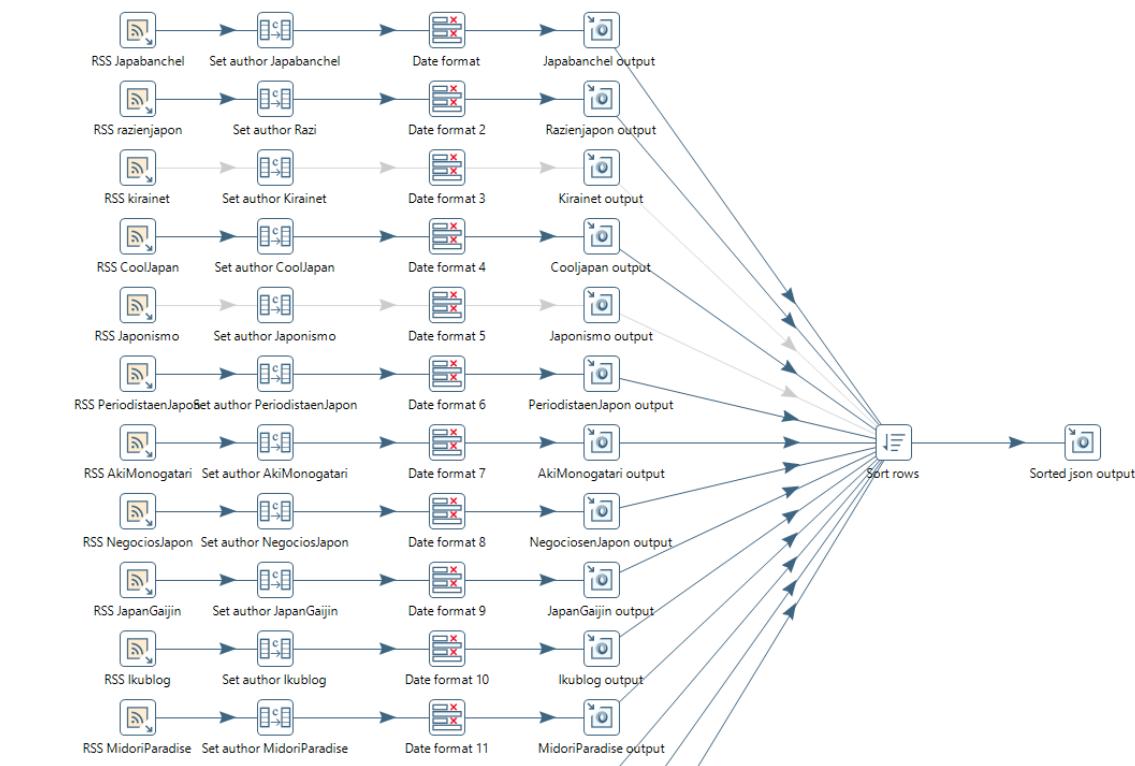


Figura 22 - Transformación Kettle para la extracción de archivos

Podemos identificar entonces:

- **Cajas RSS** – Obtienen la información del feed RSS de cada blog
- **Cajas setAuthor** – Fijan el autor al feed extraído
- **Cajas dateFormat** – Cambian el formato de la fecha para que todos los artículos tengan el mismo. Esto es necesario para poder ordenar todos los artículos juntos

- **Cajas de blog output** – Generan un JSON con el nombre de los blogs. Estos JSON serán los que se indexen (carguen) en el Elastic Search para, posteriormente, poder hacer búsquedas sobre los artículos.
- **Caja sort rows** – Ordena cronológicamente todos los artículos de todos los feeds RSS
- **Caja sorted json output** – Genera un fichero JSON con todos los artículos ordenados cronológicamente. Este fichero será el utilizado para visualizar los artículos en el frontend

Una vez el fichero de la transformación está preparado, fue necesario el poder ejecutar la transformación desde el backend. Para ello, el paquete com.puntojapon.articles contiene el conjunto de funciones para la ejecución de la transformación, la búsqueda sobre artículos y la carga de los datos en ElasticSearch.

4.1.2.1.2 CARGA DE DATOS

Para poder habilitar las búsquedas sobre los artículos es necesario guardar los archivos JSON de cada blog en una base de datos de documentos. Elastic Search permite realizar búsquedas sobre documentos indexados proporcionando un gran rendimiento. [18]

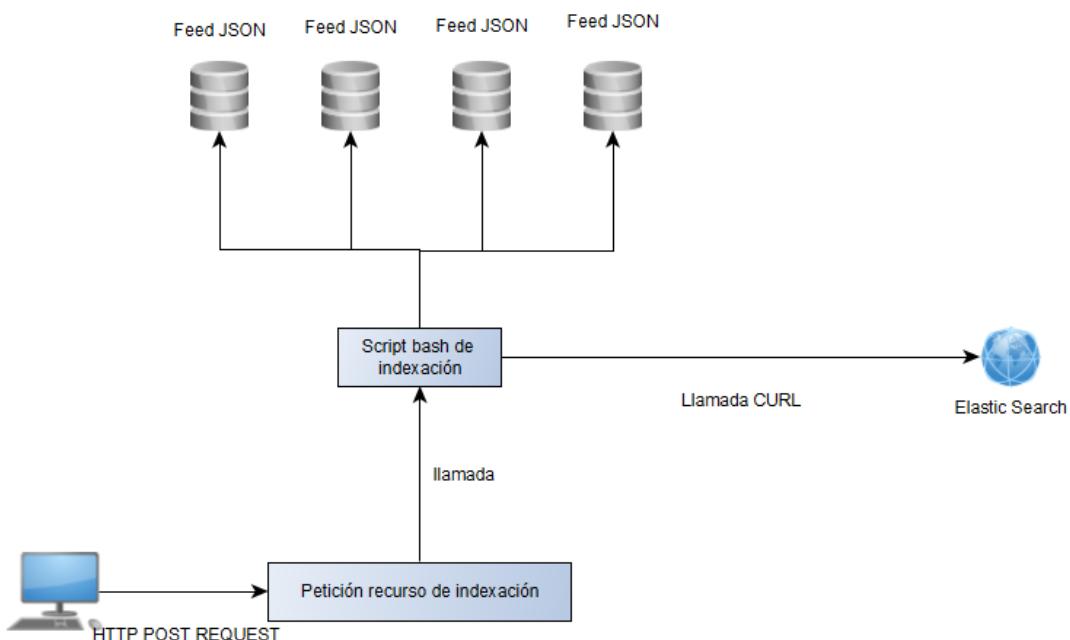


Figura 23 - Diagrama de carga de artículos en Elastic Search

De esta manera cada vez que se llamará al recurso de la API REST de actualizar repositorio, se ejecutaría el script bash que, mediante llamadas CURL indexa los ficheros JSON de cada blog en el repositorio de blogs/artículos de Elastic Search

```
#!/bin/bash

COUNTER=1
for file in src/main/resources/articles/*.json; do
DATA=$(less $file)
curl -XPUT 'http://91.134.143.80:9200/blogs/articulos/'$COUNTER -d "$DATA" -vn
sleep 4
COUNTER=$((COUNTER+1))
done
echo "finished"
```

Figura 24 - Fichero para la indexación de ficheros JSON en Elastic Search

El script en bash permite recorrer la carpeta de los ficheros JSON y enviarlos en una petición PUT mediante el proceso del servidor CURL al servicio de ElasticSearch.

Concretamente, el script bash se ejecuta mediante el cron del servidor cada hora para mantener el pool de artículos de elasticSearch actualizado.

```
# m h dom mon dow    command
0 * * * * curl -X POST http://www.infojapon.com:8081/articulos/actualizarRepositorio
```

Figura 25 - Configuración para la actualización del repositorio de artículos

4.1.2.1.3 MOSTRAR ARTÍCULOS

Las peticiones de mostrar artículos siempre tendrán como respuesta (salida) el JSON de los artículos ordenados cronológicamente.

4.1.2.1.4 BÚSQUEDA SOBRE ARTÍCULOS

La búsqueda sobre los artículos no es algo trivial ya que ElasticSearch en primera estancia no está hecho para poder buscar sobre sub-objetos ya que necesita de una consulta avanzada [19]. Fijémonos en un JSON de un blog de ejemplo:

```
{
  "data": [
    {
      "Description": "La ciudad de Shimabara (島原市). Shimabara es una ciudad japonesa situada en la isla sur de Japón de Kyushu, concretamente en la prefectura de Nagasa",
      "PubDate": "28/02/16 11:25",
      "Author": "Japabanchel",
      "Title": "Visitando la ciudad de Shimabara (島原市) en Nagasaki.",
      "Link": "http://japabanchel.com/2016/02/28/shimabara-unzen-obana-nehanzou-buda-onsen-koi-no-mizube-michi-nampuro/"
    },
    {
      "Description": "El Diciembre del pasado año visité por primera vez la isla de Kyushu, al sur de Japón. Me decanté por Nagasaki y desde allí hice también una escapada",
      "PubDate": "21/02/16 15:54",
      "Author": "Japabanchel",
      "Title": "Viaje Nagasaki (長崎), prefectura de Kyūshū (九州).",
      "Link": "http://japabanchel.com/2016/02/21/nagasaki-kyushu-sasebo/"
    },
    {
      "Description": "Después de más de tres años viviendo en Japón ya puedo escribir con cierta seguridad sobre los tipos de restaurantes que más me gustan cuando salgo a",
      "PubDate": "30/01/16 08:46",
      "Author": "Japabanchel",
      "Title": "Guía para saber qué comer cuando visitas Japón",
      "Link": "http://japabanchel.com/2016/01/30/comida-de-japon-ramen-yakitori-yakiniku-okonomiyaki-sushi-tempura-gyoza-curry/"
    },
    {
      "Description": "18 de Enero de 2016, me levanto de la cama y al abrir la persiana automática del salón descubro con asombro como la nieve cubre el pequeño jardín de",
      "PubDate": "19/01/16 11:23",
      "Author": "Japabanchel",
      "Title": "Relato de la nieve y el caos del transporte en Japón.",
      "Link": "http://japabanchel.com/2016/01/19/nieve-japon-caos-transporte-relato/"
    },
    {
      "Description": "¡Qué rico el roscón de reyes y qué ricos los donuts de Mr. Donuts! Así hasta que te empieza a doler una muña y la idea de visitar a \u201ctu gran am",
      "PubDate": "11/01/16 10:43",
      "Author": "Japabanchel",
      "Title": "Visita al dentista en Japón.",
      "Link": "http://japabanchel.com/2016/01/11/visita-yendo-dentista-en-japon-caries-endodoncia/"
    }
  ]
}
```

Figura 26 - Fichero JSON con los artículos de un blog específico

Podemos observar que data es un array de artículos. Por defecto hay de crear un índice manualmente en ElasticSearch donde se especifique que los objetos dentro del Array “Data” son nestedObjects ya que no pueden ser buscados directamente al pertenecer a un documento oculto generado por el Elastic Search. [20]

Por tanto el índice creado manualmente (mediante curl y la API REST de ES) es el siguiente:

```
{
  "mappings": {
    "articulos": {
      "properties": {
        "data": {
          "type": "nested",
          "properties": {
            "Description": {
              "type": "string"
            },
            "PubDate": {
              "type": "string"
            },
            "Author": {
              "type": "string"
            },
            "Title": {
              "type": "string"
            },
            "Link": {
              "type": "string"
            }
          }
        }
      }
    }
  }
}
```

Figura 27 - JSON para la creación de un índice en ElasticSearch

Una vez habilitada la búsqueda sobre sub-objetos fue necesario establecer la query para que buscara en los sub-objetos (inner objects) [21] [22]:

```
"query": {
  "nested": {
    "path": "data",
    "query": {
      "match": {
        "data.Title": "Renovar el visado japonés de cónyuge (y otros) en Inmigración."
      }
    },
    "inner_hits": {}
  }
}
```

Figura 28 - JSON de consulta sobre Elastic Search

De esta manera, al ejecutar la query los artículos que hicieran match con el título expuesto iban a ser aquellos devueltos dentro del array “inner_hits” ya que en el array de “data” principal saldrán todos los artículos del blog.

```
  }
},
"inner_hits": {
  "data": {
    "hits": {
      "total": 2,
      "max_score": 1.4962151,
      "hits": [
        {
          "_index": "blogs",
          "_type": "articulos",
          "_id": "11",
          "_nested": {
            "field": "data",
            "offset": 4
          },
          "_score": 1.4962151,
          "_source": {
            "Description": "¿Qué es lo que ocurre el día que un MMORPG cierra sus servidores y a comunicaci\u00f3n, y el anime es uno de ellos. Teniendo series de tem\u00e1tica similar, tanto la popular Sword Ar",
            "PubDate": "23/06/16 14:24",
            "Author": "Pesodelaire",
            "Title": "Overlord, un anime sobrevalorado",
            "Link": "http://www.elpesodelaire.com/2016/06/overlord.html"
          }
        },
        {
          "_index": "blogs",
          "_type": "articulos",
          "_id": "11",
          "_nested": {
            "field": "data",
            "offset": 5
          },
          "_score": 1.3091882,
          "_source": {
            "Description": "Se acerca el verano y con \u00e9l, una cantidad ingente de estrenos de animemos querido hacer una selecci\u00f3n de los que, a nuestro parecer, son los m\u00e1s interesantes. Eso s\u00ed, tendr",
            "PubDate": "22/06/16 09:47",
            "Author": "Pesodelaire",
            "Title": "Estrenos de anime: Verano 2016",
            "Link": "http://www.elpesodelaire.com/2016/06/estrenos-anime-verano-2016.html"
          }
        }
      ]
    }
  }
}
```

Figura 29 - JSON respuesta a consulta en Elastic Search

Por tanto, se implementó en java una clase que llamará al proceso [23] del servidor CURL y, dinámicamente, ejecutara la consulta según el título que el usuario final quiera buscar en los artículos y luego, el sistema frontend se encarga de recoger solo los artículos dentro del array "Inner hits".

```
ProcessBuilder processBuilder = new ProcessBuilder("curl", "-s", "-XPOST",
    "http://91.134.143.80:9200/blogs/articulos/_search", "-d", query);
```

Figura 30 - Construcción de consulta a Elastic Search en java

4.1.2.2 DESARROLLO DE LOS PROCESOS DE EXTRACCIÓN DE DATOS (CRAWLING)

En los procesos crawling la entrada es el conjunto de parámetros de la búsqueda y la salida esperada es un JSON con los resultados de la búsqueda. Pese a que el proceso de extracción de información por parte del agente inteligente desarrollado sigue un mismo patrón para obtener y guardar la información, el objeto JSON generado es diferente.

El JSON generado cuenta con dos partes:

- Cabecera de búsqueda (1): Permite saber parámetros referentes a la búsqueda tales como si ha sido fallida o no. Cuantos resultados se han obtenido y cuál ha sido el tipo de búsqueda.
- Resultados de la búsqueda (2): Muestra un array con todos los resultados obtenidos



```
{
  "searchType": "Houses search",
  "searchFound": 15,
  "page": 1,
  "hasNextPage": true,
  "searchState": true,
  "houseList": [
    {
      "buildingType": "Shared Guesthouse - Guest House",
      "place": "in Shiraitodai Fuchu-shi, Tokyo",
      "rent": "Rent ¥40,000",
      "size": "Size 13.00 m\u00b2",
      "deposit": "Deposit ¥0",
      "keyMoney": "¥0",
      "floor": "Floor 3 / 4F",
      "maintenanceFee": "Maintenance Fee ¥10,000 / mth",
      "nearestStation": "Musashinodai Station (5 min. walk)",
      "agent": "Irodori Factory",
      "imageUrl": "https://www.realestate.co.jp/image/2673/5238587-FFFFFF-280x180-c.jpg",
      "map": "東京都府中市白糸台5-38-1",
      "link": "https://www.realestate.co.jp/en/rent/view/330389"
    },
    {
      "buildingType": "Private Guesthouse - Guest House",
      "place": "in Honisshiki Edogawa-ku, Tokyo",
      "rent": "Rent ¥39,000",
      "size": "Size 10.00 m\u00b2",
      "deposit": "Deposit ¥0",
      "keyMoney": "¥0",
      "floor": "Floor 2 / 2F",
      "maintenanceFee": "Maintenance Fee ¥10,000 / mth",
      "nearestStation": "Shinkoiwa Station (13 min. walk)",
      "agent": "Yours Corporation Co.,Ltd.",
      "imageUrl": "https://www.realestate.co.jp/image/2629/5355335-FFFFFF-280x180-c.jpg",
      "map": "東京都江戸川区本一色1-21-16",
      "link": "https://www.realestate.co.jp/en/rent/view/254473"
    }
  ]
}
```

Figura 31 - JSON ejemplo sobre las respuestas en JSON de la solución

El proceso al recibir una petición de búsqueda es el siguiente:

- Llamada a la creación del agente inteligente pasándole los parámetros de la búsqueda para identificar que funciones utilizar (búsqueda sobre estudios, vivienda...)
- El agente inteligente obtendrá el código fuente de una web determinada por el tipo de búsqueda y procederá a la extracción de datos basándose en un conjunto de expresiones regulares, consultas sobre el DOM y búsquedas XPATH predefinidas. [24]

```
// Phone and How to arrive
if (element.select("table.tableStyle04 > tbody > tr > td > table > tbody > tr").first()
    .nextElementSibling().select("td").first().nextElementSibling() != null) {
    languageSchool.setPhone(element.select("table.tableStyle04 > tbody > tr > td > table > tbody > tr")
        .first().nextElementSibling().select("td").first().nextElementSibling().text().trim());
}

languageSchool
    .setHowToArrive(element.select("table.tableStyle04 > tbody > tr > td > table > tbody > tr")
        .first().nextElementSibling().select("td").first().nextElementSibling()
        .nextElementSibling().text().trim());
}
```

Figura 32 - Proceso de búsqueda de datos para su extracción mediante consultas del árbol DOM

- Una vez extraídos los datos se almacenarán en objetos locales determinados por la búsqueda denominados objetos maestros.
- El conjunto de los datos maestros se volcarán en un string resultante en formato JSON que se utilizará como salida de la función.

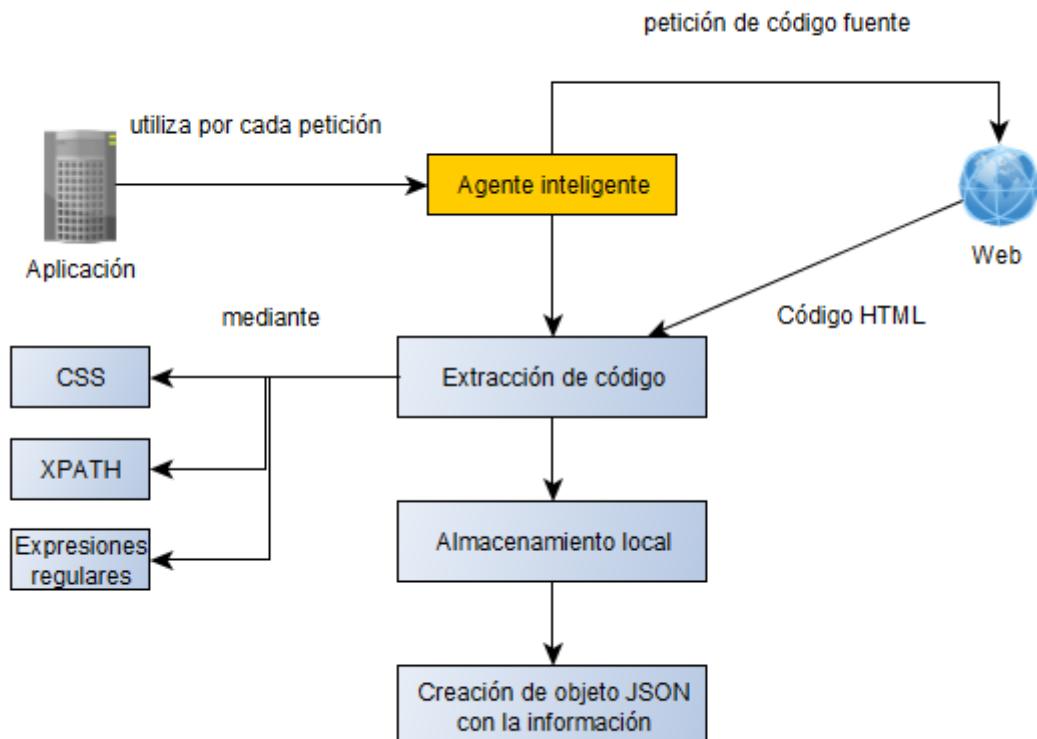


Figura 33 - Proceso de crawling

4.1.2.2.1 DESARROLLO DEL PAQUETE RELACIONADO CON LOS ESTUDIOS

El paquete com.puntojapon.colleges contiene el conjunto de funciones que permiten la creación de un agente inteligente cuyo objetivo es la extracción y carga de los datos referentes a los estudios de grado, posgrado y formación profesional. Debido a que estas tres búsquedas son sobre la misma web [25], siguen un patrón bien definido (ver diagrama de clases). Este patrón se definió en una fase de análisis de las webs que contendrían la información a extraer. Se buscaron similitudes en las búsquedas para una mayor modularidad por lo tanto para las 3 búsquedas nombradas anteriormente existen las siguientes clases:

- Clase de almacenamiento de datos maestros
- Clase para el montaje de la URL y comprobación de datos de entrada válidos
- Clase para el crawleo web



Figura 34 - Estructura de clases de crawler

Pese a que sigan un patrón definido (lista de escuelas – escuelas – departamentos) los parámetros de búsqueda varían según el tipo de escuela. Además, la recopilación de datos es en profundidad, el agente inteligente navega a la primera página, obtiene los datos, los almacena y continúa a la siguiente página si existe de manera recursiva hasta que no haya más páginas [26].

```
if (next == false) { // no ha encontrado mas universidades en esta
    // pagina
    System.out.println("Colegios profesionales: " + getCollegeCounter());
    return jsonTechSchoolsList;
} else {
    return jsonTechSchoolsList = getCollegeList(nextPageString, prefectureSearchName, TechSchoolsList,
        jsonTechSchoolsList, counter, remoteIp);
}
```

Figura 35 - Implementación de algoritmo divide y vencerás en el proceso crawling

Por otro lado el paquete com.puntojapon.languageschools contiene el conjunto de funciones para buscar específicamente colegios de lengua japonesa ya que obtiene la información de <http://www.nisshinkyo.org/> por lo que el patrón de extracción de información es completamente diferente.

4.1.2.2.1.1 UNIVERSIDADES

El bloque de funciones referente a universidades es el siguiente:

4.1.2.2.1.1.1 BÚSQUEDA DE UNIVERSIDADES

En los parámetros de búsqueda de universidades el usuario indica lo siguiente:

- Prefectura
- Especialidad
- Nombre de universidad (opcional)
- Tipo de universidad (opcional)
- Mes de admisión (opcional)
- Fecha límite (mes) (opcional)
- Admisión de aplicaciones (opcional)
- Certificación de inglés necesaria (opcional)
- Examen de entrada (EJU) (Opcional)

A continuación teniendo en cuenta los parámetros de entrada, el agente inteligente utilizará los parámetros como parámetros de búsqueda en la web a crawlear armando una URL siguiendo la codificación de la página. Una vez armada la URL el agente inteligente la parseará extrayendo las universidades de cada página de la búsqueda.

Ejemplo:

URL:

<http://www.infojapon.com:8081/universidades/Tokyo/Language?nameUni=&typeUni=&admissionMonth=&deadLine=&eju=&engExam=&admisionUni=>

Respuesta:

```
{
  "searchType": "Tokyo Language",
  "searchBrand": "56",
  "searchState": true,
  "collegeList": [
    {
      "search": true,
      "id": "/en/univ/359/",
      "japaneseName": "明治学院大学",
      "name": "Meiji Gakuin University",
      "prefecture": "Tokyo, Kanagawa",
      "type": "Private",
      "imageUrl": "http://www.ipss.jp/uploads/univ/359/main.jpg",
      "guideUrl": "",
      "title": "150 years of experience in educating for the future",
      "description": "Meiji Gakuin University traces its origins to the Hepburn Academy—an English school founded by the American missionary James C. Hepburn in 1863. The University will celebrate its 150th anniversary ... [Meiji Gakuin University Show more]",
      "officialUrl": "",
      "faculties": [
        "collegeName": "Meiji Gakuin University",
        "collegeFacultyList": [
          {
            "facultyName": "Faculty of Letters",
            "facultyUrl": "/en/univ/359/930/"
          },
          {
            "facultyName": "Faculty of International Studies",
            "facultyUrl": "/en/univ/359/934/"
          }
        ]
      ],
      "place": [
        {
          "search": true,
          "id": "/en/univ/352/",
          "japaneseName": "法政大学",
          "name": "Sophia University",
          "prefecture": "Tokyo",
          "type": "Private",
          "imageUrl": "http://www.ipss.jp/uploads/univ/352/main.jpg",
          "guideUrl": "http://www.ipss.jp/uploads/univ/352/guide.zip",
          "title": "Define yourself and your own future.",
          "description": "Global society is constantly in motion, driven by the various people of the world, who all hold their own cultures and ideas. In order to take one's place in this evolving global society, it is ... [Hosei University Show more]",
          "officialUrl": "",
          "faculties": [
            "collegeName": "Hosei University",
            "collegeFacultyList": [
              ...
            ]
          ]
        }
      ]
    }
  ]
}
```

Figura 36 - Respuesta JSON con lista de universidades

4.1.2.2.1.1.2 INFORMACIÓN DE UNIVERSIDAD

Para obtener la información de la universidad el proceso es transparente para el usuario. Dada una ID de universidad el crawler visitará la página específica de esa universidad y extraerá la información pertinente referente a la universidad (nombre, dirección, grados, departamentos...)

Ejemplo:

URL: <http://jdecastroc.ovh:8081/universidades/id/388>

Respuesta:

```
{
  "search": true,
  "id": "http://www.jpac.jp/en/univ/388/",
  "japaneseName": "東洋英和女学院大学",
  "name": "Toyo Eiwa University",
  "prefecture": "Kanagawa, Tokyo",
  "type": "Private",
  "imageUrl": "/uploads/univ/388/main.jpg",
  "guideline": "東洋英和女学院大学",
  "description": "Toyo Eiwa University is located on the Yokohama Campus, about one hour by train from central Tokyo. There are approximately 2,300 students enrolled. The campus is endowed with rich natural resources, showcased by seasonal flora. Further, it boasts excellent facilities including a chapel, library, club house, Aqua-Exercise Center (containing a heated swimming pool with an adjacent training room), playing fields, floodlit tennis courts and golf practice range. Our aim is to provide a well-developed environment that encourages students to fully enjoy campus life.",
  "officialUrl": "http://www.toyoeiwa.ac.jp/daigaku/",
  "faculties": [
    {
      "collegeName": "Toyo Eiwa University",
      "collegeFacultyList": [
        {
          "facultyName": "Social Sciences",
          "facultyUrl": "/en/univ/388/1027/"
        },
        {
          "facultyName": "Human Sciences",
          "facultyUrl": "/en/univ/388/1026/"
        }
      ]
    }
}
```

Figura 37 - JSON respuesta con información de universidad

4.1.2.2.1.1.3 INFORMACIÓN DE DEPARTAMENTO

Para obtener la información de un departamento el proceso también es transparente para el usuario ya que, en la parte de la interfaz el sólo selecciona el departamento, sin embargo, dada una ID de universidad y un ID de departamento el crawler visitará las siguientes páginas relacionadas con el departamento en lugar de una sola página.:

- Página de acceso
- Página de admisiones
- Página de instalaciones
- Página de soporte al estudiante
- Página de información universitaria del departamento

Ejemplo:

Usuario final visita:

http://www.infojapon.com/estudiar/departamento?tipo=univ&id=388&id_depart=1026

Aplicación llama a:

- <http://www.infojapon.com:8081/universidades/id/388/1026/access>
- <http://www.infojapon.com:8081/universidades/id/388/1026/admissions>
- <http://www.infojapon.com:8081/universidades/id/388/1026/facilities>
- <http://www.infojapon.com:8081/universidades/id/388/1026/support>
- <http://www.infojapon.com:8081/universidades/id/388/1026/info>

4.1.2.2.1.2 COLEGIOS DE POSGRADO

El bloque de funciones referente a los colegios de posgrado es muy similar al de universidades:

4.1.2.2.1.2.1 BÚSQUEDA DE COLEGIOS DE POSGRADO

En los parámetros de búsqueda de universidades el usuario indica lo siguiente:

- Prefectura
- Tipo de estudio de posgrado
- Nombre de colegio (opcional)
- Idioma de las clases (opcional)
- Tipo de universidad (opcional)

Ejemplo:

URL:

<http://www.infojapon.com:8081/posgrado/Tokyo?nameGrad=&typeGrad=master&typeCourse=&englishCourse=>

Respuesta:

```
{
  "searchType": "Tokyomaster",
  "searchFound": 122,
  "searchState": true,
  "collegeList": [
    {
      "search": true,
      "id": "/en/grad/233/",
      "japaneseName": "国際基督教大学大学院(ICU)",
      "name": "International Christian University (ICU)",
      "prefecture": "Tokyo",
      "type": "Private",
      "imageUrl": "http://www.jpss.jp/uploads/grad/233/main.jpg",
      "guideUrl": "http://www.jpss.jp/uploads/grad/233/guide.zip",
      "title": "Developing Expertise through Advanced Liberal Studies",
      "description": "We nurture potential leaders who can bring Japan and the world together, trained through a bilingual English and Japanese education to become the highly skilled professionals.  
[«International Christian University (ICU)» Show more]",
      "officialUrl": "",
      "faculties": {
        "collegeName": "International Christian University (ICU)",
        "collegeFacultyList": [
          {
            "facultyName": "Graduate School of Arts and Sciences",
            "facultyUrl": "/en/grad/233/674/"
          }
        ]
      }
    },
    {
      "search": true,
      "id": "/en/grad/291/",
      "japaneseName": "法政大学",
      "name": "Hosei University",
      "prefecture": "Tokyo",
      "type": "Private",
      "imageUrl": "http://www.jpss.jp/uploads/grad/291/main.jpg",
      "guideUrl": "http://www.jpss.jp/uploads/grad/291/guide.zip",
      "title": "Go 10 years ahead in one year",
      "description": "The Graduate School at Hosei University was opened in 1951, offering Master's degree programmes in the Graduate School of Humanities and the Graduate School of Sociology. The Graduate Schools of Hosei... [«Hosei University» Show more]",
      "officialUrl": "",
      "faculties": {
        "collegeName": "Hosei University",
        "collegeFacultyList": [
        ]
      }
    }
  ]
}
```

Figura 38 - Respuesta JSON con lista de colegios de posgrado

4.1.2.2.1.2.2 INFORMACIÓN DE COLEGIO DE POSGRADO

Para obtener la información del colegio del posgrado el proceso vuelve a ser transparente para el usuario. Dada una ID de colegio de posgrado el crawler visitará la página específica de ese colegio y extraerá la información pertinente referente a la universidad (nombre, dirección, grados, departamentos...).

Ejemplo:

URL: <http://www.infojapon.com:8081/posgrado/id/240>

Respuesta:

```
{
  "search": true,
  "id": "http://www.jpss.jp/en/grad/240/",
  "japaneseName": "上智大学",
  "name": "Sophia University",
  "prefecture": "Tokyo",
  "type": "Private",
  "imageUrl": "/uploads/grad/240/main.jpg",
  "guideUrl": "",
  "title": "SOPHIA in the World",
  "description": "Founded by the Jesuits in 1913, Sophia University has been a pioneer in international education in Japan, attracting students from all over the world. In 1949 Sophia established the first program in Japan taught entirely in English, and it continues to lead the Japanese academic world in offering international programs. Sophia is located in central Tokyo, near Japan's political, financial, business, and cultural centers.",
  "officialUrl": "http://www.sophia.ac.jp/eng/e_top",
  "faculties": {
    "collegeName": "Sophia University",
    "collegeFacultyList": [
      {
        "facultyName": "Global Environmental Studies",
        "facultyUrl": "/en/grad/240/1765/"
      },
      {
        "facultyName": "Global Studies",
        "facultyUrl": "/en/grad/240/1865/"
      },
      {
        "facultyName": "Science and Technology",
        "facultyUrl": "/en/grad/240/700/"
      },
      {
        "facultyName": "Theology",
        "facultyUrl": "/en/grad/240/694/"
      },
      {
        "facultyName": "Philosophy",
        "facultyUrl": "/en/grad/240/695/"
      },
      {
        "facultyName": "Humanities",
        "facultyUrl": "/en/grad/240/696/"
      },
      {
        "facultyName": "Law",
        "facultyUrl": "/en/grad/240/697/"
      }
    ]
  }
}
```

Figura 39 - Respuesta JSON con información del colegio de posgrado

4.1.2.2.1.2.3 INFORMACIÓN DE DEPARTAMENTO

Para obtener la información de un departamento el proceso también es transparente para el usuario ya que, en la parte de la interfaz el sólo selecciona el departamento, sin embargo, dada una ID de colegio de posgrado y un ID de departamento el crawler visitará las siguientes páginas relacionadas con el departamento en lugar de una sola página tal y como ocurre con el crawler de universidades en su rama de extracción de información de departamento:

Ejemplo:

Usuario final visita:

http://www.infojapon.com/estudiar/departamento?tipo=grad&id=240&id_depart=1765

Aplicación llama a:

- <http://www.infojapon.com:8081/posgrado/id/240/1765/access>
- <http://www.infojapon.com:8081/posgrado/id/240/1765/admissions>
- <http://www.infojapon.com:8081/posgrado/id/240/1765/facilities>
- <http://www.infojapon.com:8081/posgrado/id/240/1765/support>
- <http://www.infojapon.com:8081/posgrado/id/240/1765/info>

4.1.2.2.1.3 COLEGIOS DE FORMACIÓN PROFESIONAL

El bloque de funciones referente a los colegios de posgrado es similar al de universidades y colegios de posgrado pero con un par de rasgos diferentes:

4.1.2.2.1.3.1 BÚSQUEDA DE COLEGIOS DE FORMACIÓN PROFESIONAL

En los parámetros de búsqueda de universidades el usuario indica lo siguiente:

- Prefectura
- Nombre de escuela (opcional)

Ejemplo:

URL: <http://www.infojapon.com:8081/fp/Tokyo?nameTech=>

Respuesta:

```
{
  "searchType": "Tokyo",
  "searchFound": 5,
  "searchState": true,
  "collegeList": [
    {
      "search": true,
      "id": "/en/tech/6/",
      "japaneseName": "東京観光専門学校",
      "name": "Tokyo Institute of Tourism",
      "prefecture": "Tokyo",
      "type": "",
      "imageUrl": "http://www.jpss.jp/uploads/tech/6/main.jpg",
      "guideUrl": "",
      "title": "We are an integrated school covering all fields of sightseeing services the tourism industry.",
      "description": "Our school, with a history of 43 years, aims at producing professionals such as interpreters and cafe or hotel staffs who will be internationally active in the tourism service industry. [«Tokyo Institute of Tourism» Show more]",
      "officialUrl": "",
      "faculties": {
        "collegeName": "Tokyo Institute of Tourism",
        "collegeFacultyList": [
          {
            "facultyName": "Interpreter Guide Course (two-year curriculum)",
            "facultyUrl": "/en/tech/6/faculty/#Fac60"
          },
          {
            "facultyName": "Bridal Business Course (two-year curriculum)",
            "facultyUrl": "/en/tech/6/faculty/#Fac61"
          },
          {
            "facultyName": "Hotel Course (two-year curriculum)",
            "facultyUrl": "/en/tech/6/faculty/#Fac62"
          },
          {
            "facultyName": "Tourism Course (two-year curriculum)",
            "facultyUrl": "/en/tech/6/faculty/#Fac63"
          },
          {
            "facultyName": "Airline Service Course (two-year curriculum)",
            "facultyUrl": "/en/tech/6/faculty/#Fac64"
          },
          {
            "facultyName": "Food Coordinator & Sommelier Course (two-year curriculum)",
            "facultyUrl": "/en/tech/6/faculty/#Fac65"
          }
        ]
      }
    }
  ]
}
```

Figura 40 - Respuesta JSON con lista de colegios de formación profesional

4.1.2.2.1.3.2 INFORMACIÓN DE COLEGIO DE FORMACIÓN PROFESIONAL

Para obtener la información del colegio del formación profesional el proceso vuelve a ser transparente para el usuario. Dada una ID de colegio de formación profesional el crawler visitará la página específica de ese colegio y extraerá la información pertinente referente al colegio (nombre, dirección, grados, departamentos...).

Ejemplo:

URL: <http://www.infojapon.com:8081/fp/id/6>

Respuesta:

```
{
  "search": true,
  "id": "http://www.jpss.jp/en/tech/6/",
  "japaneseName": "東京観光専門学校",
  "name": "Tokyo Institute of Tourism",
  "prefecture": "Tokyo",
  "type": "Professional training college",
  "imageUrl": "/uploads/tech/6/main.jpg",
  "guideUrl": "",
  "title": "We are an integrated school covering all fields of sightseeing services the tourism industry.",
  "description": "Our school, with a history of 43 years, aims at producing professionals such as interpreters and cafe or hotel staffs who will be internationally active in the tourism service industry.",
  "officialUrl": "http://www.tit.ac.jp/",
  "faculties": {
    "collegeName": "Tokyo Institute of Tourism",
    "collegeFacultyList": []
  }
}
```

Figura 41 - Respuesta JSON con información de colegio profesional

4.1.2.2.1.3.3 INFORMACIÓN DE DEPARTAMENTO DE COLEGIO DE FORMACIÓN PROFESIONAL

Por disposición de la propia página web de origen de los datos, los departamentos se muestran todos en la misma página. Una vez accedes a uno se obtiene la información de todos ya que no contienen tanta información como podría ser un departamento universitario. El proceso vuelve a ser transparente para el usuario y permite saber todos los cursos que se llevan a cabo en el colegio de formación profesional.

Tal y como ocurre al obtener información de departamentos en los anteriores módulos, al obtener la información del departamento del colegio de formación profesional el usuario llama a un recurso y el recurso llama a tres funciones diferenciadas para montar la información.

Ejemplo:

Usuario final visita:

http://www.infojapon.com/estudiar/departamento?tipo=tech&id=5&id_depart=Fac49

Aplicación llama a:

- <http://www.infojapon.com:8081/fp/id/5/admissions>
- <http://www.infojapon.com:8081/fp/id/5/info>
- <http://www.infojapon.com:8081/fp/id/5/support>

4.1.2.2.1.4 COLEGIOS DE LENGUA JAPONESA

El conjunto de funciones de lengua japonesa está englobado en otro paquete denominado com.puntojapon.languageSchools aunque el funcionamiento es parecido al de los anteriores módulos (montaje de URL, extracción de información, salida de información).

4.1.2.2.1.4.1 BÚSQUEDA DE COLEGIOS DE LENGUA JAPONESA

En los parámetros de búsqueda de colegios de lengua japonesa el usuario sólo indica la prefectura y el sistema le devuelve una lista con todas las escuelas de japonés de esa prefectura.

Ejemplo:

URL: <http://www.infojapon.com:8081/escuelasIdiomas/tokyo>

Respuesta:

```
{
  "search": true,
  "schoolArea": "tokyo",
  "schoolCounter": 121,
  "schoolList": [
    {
      "id": "88",
      "name": "KUDAN INSTITUTE OF JAPANESE LANGUAGE AND CULTURE",
      "address": "Teitomisakicho Bidg., 2-7-10 Misaki-cho, Chiyoda-ku, Tokyo 101-0061"
    },
    {
      "id": "89",
      "name": "YMCA TOKYO JAPANESE LANGUAGE SCHOOL",
      "address": "2-5-5 Sarugaku-cho, Chiyoda-ku, Tokyo 101-0064"
    },
    {
      "id": "91",
      "name": "THE INTERNATIONAL INSTITUTE OF JAPANESE LANGUAGE",
      "address": "3-36-4 Nihonbashi Hamacho, Chuo-ku, Tokyo 103-0007"
    },
    {
      "id": "503",
      "name": "TOKYO NIHONBASHI INSTITUTE OF FOREIGN LANGUAGE",
      "address": "2-34-11 Nihonbashi Ningyocho, Chuo-ku, Tokyo 103-0013"
    },
    {
      "id": "93",
      "name": "TOKYO JAPANESE LANGUAGE CENTER",
      "address": "3-5-4 Shiba Kouen, Minato-ku, Tokyo 105-0011"
    },
    {
      "id": "94",
      "name": "UJS LANGUAGE INSTITUTE",
      "address": "3-3-29 Roppongi, Minato-ku, Tokyo 106-0032"
    },
    {
      "id": "95",
      "name": "EDO CULTURAL CENTER",
      "address": "7-10-9 Akasaka, Minato-ku, Tokyo 107-0052"
    },
    {
      "id": "96",
      "name": "THE KAZANKAI FOUNDATION THE TOA LANGUAGE INSTITUTE",
      "address": "Akasaka Kazan Bldg 3F,. 2-17-47 Akasaka, Minato-ku, Tokyo 107-0052"
    },
    {
      "id": "97",
      "name": "AOYAMA INTERNATIONAL EDUCATION INSTITUTE JAPANESE LANGUAGE CENTER",
      "address": "1-1-1 Aoyama, Minato-ku, Tokyo 107-0052"
    }
  ]
}
```

Figura 42 - Respuesta JSON con lista de colegios de japonés

4.1.2.2.1.4.2 INFORMACIÓN DE ESCUELA DE JAPONÉS

Para obtener la información del colegio de lengua japonesa el proceso vuelve a ser transparente para el usuario. Dado un ID del colegio de japonés el sistema accede a su página y recoge toda la información relevante incluidas estadísticas.

Ejemplo:

URL: <http://www.infojapon.com:8081/escuelasIdiomas/id/117>

Respuesta:

```
{  
    "search": true,  
    "id": "5047",  
    "name": " KAI JAPANESE LANGUAGE SCHOOL",  
    "japaneseName": "カイ日本語スクール",  
    "address": "3F, Miyuki Bldg., 1-15-18 Okubo, Shinjuku-ku, Tokyo 169-0072",  
    "phone": "03-3205-1356 ",  
    "fax": "03-3207-4651 ",  
    "officialUrl": "http://www.kaij.co.jp",  
    "email": "admission@kaij.co.jp",  
    "howToArrive": "Directions from the nearest station 5 min. walk from JR Shin-Okubo station.10 min. walk from JR Okubo station.",  
    "establishmentType": "Joint stock corporation",  
    "representativeName": "Yamamoto Hiroko",  
    "principalName": "Yamamoto Hiroko",  
    "startingDate": "1987/2/1",  
    "validityTerm": "1990/4/1 to 2017/3/31",  
    "teachersNumber": "29 (Including Full-timeTeachers:6)",  
    "capacity": "280 (Double Shift)",  
    "accommodations": "Unavailable ",  
    "schoolStatus": "Unavailable",  
    "admissionRequirements": "Completion of 12 years of formal school education or equivalent.",  
    "selectionProcess": "Evaluation of documents, Interview and Interview with supporter",  
    "otherCourses": "Practical conversation course; Summer course; 50+ course; Private lesson",  
    "mapUrl": "https://www.google.com/maps/search/カイ日本語スクール",  
    "registeredStudents": {  
        "Myanmar": 0,  
        "Singapore": 1,  
        "Cambodia": 0,  
        "Malaysia": 3,  
        "Thailand": 1,  
        "Russia": 3,  
        "Vietnam": 0,  
        "Mongolia": 0,  
        "Sweden": 38,  
        "Srilanka": 0,  
        "SaudiArabia": 0,  
        "China": 3,  
        "UK": 5,  
        "France": 6,  
        "Nepal": 0,  
        "Korea": 2,  
        "Others": 25,  
        "USA": 13,  
        "Philippines": 0,  
        "Switzerland": 6,  
        "India": 1,  
        "Spain": 19,  
        "Canada": 1,  
    }  
}
```

Figura 43 - Respuesta JSON con información de colegio de japonés

4.1.2.2.2 DESARROLLO DEL PAQUETE RELACIONADO CON EL TRABAJO

El paquete com.puntojapon.work contiene el conjunto de clases y funciones necesario para la extracción y carga de información referente a ofertas de trabajo.

Cabe destacar la dificultad de encontrar una web que permitiera el crawling de ofertas de trabajo ya que todas pertenecen a compañías que denegaban rotundamente en sus políticas de privacidad y uso el uso de bots en sus páginas. Esto generó una dificultad añadida hasta que se dio con una página de ofertas de trabajo (www.applyq.com) que sí que permitía los bots y que incluso dotaban al desarrollador de una API para su obtención aunque no funcionara del todo bien.

Se intentó recopilar ofertas de trabajo de páginas web como www.glassdoor.com o www.linkedin.com sin éxito ya que estaban más enfocadas a otros aspectos y las propias APIs

no permitían a usuarios gratuitos la obtención de información referente a los puestos de trabajo.

4.1.2.2.2.1 BÚSQUEDA DE OFERTAS DE TRABAJO EN JAPÓN

En los parámetros de búsqueda de trabajo el usuario indica lo siguiente:

- Prefectura
- Alquiler mínimo (opcional)
- Alquiler máximo (opcional)
- Metros cuadrados (opcional)
- Habitaciones (opcional)
- Tipo de construcción (opcional)
- Distancia a estación (opcional)
- Antigüedad (opcional)
- Sin señal inicial (opcional)
- Admiten mascotas (opcional)
- Sin depósito (opcional)
- Periodo de estancia corto (opcional)
- Sin gastos de agencia (opcional)
- Furnished (opcional)
- Internet (opcional)
- Wifi (opcional)
- Pago con tarjeta (opcional)

De esta manera el crawler va a la página web origen de la información [27] con los parámetros de búsqueda propuestos y devuelve las viviendas de la página indicada.

Cada vez que se quiera acceder a una oferta el usuario será redirigido a la página origen de la oferta.

4.1.2.2.3 DESARROLLO DEL PAQUETE RELACIONADO CON LA VIVIENDA

En los parámetros de búsqueda de viviendas el usuario indica lo siguiente:

- Prefectura
- Especialidad (área de trabajo)
- Página (indicado por el sistema en el frontend)

Ejemplo:

URL:

http://www.infojapon.com:8081/vivir/Tokyo?page=1&min_price=20000&max_price=40000&min_meter=&rooms=&distance_station=&building_type=&building_age=&low_initial_costs=&no_guarantor=&no_key_money=&pets=&no_deposit=&short_term_ok=&no_agency_fee=&furnished=&internet=&wifi=&credit_card=

Respuesta:

```
{
  "searchType": "Houses search",
  "searchFound": 15,
  "page": 1,
  "hasNextPage": true,
  "searchState": true,
  "houseList": [
    {
      "buildingType": "Shared Guesthouse - Guest House",
      "place": "in Shiraitodai Fuchu-shi, Tokyo",
      "rent": "Rent ¥40,000",
      "size": "Size 13.00 m²",
      "deposit": "Deposit ¥0",
      "keyMoney": "¥0",
      "floor": "Floor 3 / 4F",
      "maintenanceFee": "Maintenance Fee ¥10,000 / mth",
      "nearestStation": "Musashinodai Station (5 min. walk)",
      "agent": "Irodori Factory",
      "imageUrl": "https://www.realestate.co.jp/image/2673/5238587-FFFF-280x180-c.jpg",
      "map": "東京都府中市白糸台5-38-1",
      "link": "https://www.realestate.co.jp/en/rent/view/330389"
    },
    {
      "buildingType": "Private Guesthouse - Guest House",
      "place": "in Honisshiki Edogawa-ku, Tokyo",
      "rent": "Rent ¥39,000",
      "size": "Size 10.00 m²",
      "deposit": "Deposit ¥0",
      "keyMoney": "¥0",
      "floor": "Floor 2 / 2F",
      "maintenanceFee": "Maintenance Fee ¥10,000 / mth",
      "nearestStation": "Shinkoiwa Station (13 min. walk)",
      "agent": "Yours Corporation Co.,Ltd.",
      "imageUrl": "https://www.realestate.co.jp/image/2629/5355335-FFFF-280x180-c.jpg",
      "map": "東京都江戸川区本一色1-21-16",
      "link": "https://www.realestate.co.jp/en/rent/view/254473"
    },
    {
      "buildingType": "Shared Guesthouse - Guest House",
      "place": "in Misuji Taito-ku, Tokyo",
      "rent": "Rent ¥32,000",
      "size": "Size 19.20 m²",
      "deposit": "Deposit ¥10,000",
      "keyMoney": "¥0",
      "floor": "Floor 6 / 6F",
      "maintenanceFee": "Maintenance Fee ¥8,000 / mth",
      "nearestStation": "Kuramae Station (4 min. walk)",
      "agent": "Ietomo Share House",
    }
  ]
}
```

Figura 44 - Respuesta JSON con lista de viviendas

4.1.2.3 DESARROLLO DEL PAQUETE DE FUNCIONES COMUNES EN EL PROYECTO

Existen un conjunto de clases de apoyo utilizables por todas las funciones del proyecto de forma recurrente:

4.1.2.3.1 APPLOGER

La clase applogger consta de un conjunto de funciones que permite el acceso concurrente al fichero app.log para escribir registros. Estos registros pueden ser de información, de atención, de error y de debug.

4.1.2.3.2 RANDOM USER AGENT

La clase random user agent permite generar de manera aleatoria un user-agent para enviar en las peticiones del bot. Esto se hizo así para evitar complicaciones antes de la defensa del proyecto pero el desarrollador se aseguró, en cualquier caso, de cumplir con las políticas de privacidad y de uso de cada página crawleada.

4.1.3 DESARROLLO DEL CONTROLADOR

Toda la programación relacionada con el controlador se basó en una única clase: MainController. Esta clase utiliza anotaciones del framework Spring para el levantamiento del servicio REST [28].

Las diferentes anotaciones utilizadas son las siguientes:

- `@RestController`: Indica la clase que va a ejercer como controlador REST y que va a contener los diferentes recursos

```
@RestController
public class MainController {
```

Figura 45 - Implementación etiqueta @RestController de Spring

- `@RequestMapping`: Indica que la función va a ser un recurso accesible por http. Este indicador debe ir acompañado de la ruta donde se alojará el recurso, el método que va a estar escuchando (GET, POST, PUT...) y el formato que producirá en su salida (JSON, XML, HTML..).
- `@ResponseStatus`: Indica el estado HTTP que devolverá la aplicación una vez accedido al recurso mediante petición.
- `@ResponseBody`: Indica el cuerpo de la función del recurso
- `@PathVariable`: Indica los parámetros que forman parte de la ruta del recurso
- `@RequestParam`: Indica los parámetros que no forman parte de la ruta del recurso pero si de la petición

```
@RequestMapping(value = "/posgrado/{prefecture}", method = RequestMethod.GET, produces = "application/json")
@ResponseStatus(HttpStatus.OK)
public @ResponseBody String searchGradSchool(HttpServletRequest response, HttpServletResponse request,
    @PathVariable("prefecture") String[] prefecture, @RequestParam(value = "nameGrad") String nameGrad,
    @RequestParam(value = "typeGrad") String[] typeGrad,
    @RequestParam(value = "typeCourse") String[] typeCourse,
    @RequestParam(value = "englishCourse") String[] englishCourse) throws Exception {
```

Figura 46 - Implementación de función como recurso de la API REST utilizando Spring

El controlador posee una clase embebida para retornar respuestas con estado HTTP 404 cuando un recurso llama a un crawler pero este no puede completar su función.

```
@ResponseStatus(value = HttpStatus.NOT_FOUND, reason = "Search results were empty") // 404
public class RequestNotFoundException extends RuntimeException {

    private static final long serialVersionUID = 1L;

}
```

Figura 47 - Implementación de función para respuesta de errores 404

El resto de funciones de MainController indica el conjunto de recursos de la API REST de InfoJapón.

4.1.3.1 RECURSOS API REST

4.1.3.1.1 BÚSQUEDA DE UNIVERSIDADES

Función: searchUni(response, request, prefecture, typeStudiesList, nameUni, typeUni, admisionMonth, deadLine, eju, engExam, admisionUni);

Ruta de recurso:

<http://www.infojapon.com:8081/universidades/{prefecture}/{typeStudies}?nameUni=&typeUni=&admisionMonth=&deadLine=&eju=&engExam=&admisionUni=>

Parámetros de ruta: prefecture, typeStudies

Parámetros de petición: nameUni, typeUni, admisionMonth, deadLine, eju, engExam, admisionUni

Salida: JSON con lista de universidades.

4.1.3.1.2 BÚSQUEDA DE COLEGIOS DE POSGRADO

Función: searchGradSchool(response, request, prefecture, nameGrad, typeGrad, typeCourse, englishCourse);

Ruta de recurso:

<http://www.infojapon.com:8081/posgrado/{prefecture}?nameGrad=&typeGrad=&typeCourse=&englishCourse=>

Parámetros de ruta: prefecture

Parámetros de petición: nameGrad, typeGrad, typeCourse, englishCourse

Salida: JSON con lista de colegios de posgrado.

4.1.3.1.3 BÚSQUEDA DE COLEGIOS DE FORMACIÓN PROFESIONAL

Función: searchTechSchool (response, request, prefecture, nameTech);

Ruta de recurso: <http://www.infojapon.com:8081/fp/{prefecture}?nameTech=>

Parámetros de ruta: prefecture

Parámetros de petición: nameTech

Salida: JSON con lista de colegios de formación profesional

4.1.3.1.4 MOSTRAR UNIVERSIDAD

Función: showUniversity (response, request, prefecture, id);

Ruta de recurso: <http://www.infojapon.com:8081/universidades/id/{id}>

Parámetros de ruta: id

Parámetros de petición:

Salida: JSON con información de la universidad

4.1.3.1.5 MOSTRAR COLEGIO DE POSGRADO

Función: showGradSchool (response, request, id);

Ruta de recurso: <http://www.infojapon.com:8081/posgrado/id/{id}>

Parámetros de ruta: id

Parámetros de petición:

Salida: JSON con información del colegio de posgrado

4.1.3.1.6 MOSTRAR COLEGIO DE FORMACIÓN PROFESIONAL

Función: showTechSchool (response, request, id);

Ruta de recurso: <http://www.infojapon.com:8081/fp/id/{id}>

Parámetros de ruta: id

Parámetros de petición:

Salida: JSON con información del colegio de posgrado

4.1.3.1.7 MOSTRAR INFORMACIÓN DE ADMISIONES DE DEPARTAMENTO UNIVERSITARIO

Función: showUniversityFacultyAdmissions (response, request, idUniversity, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/universidades/id/{idUniversity}/{idFaculty}/admissions>

Parámetros de ruta: idFaculty, idUniversity

Parámetros de petición:

Salida: JSON con información de admisiones del departamento universitario especificado

4.1.3.1.8 MOSTRAR INFORMACIÓN DE DEPARTAMENTO UNIVERSITARIO

Función: showUniversityFacultyInfo (response, request, idUniversity, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/universidades/id/{idUniversity}/{idFaculty}/info>

Parámetros de ruta: idFaculty, idUniversity

Parámetros de petición:

Salida: JSON con información del departamento universitario especificado

4.1.3.1.9 MOSTRAR INFORMACIÓN DE APOYO AL ESTUDIANTE DEL DEPARTAMENTO UNIVERSITARIO

Función: showUniversityFacultySupport (response, request, idUniversity, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/universidades/id/{idUniversity}/{idFaculty}/support>

Parámetros de ruta: idFaculty, idUniversity

Parámetros de petición:

Salida: JSON con información de apoyo al estudiante del departamento universitario especificado

4.1.3.1.10 MOSTRAR INFORMACIÓN DE INSTALACIONES DEL DEPARTAMENTO UNIVERSITARIO

Función: showUniversityFacultyFacilities (response, request, idUniversity, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/universidades/id/{idUniversity}/{idFaculty}/facilities>

Parámetros de ruta: idFaculty, idUniversity

Parámetros de petición:

Salida: JSON con información de las instalaciones del departamento universitario especificado

4.1.3.1.11 MOSTRAR INFORMACIÓN DE ACCESO DEL DEPARTAMENTO UNIVERSITARIO

Función: showUniversityFacultyAccess (response, request, idUniversity, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/universidades/id/{idUniversity}/{idFaculty}/access>

Parámetros de ruta: idFaculty, idUniversity

Parámetros de petición:

Salida: JSON con información de acceso del departamento universitario especificado

4.1.3.1.12 MOSTRAR INFORMACIÓN DE ADMISIONES DE DEPARTAMENTO DE COLEGIO DE POSGRADO

Función: showGradSchool FacultyAdmissions (response, request, idGraduate, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/posgrado/id/{idGraduate}/{idFaculty}/admissions>

Parámetros de ruta: idFaculty, idGraduate

Parámetros de petición:

Salida: JSON con información de admisiones del departamento del colegio de posgrado especificado

4.1.3.1.13 MOSTRAR INFORMACIÓN DE DEPARTAMENTO DE COLEGIO DE POSGRADO

Función: showGradSchool FacultyInfo (response, request, idGraduate, idFaculty);

Ruta de recurso: <http://www.infojapon.com:8081/posgrado/id/{idGraduate}/{idFaculty}/info>

Parámetros de ruta: idFaculty, idGraduate

Parámetros de petición:

Salida: JSON con información del departamento del colegio de posgrado especificado

4.1.3.1.14 MOSTRAR INFORMACIÓN DE APOYO AL ESTUDIANTE DEL DEPARTAMENTO DE COLEGIO DE POSGRADO

Función: showGradSchool FacultySupport (response, request, idGraduate, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/posgrado/id/{idGraduate}/{idFaculty}/support>

Parámetros de ruta: idFaculty, idGraduate

Parámetros de petición:

Salida: JSON con información de apoyo al estudiante del departamento del colegio de posgrado especificado

4.1.3.1.15 MOSTRAR INFORMACIÓN DE INSTALACIONES DEL DEPARTAMENTO DE COLEGIO DE POSGRADO

Función: showGradSchool FacultyFacilities (response, request, idGraduate, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/posgrado/id/{idGraduate}/{idFaculty}/facilities>

Parámetros de ruta: idFaculty, idGraduate

Parámetros de petición:

Salida: JSON con información de las instalaciones del departamento del colegio de posgrado especificado

4.1.3.1.16 MOSTRAR INFORMACIÓN DE ACCESO DEL DEPARTAMENTO DE COLEGIO DE POSGRADO

Función: showGradSchool FacultyAccess (response, request, idGraduate, idFaculty);

Ruta de recurso:

<http://www.infojapon.com:8081/posgrado/id/{idGraduate}/{idFaculty}/access>

Parámetros de ruta: idFaculty, idGraduate

Parámetros de petición:

Salida: JSON con información de acceso del departamento del colegio de posgrado especificado

4.1.3.1.17 MOSTRAR INFORMACIÓN DE ADMISIONES DE DEPARTAMENTO DE COLEGIO DE FORMACIÓN PROFESIONAL

Función: showTechSchoolFacultyAdmissions (response, request, idTechSchool, idFaculty);

Ruta de recurso: <http://www.infojapon.com:8081 /fp/id/{idTechSchool}/admissions>

Parámetros de ruta: idFaculty, idTechSchool

Parámetros de petición:

Salida: JSON con información de admisiones del departamento del colegio de formación profesional especificado

4.1.3.1.18 MOSTRAR INFORMACIÓN DE DEPARTAMENTO DE COLEGIO DE FORMACIÓN PROFESIONAL

Función: showTechSchoolFacultyInfo (response, request, idTechSchool, idFaculty);

Ruta de recurso: <http://www.infojapon.com:8081 /fp/id/{idTechSchool}/info>

Parámetros de ruta: idFaculty, idTechSchool

Parámetros de petición:

Salida: JSON con información del departamento del colegio de formación profesional especificado

4.1.3.1.19 MOSTRAR INFORMACIÓN DE APOYO AL ESTUDIANTE DEL DEPARTAMENTO DE COLEGIO DE FORMACIÓN PROFESIONAL

Función: showTechSchoolFacultySupport (response, request, idTechSchool, idFaculty);

Ruta de recurso: <http://www.infojapon.com:8081 /fp/id/{idTechSchool}/support>

Parámetros de ruta: idFaculty, idTechSchool

Parámetros de petición:

Salida: JSON con información de apoyo al estudiante del departamento del colegio de formación profesional especificado

4.1.3.1.20 BUSCAR ESCUELAS DE IDIOMAS

Función: showLanguageSchoolList (response, request, area);

Ruta de recurso: <http://www.infojapon.com:8081/escuelasIdiomas/{area}>

Parámetros de ruta: area

Parámetros de petición:

Salida: JSON con una lista de las escuelas idiomas de la prefectura especificada

4.1.3.1.21 MOSTRAR ESCUELA DE IDIOMA

Función: showLanguageSchoolInformation (response, request, idSchool);

Ruta de recurso: <http://www.infojapon.com:8081/escuelasIdiomas/id/{idSchool}>

Parámetros de ruta: idSchool

Parámetros de petición:

Salida: JSON con información de la escuela de idioma especificada

4.1.3.1.22 MOSTRAR ARTÍCULOS

Función: getLatestArticles (response, request);

Ruta de recurso: <http://www.infojapon.com:8081/articulos>

Parámetros de ruta:

Parámetros de petición:

Salida: JSON con los artículos ordenados cronológicamente

4.1.3.1.23 BÚSQUEDA SOBRE ARTÍCULOS

Función: searchArticleByTitle (response, request, titulo);

Ruta de recurso: <http://www.infojapon.com:8081/articulos/buscar?titulo=>

Parámetros de ruta:

Parámetros de petición: titulo

Salida: JSON con los artículos ordenados cronológicamente que encajen con la búsqueda

4.1.3.1.24 BÚSQUEDA DE OFERTAS DE TRABAJO

Función: getJobs (response, request, prefecture, specialty, page);

Ruta de recurso: <http://www.infojapon.com:8081/trabajo/{prefecture}/{specialty}>

Parámetros de ruta: prefecture, specialty

Parámetros de petición: page

Salida: JSON con una lista de ofertas de trabajo

4.1.3.1.25 BÚSQUEDA DE VIVIENDAS

Función: getHouses (response, request, prefecture, page, min_price, max_price, min_meter, rooms, distance_station, building_type, building_age, low_initial_costs, no_guarantor, no_key_money, pets, no_deposit, short_term_ok, no_agency_fee, furnished, internet, wifi, credit_card);

Ruta de recurso: <http://www.infojapon.com:8081>

/vivir/{prefecture}?page=&min_price=&max_price=&min_meter=&rooms=&distance_station=&building_type=&building_age=&low_initial_costs=&no_guarantor=&no_key_money=&pets=&no_deposit=&short_term_ok=&no_agency_fee=&furnished=&internet=&wifi=&credit_card=

Parámetros de ruta: prefecture

Parámetros de petición: page, min_price, max_price, min_meter, romos, distance_station, building_type, building_age, low_initial_costs, no_guarantor, no_key_money, pets, no_deposit, short_term_ok, no_agency_fee, furnished, internet, wifi, credit_card

Salida: JSON con una lista de viviendas

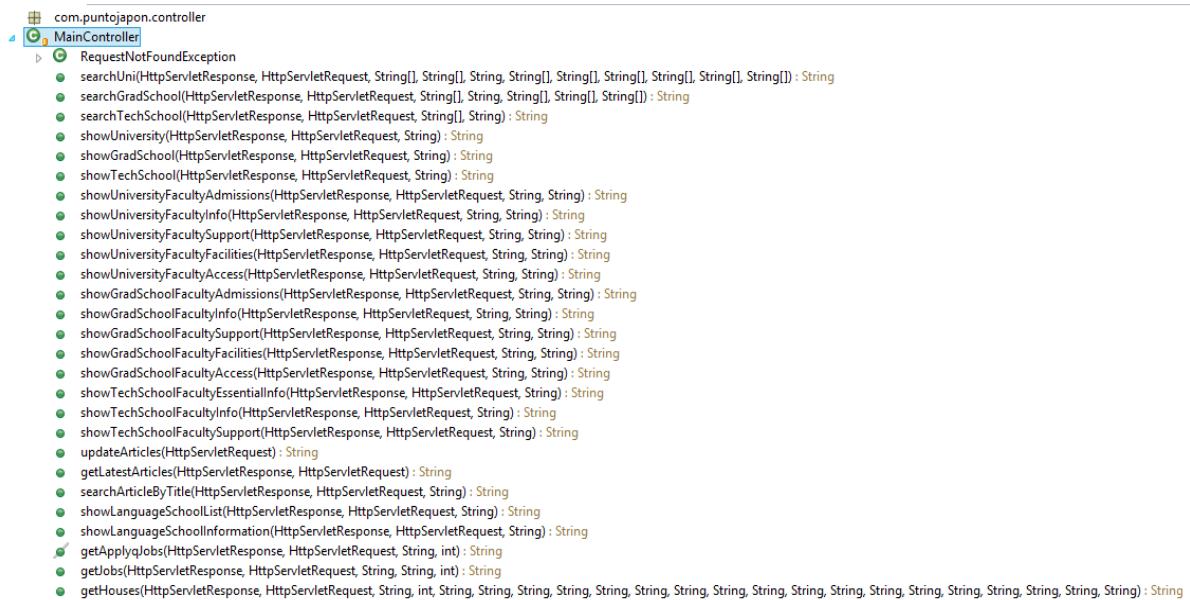


Figura 48 - Conjunto de funciones del controlador

4.2 DESARROLLO FRONTEND

El desarrollo frontend abarca todo el desarrollo de la aplicación cliente con la que el usuario interactuará con la aplicación backend. Para el desarrollo de la interfaz y la visualización de datos se han utilizado tecnologías como Javascript, HTML5, CSS y AJAX. Para el desarrollo de la aplicación cliente móvil se adaptó el código desarrollado en la aplicación cliente web para exportarlo a aplicación android (.apk) con el uso de la herramienta software Intel XDK.

Para que la aplicación alcanzara un nivel profesional en el diseño web se hizo uso de un framework CSS derivado de bootstrap que cuenta también con numerosos plugins integrados y cuyo nombre es “Canvas, the multipurpose html5 template” y se contó con la ayuda de una diseñadora.

Se estructuró la web en base a los requisitos definiendo como índice el módulo más recurrente por los usuarios que sería el de artículos y desde ahí se creó una estructura accesible para el resto de búsquedas (estudios, trabajo y vivienda).

4.2.1 MAPA APLICACIÓN CLIENTE

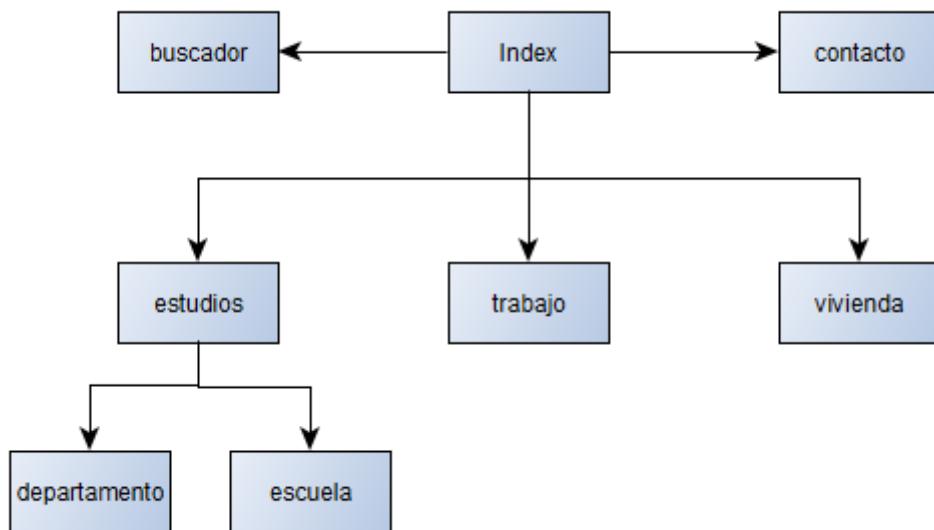
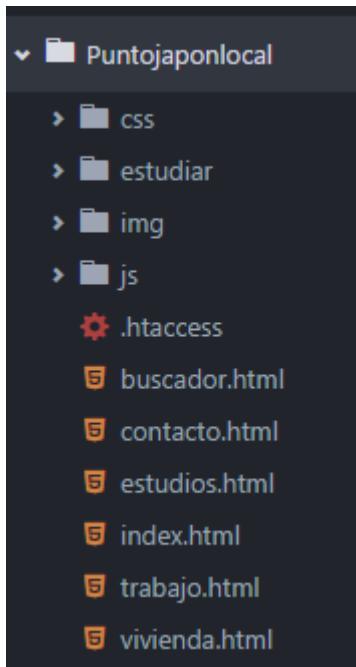


Figura 49 - Mapa aplicación cliente

4.2.2 ESTRUCTURA DE APLICACIÓN CLIENTE



- **css**: Contiene el código fuente referente a los estilos utilizados por la aplicación, tanto los del framework como los propios.
- **estudiar**: Contiene las páginas escuela y departamento
- **img**: Contiene el conjunto de imágenes utilizados por la aplicación cliente
- **js**: Contiene el conjunto de ficheros javascript utilizados para dar dinamismo a la aplicación cliente.
- **.htaccess**: Fichero de configuración referente a apache2 (ver más información en el capítulo 6) configurado para hacer accesibles los recursos web sin necesidad de especificar el formato (www.infojapon.com/estudios en lugar de www.infojapon.com/estudios.html)
- Páginas de la aplicación web en formato html.

Figura 50 - Estructura aplicación cliente

4.2.3 DESARROLLO APLICACIÓN

Una vez desarrollada la aplicación backend que escucharía peticiones HTTP en un determinado puerto, hubo que desarrollar una aplicación cliente que, mediante una interfaz sencilla e intuitiva, el usuario final pudiera realizar dichas peticiones y visualizar la información de la respuesta. Cabe destacar que para que las peticiones fueran válidas hubo que habilitar en cada función-recurso del backend la propiedad “Access-control-allow-origin” [29].

4.2.3.1 INDEX

4.2.3.1.1 INTERFAZ

El fichero index.html se desarrolló para ser la portada de la aplicación. El mismo debía mostrar los artículos de forma atractiva y en orden cronológico así como una serie de botones que fueran vistos a primera vista para acceder al resto de funciones principales de la aplicación, entendiendo por funciones principales las referentes a los objetivos: buscador de estudios, trabajo y vivienda. Por ello se determinó que dichos botones serían parte del primer plano junto al logo de la aplicación para que el usuario se percatará de ello desde el primer momento.



ESTUDIAR TRABAJAR VIVIR

Tanaka-kun wa itsumo kedaruge, la sencillez
de la pereza

14/07/16 11:45 / Peso del Aire
La temporada de verano ha llegado y con ella, la multitud de
nuevos animes que queremos ver y de los que os hablábamos en
esta entrada. La temporada de primavera apenas trajo novedades que
me interesaran y excepto un par de animes que ya os he comentado en
Primeras impresiones, el mayor adierto de la temporada [...] La entrada
Tanaka-kun wa itsumo kedaruge, la sencillez de la pereza aparece

Figura 51 - Página índice aplicación cliente

De esta manera si el usuario quería seguir leyendo simplemente debía hacer scroll por la página lo que provocaría que los archivos se fueran cargando poco a poco y, para que el menú estuviera siempre presente y el usuario pudiera tener todas las opciones en pantalla, se optó a, según el usuario hace scroll, dejar el menú minimizado en la parte superior de la pantalla accesible mediante un botón interactivo e intuitivo.

The screenshot shows the Info Japón mobile application interface. At the top left is the logo 'INFO JAPÓN'. On the right is a three-line menu icon. Below the logo, a news article by 'Razi' is displayed with a timestamp '14/07/16 05:04'. The article content discusses summer anime releases. To the right of the article is a vertical menu with options: ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. At the bottom of the screen are two more news articles: 'Fideos japoneses, guía para no liarse con ellos' and 'Muchos somos los adeptos a la gastronomía japonesa, y no es para menos. En Cool Japan recordamos hace poco que la UNESCO declaró al washoku (cultura gastronómica japonesa) patrimonio cultural intangible de la humanidad. Sin embargo, en España, durante muchos años ha habido una asociación casi automática de gastronomía japonesa con sushi. Y es curioso, [...] La entrada Fideos japoneses, guía para no liarse con ellos aparece primero en cooljapan.es.' Each article has its own timestamp and author information.

Figura 53 - Menú responsive aplicación cliente

Figura 52 - Menu responsive desplegado aplicación cliente

Por último, se apostó por incorporar un botón de búsqueda para acceder a las funcionalidades de búsquedas sobre artículos. Posicionado de forma deliberada y con un icono intuitivo, invita al usuario a buscar sobre sus intereses en la colección de artículos existentes.

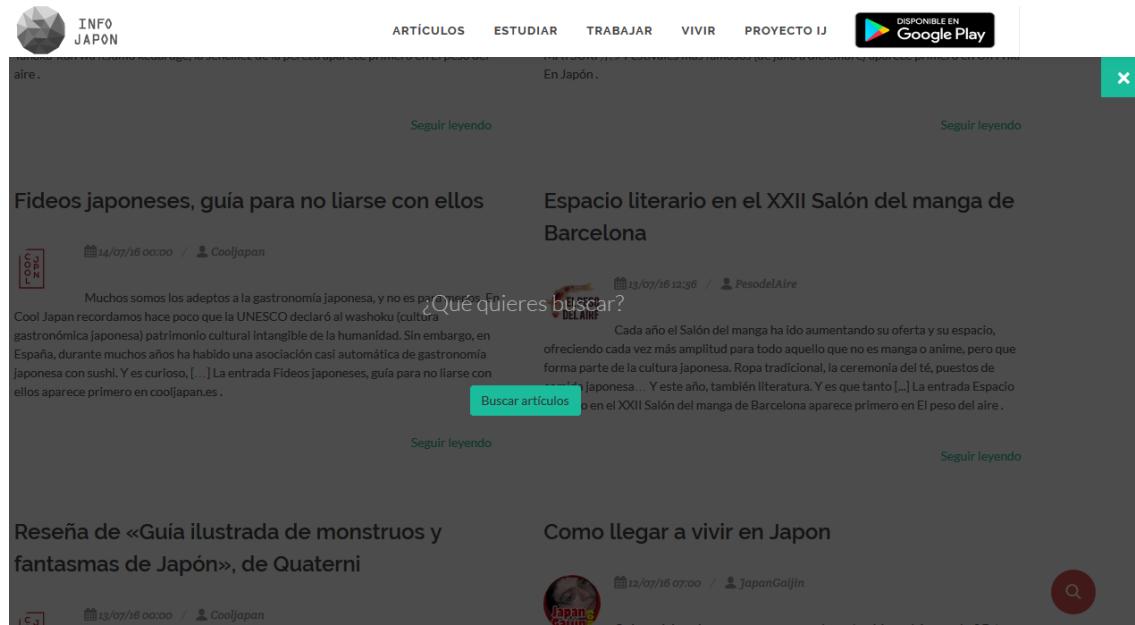


Figura 54 - Pantalla de búsqueda de artículos

De esta manera se accedería directamente a la visualización de artículos encontrados en la página buscador.html.

4.2.3.1.2 PETICIONES A LA API REST

Una vez el usuario accede al index se envía una petición al servidor REST para cargar los 10 primeros artículos disponibles en el JSON sort.json que contiene los artículos ordenados cronológicamente. El usuario simplemente necesita hacer scroll para cargar los siguientes 10 artículos y así sucesivamente.

Una de las complicaciones del desarrollo del index fue la problemática de cargar los artículos al hacer scroll. El sistema tiene en cuenta si el usuario ha llegado o no al final basándose en el tamaño completo de la ventana pero, al ser un diseño responsive que funciona tanto en ordenadores como en dispositivos portátiles (Tablets, Smartphones...), ocurría que en el ordenador podía funcionar y en los dispositivos con una menor pantalla no. Para solventar este error se creó un div limitador invisible que, al sobrepasarlo en cualquier dispositivo llamaría a la función de cargar artículos.

```
$(window).scroll(function() {  
  
    var articleOffset = $("#show-data").offset().top;  
    var articleOffsetBottom = $("#show-data").offset().top + $("#show-data").height();  
  
    if ($(this).scrollTop() > articleOffset) {  
        $('#header').fadeIn();  
    } else {  
        $('#header').fadeOut();  
    }  
  
    if (!loading && $(this).scrollTop() >= ($('#carga').offset().top + $('#carga').outerHeight() - window.innerHeight - 100)) {  
        loading = true;  
        loadArticles(articulosCargados + 10);  
        loading = false;  
    }  
});
```

Figura 55 - Código javascript para la carga de artículos al hacer scroll

PETICIÓN ASÍNCRONA (AJAX):

The screenshot shows a network request details view. At the top, it displays the URL <http://www.infojapon.com:8081/articulos>, status OK, and various metadata like infojapon.com:8081, 58,0 KB, and IP address 91.134.143.80:8081. Below this, there are tabs for 'Encabezados' (Headers), 'Respuesta' (Response), 'JSON', and 'Caché'. The 'Encabezados' tab is selected, showing the response headers:

Content-Length	59342
Content-Type	application/json; charset=UTF-8
Date	Thu, 14 Jul 2016 11:41:07 GMT
Server	Apache-Coyote/1.1
access-control-allow-orig...	*

Below the headers, the 'Solicitud' (Request) tab is visible, showing the client-side request headers:

Accept	application/json, text/javascript, */*; q=0.01
Accept-Encoding	gzip, deflate
Accept-Language	es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Connection	keep-alive
Host	www.infojapon.com:8081
Origin	http://www.infojapon.com
Referer	http://www.infojapon.com/index
User-Agent	Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0

Figura 56 - Petición AJAX artículos

VISUALIZACIÓN DEL JSON DEVUELTO:

Tanaka-kun wa itsumo kedaruge, la sencillez de la pereza



La temporada de verano ha llegado y con ella, la multitud de nuevos animes que queremos ver y de los que os hablábamos en esta entrada. La temporada de primavera apenas traía novedades que me interesarán y excepto un par de animes que ya os he comentado en Primeras impresiones, el mayor acierto de la temporada [...] La entrada Tanaka-kun wa itsumo kedaruge, la sencillez de la pereza aparece primero en El peso del aire.

[Seguir leyendo](#)

MATSURI 祭り Festivales más famosos (de julio a diciembre)



Muy buenas! Ya han pasado unos días desde mi entrada ded los festivales de enero a junio por lo que creo que es un buen momento para continuar con los de julio hasta diciembre... alomejor podeis aprovechar la información para disfrutar de alguno en julio y agosto que vienen unos cuantos ^^. Por mi parte yo... [LEER MÁS...] La entrada MATSURI 祭り Festivales más famosos (de julio a diciembre) aparece primero en Un Friki En Japón.

[Seguir leyendo](#)

Fideos japoneses, guía para no liarse con ellos



Muchos somos los adeptos a la gastronomía japonesa, y no es para menos. En Cool Japan recordamos hace poco que la UNESCO declaró al washoku (cultura gastronómica japonesa) patrimonio cultural intangible de la humanidad. Sin embargo, en España, durante muchos años ha habido una asociación casi automática de gastronomía japonesa con sushi. Y es curioso. [...] La entrada Fideos japoneses, guía para no liarse con ellos aparece primero en cooljapan.es.

[Seguir leyendo](#)

Espacio literario en el XXII Salón del manga de Barcelona



Cada año el Salón del manga ha ido aumentando su oferta y su espacio, ofreciendo cada vez más amplitud para todo aquello que no es manga o anime, pero que forma parte de la cultura japonesa. Ropa tradicional, la ceremonia del té, puestos de comida japonesa... Y este año, también literatura. Y es que tanto [...] La entrada Espacio literario en el XXII Salón del manga de Barcelona aparece primero en El peso del aire.

[Seguir leyendo](#)

Figura 57 - Pantalla de visualización de artículos cargados del JSON

4.2.3.2 BUSCADOR

4.2.3.2.1 INTERFAZ

La página buscador.html aplica la misma interfaz que la página index.html para no confundir al usuario ya que al mostrar los artículos resultados de la búsqueda, el usuario continúa viendo artículos. De esta manera desaparece el botón del buscador y aparece un botón para volver a la página principal.



MATSURI 祭り Los Grandes festivales japoneses



¡Muy buenas fríkicillos del matsuriavero! Seguimos con mis estudios de turismo y últimamente me aprendí el listado completo de los Matsuri (festival japonés) más famosos de todo el país!! Puede ser una muy buena información porque a todos nos hace ilusión visitar alguno de estos festivales tradicionales nipones ¿verdad? Lo único malo es que estos... [LEER MÁS...] La entrada MATSURI 祭り Los Grandes festivales japoneses aparece primero en Un Friki En Japón.

[Seguir leyendo](#)

MATSURI 祭り Festivales más famosos (de enero a junio)



Muy buenas! Tuve un fin de semana intenso en Yamanashi ayudando en los campos de uvas.. tengo el cuerpo medio destruido pero eso no me impedirá seguir mis estudios nipones ¡preparados para seguir con el tema de los Matsuri Japoneses? En la entrada anterior aprendimos sobre los 10 GRANDES Matsuri repartidos en Tohoku, Tokyo y... [LEER MÁS...] La entrada MATSURI 祭り Festivales más famosos (de enero a junio) aparece primero en Un Friki En Japón.

[Seguir leyendo](#)

Figura 58 - Pantalla de artículos buscados

4.2.3.2.2 PETICIONES A LA API REST

La página de buscador tiene la siguiente estructura:

www.infojapon.com/buscador?titulo=

Por tanto está esperando recibir por parámetro GET un título para enviarlo en la petición a la API REST. Si el título no se indica salta un error.



Figura 59 - Pantalla de error si no se encuentran los artículos buscados

Este título es enviado desde el index tras hacer click en el botón del buscador, introducir el título en el formulario que emerge y hacer click en “Buscar artículos”.

PETICIÓN ASÍNCRONA (AJAX):

The screenshot shows the Network tab of a browser developer tools interface. The request is for <http://www.infojapon.com:8081/articulos/buscar?titulo=festivales>. The status is 200 OK, with a response size of 11,7 KB and a timestamp of 91.134.143.80:8081. The 'Encabezados' tab is selected, showing the following headers:

Nombre	Valor
Content-Length	12030
Content-Type	application/json; charset=UTF-8
Date	Thu, 14 Jul 2016 11:52:13 GMT
Server	Apache-Coyote/1.1
access-control-allow-origin	*

The 'Solicitud' tab shows the following request headers:

Nombre	Valor
Accept	application/json, text/javascript, */*; q=0.01
Accept-Encoding	gzip, deflate
Accept-Language	es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Connection	keep-alive
Host	www.infojapon.com:8081
Origin	http://www.infojapon.com
Referer	http://www.infojapon.com/buscador?titulo=festivales
User-Agent	Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0

Figura 60 - Petición AJAX búsqueda de artículos

VISUALIZACIÓN DEL JSON DEVUELTO:

Recomendaciones de verano



01/07/16 10:51 / PesodelAire

Recomendaciones de Khardan 1- Libro: Los amigos de Kazumi Yumoto Un libro sobre la amistad y la sociedad japonesa posterior a la segunda Guerra Mundial. Una muestra clara de lo que significaba ser niño en aquella época en Japón, descubriendo valores como la amistad, el valor o el honor junto a otros aspectos más tangibles. Este [...] La entrada Recomendaciones de verano aparece primero en El peso del aire .

[Seguir leyendo](#)

[VOLVER A ARTÍCULOS](#)

Estrenos de anime: Verano 2016



22/06/16 09:47 / PesodelAire

Se acerca el verano y con él, una cantidad ingente de estrenos de anime que no hemos querido perdernos en El peso del aire. Sin embargo, y para que no os sintáis abrumados con listas interminables, hemos querido hacer una selección de los que, a nuestro parecer, son los más interesantes. Eso sí, tendréis al [...] La entrada Estrenos de anime: Verano 2016 aparece primero en El peso del aire .

[Seguir leyendo](#)

Figura 61 - Pantalla de visualización de artículos encontrados cargados del JSON

4.2.3.3 ESTUDIOS

4.2.3.3.1 INTERFAZ

La página estudios.html contiene un formulario de 4 pasos responsive que permite al usuario armar paso a paso la consulta final que se hará contra la API REST para mostrar un listado con las universidades/colegios deseados. Se mantiene el menú en la parte superior para que el usuario tenga constantemente todo el abanico de posibilidades de la aplicación a un click.

Paso 1:

Se le indica al usuario que seleccione el nivel de estudio y se visualiza información relevante con el nivel de estudio seleccionado

The screenshot shows a user interface for selecting study levels. At the top, there's a navigation bar with a logo and links for ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. Below this, four numbered circles (1, 2, 3, 4) indicate the steps. Step 1 is titled 'Nivel de estudios' and contains a dropdown menu labeled 'Grado universitario'. Step 2 is titled 'Prefectura'. Step 3 is titled 'Tipo de estudios'. Step 4 is titled 'Elige tu centro'. Below the steps, there are two columns: 'Niveles de estudio' and 'Grado universitario'. The 'Niveles de estudio' section has a note about selecting the appropriate study level and a dropdown menu. The 'Grado universitario' section has notes about university requirements and JLPT exams, along with a link to information on scholarships. A 'SIGUIENTE' button is located at the bottom right.

Figura 62 - Pantalla formulario paso 1 de estudios

Paso 2:

Se le indica al usuario que especifique la prefectura japonesa donde desea buscar las universidades / colegios. Una vez seleccionada la prefectura, se visualizan los datos de la Wikipedia referentes a esa prefectura mediante una petición AJAX. Si fuera necesario, se le da la opción al usuario de ampliar el mapa al hacer click sobre él.

INFO JAPÓN
ARTÍCULOS ESTUDIAR TRABAJAR VIVIR PROYECTO IJ

1 Nivel de estudios
2 Prefectura
3 Tipo de estudios
4 Elige tu centro

Prefectura

Selecciona la prefectura donde quieras cursar tus estudios.

13. Tokio

Tokio

Tokio (東京都, Tōkyō-to?, lit. 'capital del este'; pronunciación japonesa: [tō kyo̥-]) es la capital de facto de Japón, localizada en el centro-este de la isla de Honshū, concretamente en la región de Kanto. En conjunto forma una de las 47 prefecturas de Japón, aunque su denominación oficial es metrópolis o capital (都 -to). La ciudad es el centro de la política, economía, educación, comunicación y cultura popular del país. Cuenta también con la mayor concentración de sedes corporativas, instituciones financieras, universidades y colegios, museos, teatros, y establecimientos comerciales y de entretenimiento de todo Japón.

Con una población que supera los 13 millones de habitantes, se subdivide en 23 barrios (ku); 26 ciudades (shi); un distrito (gun) subdividido en tres pueblos (chō o -machi) y una villa (son o -mura); y cuatro subprefecturas (shichō) subdivididas en dos pueblos y siete villas, que representan a varias pequeñas islas al sur de Honshū que se extienden más allá de 1800 km de Shinjuku, capital de la metrópoli y sede de la gobernación. El centro de Tokio, con sus 23 barrios, ocupa un tercio de la metrópoli, con una población cercana a los 13,23 millones de habitantes; esta área es lo que se conoce internacionalmente como la ciudad de Tokio. En su área metropolitana viven más de 36 millones de habitantes, lo que la convierte en la mayor aglomeración urbana del mundo.

Δ near de la isla Tokio es la romanización más común del nombre en japonés al nombre de

Hokkaidō

- 1. Hokkaidō
- 2. Tōhoku
- 3. Aomori
- 4. Iwate
- 5. Miyagi
- 6. Yamagata
- 7. Fukushima

Kansai

- 24. Mie
- 25. Shiga
- 26. Kyoto
- 27. Osaka
- 28. Hyogo
- 29. Nara
- 30. Wakayama

Kantō

- 8. Ibaraki
- 9. Tochigi
- 10. Gunma
- 11. Saitama
- 12. Chiba
- 13. Tokyo
- 14. Kanagawa

Chūgoku

- 31. Tottori
- 32. Shimane
- 33. Okayama
- 34. Hiroshima
- 35. Yamaguchi

Chūbu

- 15. Niigata
- 16. Toyama
- 17. Gifu
- 18. Fukui
- 19. Yamanashi
- 20. Shizuoka
- 21. Gifu
- 22. Shizuoka
- 23. Aichi

Shikoku

- 36. Tokushima
- 37. Kagawa
- 38. Ehime
- 39. Kochi

Kyūshū & Okinawa

- 40. Fukuoka
- 41. Saga
- 42. Nagasaki
- 43. Ōita
- 44. Miyazaki
- 45. Kagoshima
- 46. Okinawa

Figura 63 - Pantalla formulario paso 2 de estudios

Paso 3:

Se le indica al usuario que rellene, dependiendo de los estudios elegidos (grado universitario, master/doctarado, fp, lengua japonesa) un conjunto de parámetros de búsqueda variables.

- Estudios universitarios:

The screenshot shows the third step of a search process for university studies. At the top, there's a navigation bar with 'INFO JAPÓN' and links for 'ARTÍCULOS', 'ESTUDIAR', 'TRABAJAR', 'VIVIR', and 'PROYECTO IJ'. Below the navigation, four numbered circles (1, 2, 3, 4) indicate the sequence: 1. Nivel de estudios, 2. Prefectura, 3. Tipo de estudios (highlighted in green), and 4. Elige tu centro. The main form area has two sections: 'Tipos de estudios universitarios' (with a dropdown menu showing 'Ciencias sociales') and 'Opciones avanzadas (opcionales)' (with dropdown menus for 'NOMBRE DE UNIVERSIDAD' (empty), 'TIPO DE UNIVERSIDAD' (Pública), 'MES DE ADMISIÓN' (Indiferente), 'FECHA LÍMITE (MES)' (Indiferente), 'ADMISIÓN DE APLICACIONES' (Aplicación desde Japón), 'CERTIFICACIÓN DE INGLÉS NECESARIA' (Necesario TOEFL u otros), 'EXAMEN DE ENTRADA (EJU)' (Indiferente), and 'IDIOMA DE LAS CLASES' (Lecciones en inglés)). At the bottom are 'ANTERIOR' and 'INICIAR BÚSQUEDA' buttons.

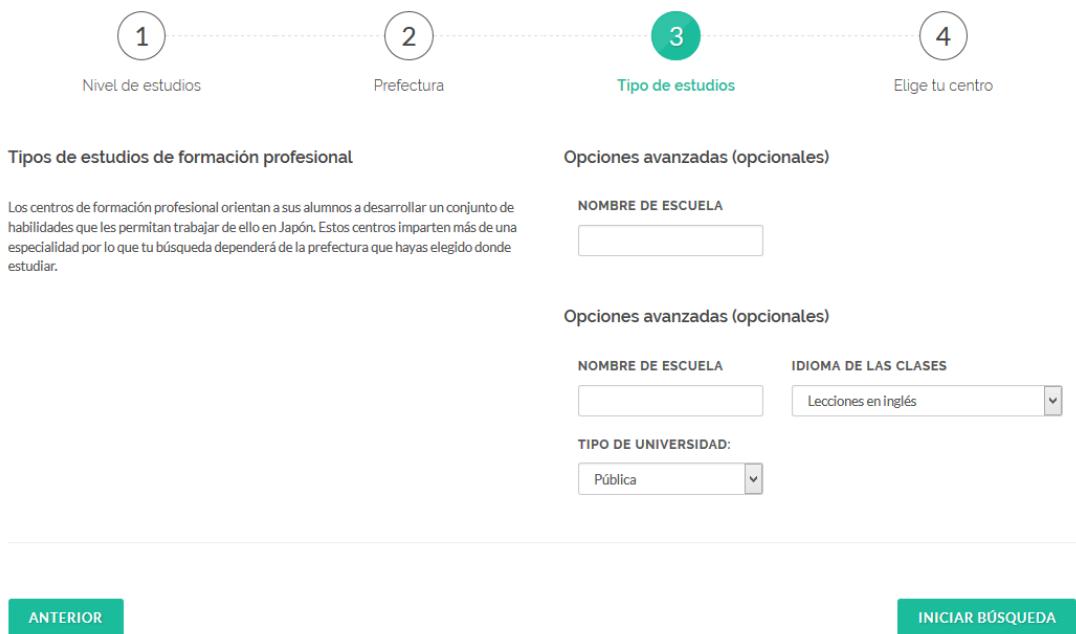
Figura 64 - Pantalla paso 3 estudios universitarios

- Estudios de posgrado

The screenshot shows the third step of a search process for postgraduate studies. The layout is similar to Figure 64, with a navigation bar, numbered steps (1-4), and search sections for 'Tipos de estudios de posgrado (Master y doctorado)' (dropdown menu showing 'Master') and 'Opciones avanzadas (opcionales)' (dropdown menus for 'NOMBRE DE ESCUELA' (empty), 'IDIOMA DE LAS CLASES' (Lecciones en inglés), 'TIPO DE UNIVERSIDAD' (Pública), and 'IDIOMA DE LAS CLASES' (Lecciones en inglés)). At the bottom are 'ANTERIOR' and 'INICIAR BÚSQUEDA' buttons.

Figura 65 - Pantalla paso 3 estudios de posgrado

- Estudios de formación profesional



1 Nivel de estudios 2 Prefectura 3 **Tipo de estudios** 4 Elige tu centro

Tipos de estudios de formación profesional

Los centros de formación profesional orientan a sus alumnos a desarrollar un conjunto de habilidades que les permitan trabajar de ello en Japón. Estos centros imparten más de una especialidad por lo que tu búsqueda dependerá de la prefectura que hayas elegido donde estudiar.

Opciones avanzadas (opcionales)

NOMBRE DE ESCUELA

Opciones avanzadas (opcionales)

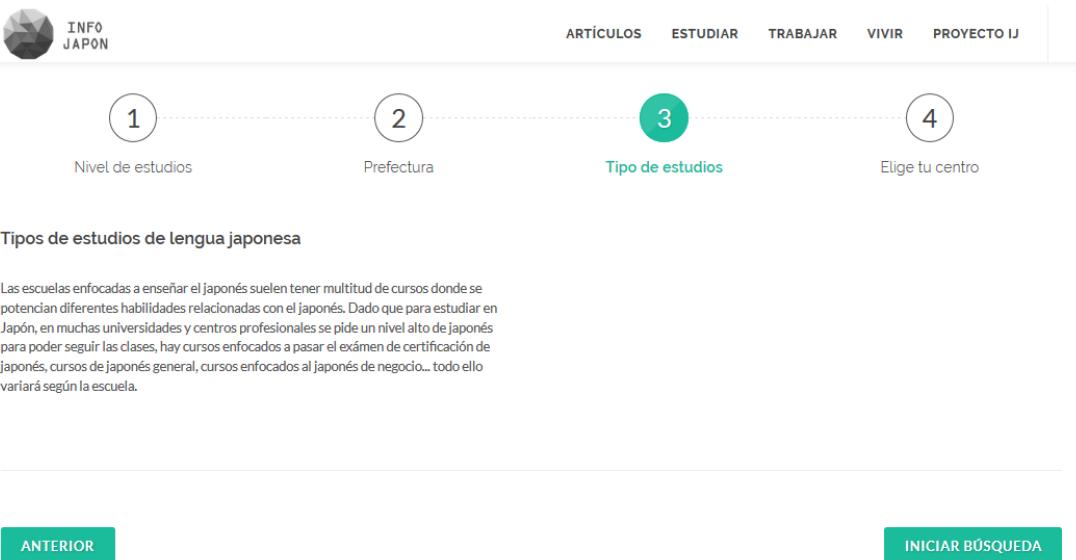
NOMBRE DE ESCUELA IDIOMA DE LAS CLASES Lecciones en inglés

TIPO DE UNIVERSIDAD: Pública

ANTERIOR **INICIAR BÚSQUEDA**

Figura 66 - Pantalla paso 3 estudios de formación profesional

- Estudios de japonés



1 Nivel de estudios 2 Prefectura 3 **Tipo de estudios** 4 Elige tu centro

Tipos de estudios de lengua japonesa

Las escuelas enfocadas a enseñar el japonés suelen tener multitud de cursos donde se potencian diferentes habilidades relacionadas con el japonés. Dado que para estudiar en Japón, en muchas universidades y centros profesionales se pide un nivel alto de japonés para poder seguir las clases, hay cursos enfocados a pasar el examen de certificación de japonés, cursos de japonés general, cursos enfocados al japonés de negocio... todo ello variará según la escuela.

ANTERIOR **INICIAR BÚSQUEDA**

Figura 67 - Pantalla paso 3 estudios de lengua japonesa

Paso 4:

El paso 4 contiene los resultados de la búsqueda que tenía como parámetros los datos provistos en el formulario entre los pasos 1 – 3. Al hacer click sobre el paso 4 se ejecuta la petición y se visualizan los datos que darán acceso a cada escuela / departamento:

The screenshot shows a search interface with four numbered steps: 1. Nivel de estudios, 2. Prefectura, 3. Tipo de estudios, and 4. Elige tu centro. Step 4 is highlighted with a green circle. Below the steps, two university profiles are listed: Meiji Gakuin University and Toyo Eiwa University.

Meiji Gakuin University - 明治学院大学

Tokyo, Kanagawa / Private
150 years of experience in educating for the future

Meiji Gakuin University traces its origins to the Hepburn Academy—an English school founded by the American missionary James C. Hepburn in 1863. The University will celebrate its 150th anniversary ... [[Meiji Gakuin University](#) Show more]

FACULTY OF SOCIOLOGY AND SOCIAL WORK

Toyo Eiwa University - 東洋英和女学院大学

Kanagawa, Tokyo / Private
英和スプリット

Toyo Eiwa University is located on the Yokohama Campus, about one hour by train from central Tokyo. There are approximately 2,300 students enrolled. The campus is endowed with rich natural resources... [[Toyo Eiwa University](#) Show more]

Figura 68 - Pantalla paso 4 con lista de elementos devueltos

El conjunto de transiciones del formulario así como la carga selectiva de información y las peticiones selectivas dependiendo de las partes del formulario llenado son todas abarcadas mediante javascript.

4.2.3.3.2 PETICIONES A LA API REST

La página de estudios.html contiene diversas peticiones HTTP:

4.2.3.3.2.1 PETICIÓN WIKIPEDIA

Cada vez que se selecciona una prefectura se realiza una petición GET a la API de Wikipedia para obtener la información de la prefectura seleccionada.

URL	Estado	Dominio	Tamaño	IP Remota
GET api.php?format=json&acti...e_Iw	200 OK	es.wikipedia.org	326 B	91.198.174.192:443
Parámetros	Encabezados	Respuesta	Caché	

Figura 69 - Petición Wikipedia para la carga de información de prefectura

4.2.3.3.2.2 PETICIÓN BÚSQUEDA DE ESTUDIOS UNIVERSITARIOS

Si se ha seleccionado y llenado el formulario de estudios universitarios se realizará una petición a la API REST para obtener la lista de universidades.

Recurso ejemplo:

<http://www.infojapon.com:8081/universidades/Kanagawa/Sociology?nameUni=&typeUni=&admissionMonth=&deadLine=&eju=&engExam=&admisionUni=>



The screenshot shows an AJAX request for "GET Sociology?nameUni=&typeUni=&engExam=&admisionUni=". The response status is 200 OK from infojapon.com:8081. The "Encabezados" (Headers) section shows the following details:

Nombre	Valor
Content-Length	13452
Content-Type	application/json; charset=UTF-8
Date	Thu, 14 Jul 2016 12:13:56 GMT
Server	Apache-Coyote/1.1
access-control-allow-origin	*

The "Encabezados de solicitud" (Request Headers) section shows the following details:

Nombre	Valor
Accept	application/json, text/javascript, */*; q=0.01
Accept-Encoding	gzip, deflate
Accept-Language	es-ES, es;q=0.8, en-US;q=0.5, en;q=0.3
Connection	keep-alive
Host	www.infojapon.com:8081
Origin	http://www.infojapon.com
Referer	http://www.infojapon.com/estudios
User-Agent	Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0

Figura 70 - Petición AJAX búsqueda de universidades

4.2.3.3.2.3 PETICIÓN BÚSQUEDA DE ESTUDIOS DE POSGRADO

Si se ha seleccionado y llenado el formulario de estudios de posgrado se realizará una petición a la API REST para obtener la lista de los colegios de posgrado.

Recurso ejemplo:

<http://www.infojapon.com:8081/posgrado/Tokyo?nameGrad=&typeGrad=master&typeCourse=public&englishCourse=englishYes>



The screenshot shows an AJAX request for "GET Tokyo?nameGrad=&typeGrad=&englishCourse=englishYes". The response status is 200 OK from infojapon.com:8081. The "Encabezados" (Headers) section shows the following details:

Nombre	Valor
Content-Length	2980
Content-Type	application/json; charset=UTF-8
Date	Thu, 14 Jul 2016 12:16:32 GMT
Server	Apache-Coyote/1.1
access-control-allow-origin	*

The "Encabezados de solicitud" (Request Headers) section shows the following details:

Nombre	Valor
Accept	application/json, text/javascript, */*; q=0.01
Accept-Encoding	gzip, deflate
Accept-Language	es-ES, es;q=0.8, en-US;q=0.5, en;q=0.3
Connection	keep-alive
Host	www.infojapon.com:8081
Origin	http://www.infojapon.com
Referer	http://www.infojapon.com/estudios
User-Agent	Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0

Figura 71 - Petición AJAX búsqueda de colegios de posgrado

4.2.3.3.2.4 PETICIÓN BÚSQUEDA DE ESTUDIOS DE FORMACIÓN PROFESIONAL

Si se ha seleccionado y llenado el formulario de estudios de formación profesional se realizará una petición a la API REST para obtener la lista de los colegios de formación profesional.

Recurso ejemplo: <http://www.infojapon.com:8081/fp/Tokyo?nameTech=>



GET Tokyo?nameTech= 200 OK infojapon.com:8081

Parámetros Encabezados Respuesta JSON Caché

Encabezados de respuesta ver fuente

Content-Length	12586
Content-Type	application/json; charset=UTF-8
Date	Thu, 14 Jul 2016 12:17:48 GMT
Server	Apache-Coyote/1.1
access-control-allow-origin	*

Encabezados de solicitud ver fuente

Accept	application/json, text/javascript, */*; q=0.01
Accept-Encoding	gzip, deflate
Accept-Language	es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Connection	keep-alive
Host	www.infojapon.com:8081
Origin	http://www.infojapon.com
Referer	http://www.infojapon.com/estudios
User-Agent	Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0

Figura 72 - Petición AJAX búsqueda de colegios de formación profesional

4.2.3.3.2.5 PETICIÓN BÚSQUEDA DE ESTUDIOS DE JAPONÉS

Si se ha seleccionado y llenado el formulario de estudios de japonés se realizará una petición a la API REST para obtener la lista de los colegios de japonés

Recurso ejemplo: <http://www.infojapon.com:8081/escuelasIdiomas/tokyo>



GET tokyo 200 OK infojapon.com:8081

Parámetros Encabezados Respuesta JSON Caché

Encabezados de respuesta ver fuente

Content-Length	18859
Content-Type	application/json; charset=UTF-8
Date	Thu, 14 Jul 2016 12:18:57 GMT
Server	Apache-Coyote/1.1
access-control-allow-origin	*

Encabezados de solicitud ver fuente

Accept	application/json, text/javascript, */*; q=0.01
Accept-Encoding	gzip, deflate
Accept-Language	es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Connection	keep-alive
Host	www.infojapon.com:8081
Origin	http://www.infojapon.com
Referer	http://www.infojapon.com/estudios
User-Agent	Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0

Figura 73 - Petición AJAX búsqueda de colegios de japonés

Todas las listas devueltas tienen paginación propia intuitiva:

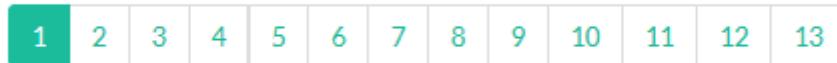


Figura 74 - Sistema de paginación de resultados

4.2.3.4 ESCUELA

La página escuela.html recibe por parámetros el tipo de escuela y el id de la escuela. Estos parámetros determinarán la manera de visualizar la escuela que se ha pasado por la id y la petición a realizar a la API REST como veremos a continuación.

4.2.3.4.1 MOSTRAR UNIVERSIDAD

Ejemplo: <http://www.infojapon.com/estudiar/escuela?tipo=univ&id=359>

Petición asíncrona: GET <http://www.infojapon.com:8081/universidades/id/359>

Visualización de datos:

The screenshot shows the homepage of Meiji Gakuin University. At the top, there's a navigation bar with links for ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. On the left, there's a logo for 'INFO JAPÓN'. The main content features a large image of the university's campus in Tokyo, Kanagawa. Text on the right reads: 'Tokyo, Kanagawa / Private' and '150 years of experience in educating for the future'. Below this, a detailed paragraph describes the university's history, mentioning its founding by James C. Hepburn in 1863, its 150th anniversary in 2013, and its focus on English language education and international exchange. It also notes the presence of campuses in Tokyo (Shirokane) and Yokohama (Totsuka). At the bottom, there are buttons for various faculties: FACULTY OF LETTERS, FACULTY OF ECONOMICS, FACULTY OF SOCIOLOGY AND SOCIAL WORK, FACULTY OF LAW, FACULTY OF INTERNATIONAL STUDIES, and FACULTY OF PSYCHOLOGY.

Fuente: <http://www.nishinkyo.org/>

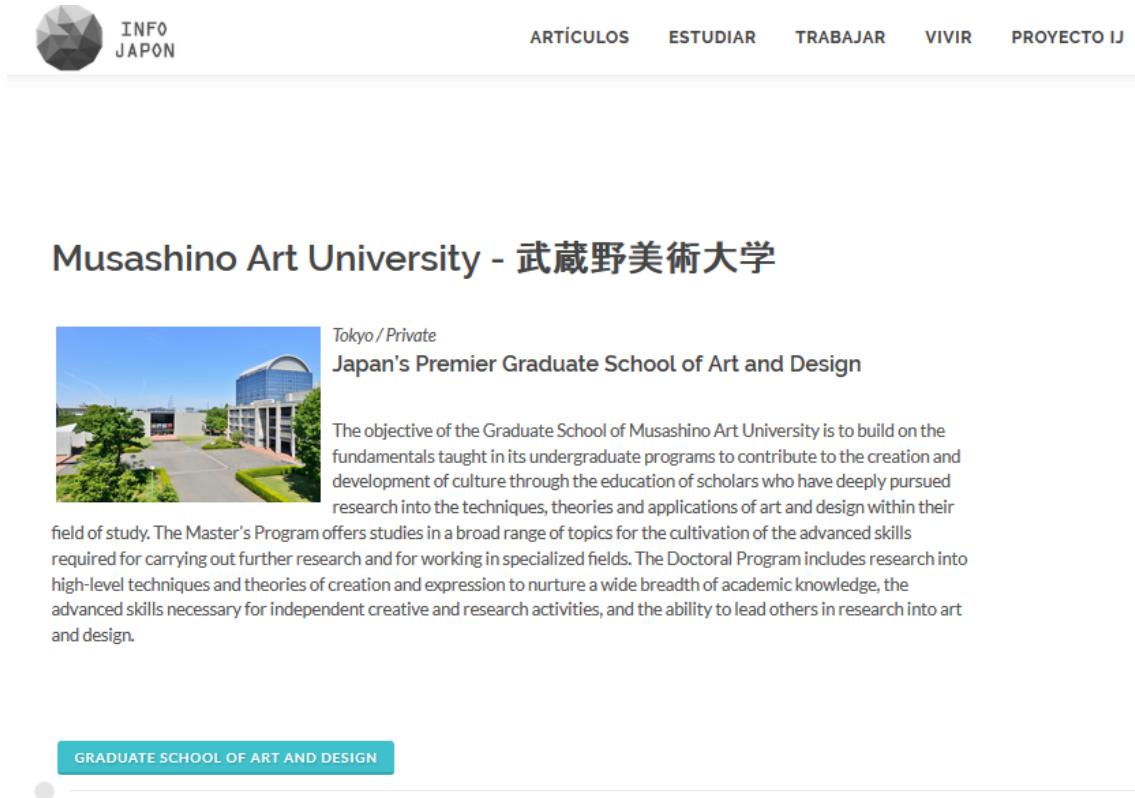
Figura 75 - Pantalla de universidad

4.2.3.4.2 MOSTRAR COLEGIO DE POSGRADO

Ejemplo: <http://www.infojapon.com/estudiar/escuela?tipo=grad&id=296>

Petición asíncrona: GET <http://www.infojapon.com/estudiar/escuela?tipo=grad&id=296>

Visualización de datos:



The screenshot shows the header of the Info Japón website with a dark grey navigation bar. On the left is a circular icon with a geometric pattern and the text 'INFO JAPÓN'. To the right are five menu items: ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. Below the header, the main content area features the title 'Musashino Art University - 武藏野美術大学' in large, bold, dark blue font. To the left of the title is a small thumbnail image of a modern building complex under a clear blue sky. To the right of the title is a subtitle 'Tokyo / Private' and 'Japan's Premier Graduate School of Art and Design'. A detailed description follows: 'The objective of the Graduate School of Musashino Art University is to build on the fundamentals taught in its undergraduate programs to contribute to the creation and development of culture through the education of scholars who have deeply pursued research into the techniques, theories and applications of art and design within their field of study. The Master's Program offers studies in a broad range of topics for the cultivation of the advanced skills required for carrying out further research and for working in specialized fields. The Doctoral Program includes research into high-level techniques and theories of creation and expression to nurture a wide breadth of academic knowledge, the advanced skills necessary for independent creative and research activities, and the ability to lead others in research into art and design.' At the bottom of the content area is a teal-colored button labeled 'GRADUATE SCHOOL OF ART AND DESIGN'.

Fuente: <http://www.nisshinkyo.org/>

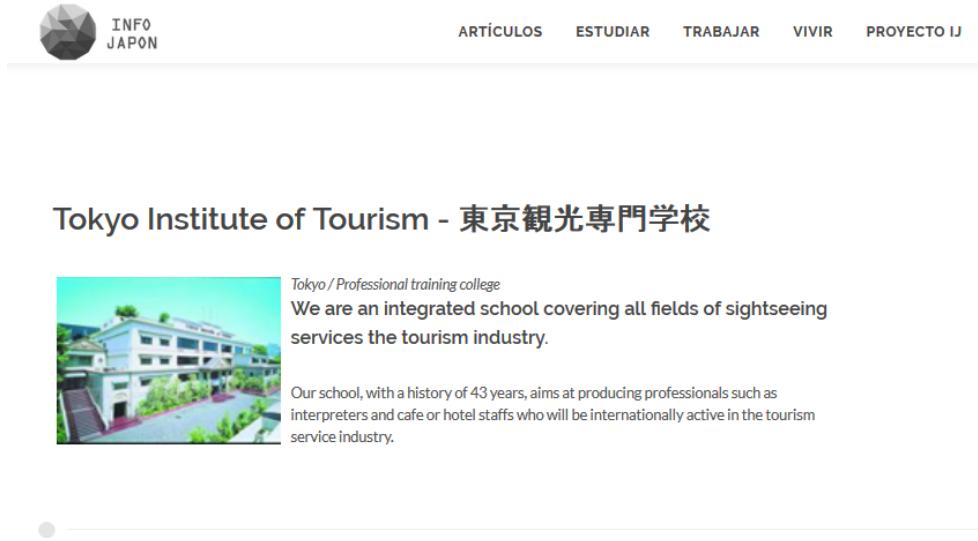
Figura 76 - Pantalla de colegio de posgrado

4.2.3.4.3 MOSTRAR COLEGIO DE FORMACIÓN PROFESIONAL

Ejemplo: <http://www.infojapon.com/estudiar/escuela?tipo=tech&id=6>

Petición asíncrona: GET <http://www.infojapon.com:8081/fp/id/6>

Visualización de datos:



The screenshot shows a web page for the Tokyo Institute of Tourism. At the top left is the Info Japón logo. To the right are navigation links: ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. The main title is "Tokyo Institute of Tourism - 東京観光専門学校". Below the title is a photograph of a modern, multi-story building. To the right of the photo, the text reads: "Tokyo / Professional training college. We are an integrated school covering all fields of sightseeing services the tourism industry." Below this, another text block states: "Our school, with a history of 43 years, aims at producing professionals such as interpreters and cafe or hotel staffs who will be internationally active in the tourism service industry."

Fuente: <http://www.nisshinkyo.org/>

Figura 77 - Pantalla de colegio de formación profesional

4.2.3.4.4 MOSTRAR COLEGIO DE LENGUA JAPONESA

Ejemplo: <http://www.infojapon.com/estudiar/escuela?tipo=japones&id=117>

Petición asíncrona: GET <http://www.infojapon.com:8081/escuelasIdiomas/id/117>

Visualización de datos:

KAI JAPANESE LANGUAGE SCHOOL

カイ日本語スクール

Información de la escuela

Dirección: 3F, Miyuki Bldg, 1-15-18 Okubo, Shinjuku-ku, Tokyo 169-0072
Teléfono: 03-3207-1356
Fax: 03-3207-4651
Web oficial: <http://www.kaij.jp>
Email: admission@kaij.co.jp

Tipo de establecimiento: Joint stock corporation
Nombre del representante: Yamamoto Hiroko
Nombre del director: Yamamoto Hiroko
Fundación: 1987/2/1
Período de licencia: 1990/4/1 to 2017/3/31
Número de profesores: 29 (Including Full-timeTeachers: 6)
Capacidad: 280 (Double Shift)
Alojamiento: Unavailable
Estado bajo la ley de educación escolar japonesa: Unavailable

Requisitos de admisión:
Completion of 12 years of formal school education or equivalent.
Proceso de selección:
Evaluation of documents, Interview and Interview with supporter
Otros cursos:
Practical conversation course; Summer course; 50+ course; Private lesson

Cómo llegar:
Directions from the nearest station 5 min. walk from JR Shin-Okubo station. 10 min. walk from JR Okubo station.

Figura 78 - Pantalla 1 de colegio de japonés

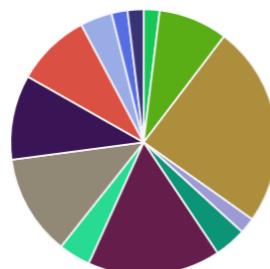
Curso	Propósito	Duración	Horas de clase	Semanas	Meses de admisión	Tasa de selección	Tasa de admisión	Tasa de clases	Otros pagos	Pago total
General Japanese Course Morning Class	General	2years	1,600 h	80	4,7,10,1	20,000 ¥	20,000 ¥	1,520,000 ¥	20,000 ¥	1,580,000 ¥
General Japanese Course Afternoon Class	General	2years	1,600 h	80	4,7,10,1	20,000 ¥	20,000 ¥	1,520,000 ¥	20,000 ¥	1,580,000 ¥

Características

- We train students' skill of the communication in Japanese and raise talented person functioning in Japanese environment.
- We build multi-national learning environment and offer education to raise autonomy.
- To improve the students' learning effect, we are providing digital audio materials for self-study.

Estadísticas

Nacionalidades de estudiantes



Destino de graduados

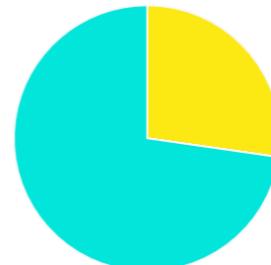


Figura 79 - Pantalla 2 de colegio de japonés

4.2.3.5 DEPARTAMENTO

La página de departamento.html está orientada sobre todo a los departamentos pertenecientes a las universidades, colegios de posgrado y colegios de formación profesional.

Una vez en listado el conjunto de universidades/colegios o incluso en la misma página de la propia universidad/colegio, se listan los diferentes departamentos y son accesibles mediante los recuadros azules.



Figura 80 - Botón de acceso a departamentos

Cabe destacar que la página departamento.html recibe como parámetros el tipo de escuela/universidad, la id de la misma y la id del departamento

4.2.3.5.1 INTERFAZ

Se ha continuado con el mismo estilo repartiendo los elementos del departamento en la parte inferior y los elementos ya vistos de la universidad/colegio en la parte superior. Se mantiene el menú superior por las razones expuestas anteriormente.

INFO JAPON

ARTÍCULOS ESTUDIAR TRABAJAR VIVIR PROYECTO IJ

J. F. Oberlin University - 桜美林大学

Tokyo, Kanagawa / Private / Guía universitaria

Fostering true global citizens through "Liberal Arts Education" and "Comprehensive Student Life Support"

What kind of school is J. F. Oberlin University? J. F. Oberlin University is a school located in Tokyo that places an emphasis on liberal arts education. The university's comprehensive programs are divided into two main programs: "Liberal Arts," through which learning in a variety of fields allows students to create their own futures, and "Professional Arts," which provides practical learning for students' future endeavors and careers.

What is the curriculum like? The curriculum at J. F. Oberlin University is notable for using a major/minor system. As a requirement for graduation, students must complete the specialized programs and courses within their affiliated field as the major component, but students may also select a minor in another field if they wish to do so. Some students also design their own double major program and graduate with two degrees in four years of study.

Why should I choose J. F. Oberlin University for study in Japan? There are approximately 500 international students from 20 countries studying on campus. International students choose J. F. Oberlin University not only for its educational programs but also for the comprehensive student support systems that include the university's scholarship programs, two international dormitories, and an employment rate of 100% for international student graduates.

Why is J. F. Oberlin University so popular with students from China? The predecessor to J. F. Oberlin University was the Chongzhen Vocational and Grammar School for Girls, which was founded in Beijing in 1921 to provide learning opportunities for poor children. This history has made J. F. Oberlin University a popular choice for Chinese students to this day. There is a local office in Beijing, and entrance examinations are held (for April or September enrollment) in other major cities in China, such as Shanghai, Tianjin, and Dalian, in cooperation with Chinese universities. The university's Admissions office has Chinese-speaking staff available at all times, so please contact us with any questions you may have.

ARTS AND SCIENCES BUSINESS MANAGEMENT GLOBAL COMMUNICATION PERFORMING AND VISUAL ARTS

HEALTH AND WELFARE

Figura 81 - Pantalla departamentos 1

Arts and Sciences

Noticias

Acceso Admisiones Instalaciones Ayuda al estudiante Información

Acceso a facultad

Machida Campus

Dirección: 3758 Tokiwa-machi, Machida-shi, Tokyo

Sitios cercanos:

- About one hour, 15 minutes from Shinjuku Station and Shibuya Station
- About 30 minutes from Hachioji Station More details can be found on our web site: <http://www.obirin.ac.jp/en/access/machida.html>
- A free shuttle bus (approx. 8 minutes) is available from Fuchinobe Station on the JR Yokohama Line.
- A free shuttle bus (approx. 20 minutes) is available from Tama Center Station, which can be reached using the Keio, Odakyu, or Tama Monorail lines. (Access from major stations) • About one hour, 15 minutes from Shinjuku Station and Shibuya Station • About 30 minutes from Hachioji Station More details can be found on our web site: <http://www.obirin.ac.jp/en/access/machida.html>



Figura 82 - Pantalla departamentos 2

4.2.3.5.2 PETICIONES A LA API REST

Pese a que el contenido se muestra de forma uniforme, se realizan un total de 6 llamadas a la API REST.

URL ejemplo:

http://www.infojapon.com/estudiar/departamento?tipo=univ&id=271&id_depart=687

Peticiones asíncronas (AJAX):

- <http://www.infojapon.com:8081/universidades/id/271>
- <http://www.infojapon.com:8081/universidades/id/271/687/access>
- <http://www.infojapon.com:8081/universidades/id/271/687/admissions>
- <http://www.infojapon.com:8081/universidades/id/271/687/facilities>
- <http://www.infojapon.com:8081/universidades/id/271/687/support>
- <http://www.infojapon.com:8081/universidades/id/271/687/info>

☒ GET 271	200 OK	infojapon.com:8081
☒ GET access	200 OK	infojapon.com:8081
☒ GET admissions	200 OK	infojapon.com:8081
☒ GET facilities	200 OK	infojapon.com:8081
☒ GET support	200 OK	infojapon.com:8081
☒ GET info	200 OK	infojapon.com:8081

Figura 83 - Peticiones AJAX información departamental

Dependiendo del nivel de información de la página web origen crawleada habrá campos vacíos por lo que el usuario será informado que no hay información disponible en según qué campos.

4.2.3.6 TRABAJO

La página de trabajo.html consta de un formulario de 3 pasos empleando el mismo estilo que en las anteriores páginas y formularios.

Paso 1

Se le indica al usuario que indique la especialidad sobre la que quiere buscar ofertas de trabajo. Además, se visualiza en pantalla información relevante al trabajo en Japón.

The screenshot shows a web page with a header containing the 'INFO JAPÓN' logo and navigation links for ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. Below the header, there are three numbered circles (1, 2, 3) corresponding to the steps of the form:

- 1** Especialidad: A dropdown menu labeled "Selecciona una especialidad".
- 2** Prefectura: A section titled "Tabajo en Japón" with descriptive text about Tokyo's economy and a list of requirements for working in Japan.
- 3** Buscar empleos: A section titled "Trabajo según el visado (Depende del país de origen)" with a list of visa types and their requirements.

Step 1: Especialidad

Selecciona la especialidad sobre la que quieras buscar ofertas de trabajo.

Selecciona una especialidad ▾

Step 2: Tabajo en Japón

El área metropolitana de Tokio y las prefecturas de sus alrededores, es el área urbana más poblada, y es característica por su desenfrenada actividad. Se posiciona como una de las mayores potencias económicas del mundo, con el puesto 18 según el PIB por hora trabajada. Es ícono en tecnología y electrónica, exportador e importador de miles de productos y con un tremendo potencial industrial.

Para trabajar en Japón es necesario cumplir con los siguientes requisitos:

- Tener estudios universitarios relacionados con el puesto de trabajo a desempeñar (En caso contrario 10 años de experiencia laboral demostrable en dicho puesto)
- En general, alto nivel de japonés
- Recomendable y en algunos casos obligatorio tener buen nivel de inglés

Step 3: Trabajo según el visado (Depende del país de origen):

- Visado de turista: no permite trabajar en Japón
- Visado de estudiante: permiso de trabajo (de 28 horas semanales) con la tarjeta de residente. El empleo no debe realizarse en establecimientos que tenga negocios de entretenimiento o diversión, que estén relacionados con el sexo o aquellos perjudiquen o menoscaben los principios morales de los estudiantes.
- Visa de trabajo: Permite trabajar en Japón en tu área de conocimientos. Requiere de un contrato con una empresa japonesa y requiere de renovación en inmigración cada cierto tiempo.

Los extranjeros de las siguientes categorías de residencia están autorizados a trabajar en Japón sin limitaciones para desarrollar actividades remuneradas y pueden cambiar de profesión cuando lo deseen: "residente permanente", "cónyuge de japonés", "cónyuge de residente permanente" y "residente fijo".

Fuente: <http://internacional.universia.net>

Figura 84 - Paso 1 formulario de trabajo

Paso 2

Tal y como ocurre en el módulo de estudios se especifica la prefectura en la que se quieren buscar las ofertas de empleo. Una vez seleccionada la prefectura se mostrará información relacionada con la misma obtenida de Wikipedia.

The screenshot shows a three-step search interface:

- Step 1 (Especialidad):** A dropdown menu labeled "13. Tokyo".
- Step 2 (Prefectura):** A map of Japan with numbered regions corresponding to the dropdown selection. The legend indicates regions by color: Hokkaidō (purple), Tōhoku (pink), Kantō (orange), Chūbu (green), Shikoku (blue), and Kyūshū & Okinawa (grey). The map shows the following numbers: 1 (Hokkaidō), 2 (Tōhoku), 3 (Kantō), 4 (Chūbu), 5 (Shikoku), 6 (Kyūshū & Okinawa).
- Step 3 (Buscar empleos):** A placeholder text "Buscar empleos" at the bottom right.

Below the map, there is descriptive text about Tokyo's status as the capital and its administrative divisions.

Text below the map:

Tokio (東京都, Tōkyō-to, lit. 'capital del este'; pronunciación japonesa: [to̞ ki̞o̞]) es la capital de facto de Japón, localizada en el centro-oeste de la isla de Honshu, concretamente en la región de Kantō. En conjunto forma una de las 47 prefecturas de Japón, aunque su denominación oficial es metrópolis o capital (都 -to). La ciudad es el centro de la política, economía, educación, comunicación y cultura popular del país. Cuenta también con la mayor concentración de sedes corporativas, instituciones financieras, universidades y colegios, museos, teatros, y establecimientos comerciales y de entretenimiento de todo Japón. Con una población que supera los 13 millones de habitantes, se subdivide en 23 barrios (区 -ku); 26 ciudades (市 -shi); un distrito (都 -gun) subdividido en tres pueblos (村 -chō o -machi) y una villa (村 -son o -mura); y cuatro subprefecturas (支厅 -shichō) subdivididas en dos pueblos y siete villas, que representan a varias pequeñas islas al sur de Honshu que se extienden más allá de 1800 km de Shinjuku, capital de la metrópoli y sede de la gobernanza. El centro de Tokio, con sus 23 barrios, ocupa un tercio de la metrópoli, con una población cercana a los 13,23 millones de habitantes; esta área es lo que se conoce internacionalmente como la ciudad de Tokio. En su área metropolitana viven más de 36 millones de habitantes, lo que la convierte en la mayor aglomeración urbana del mundo. A pesar de que Tokio es la romanización más común del nombre en japonés, el nombre de la ciudad es Tokio en español y otros idiomas —entre ellos el alemán y el neerlandés—. En

Figura 85 - Paso 2 formulario de trabajo

Paso 3

Una vez completado el formulario se mostrará una lista con las ofertas de trabajo recibidas.

URL ejemplo: <http://www.infojapon.com/trabajo>

Petición AJAX: www.infojapon.com:8081/trabajo/Tokyo/administrative?page=0

4.2.3.7 VIVIENDA

La página de vivienda.html consta de un formulario de 3 pasos empleando el mismo estilo que las anteriores páginas y formularios.

Paso 1

El usuario, como en anteriores formularios, debe elegir la prefectura donde desea buscar la vivienda. Una vez seleccionada la prefectura se mostrará la información relacionada con la misma obtenida de la Wikipedia.

The screenshot shows a web interface for searching for houses in Japan. At the top, there is a navigation bar with the 'INFO JAPÓN' logo, followed by links for ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. Below the navigation bar, three circular steps are shown: Step 1 (Localización), Step 2 (Características), and Step 3 (Buscar viviendas). The main area is titled 'Prefectura' and contains a dropdown menu where '14. Kanagawa' has been selected. A note below the dropdown says 'Selecciona la prefectura donde quieras buscar las viviendas.' To the right of the dropdown is a large map of Japan with each prefecture numbered from 1 to 47. A legend on the right side identifies the regions: Hokkaidō (purple), Tōhoku (pink), Kantō (orange), Chūbu (green), Kansai (light green), Shikoku (blue), and Kyūshū & Okinawa (grey). The map shows the following prefecture numbers: 1. Hokkaidō, 2. Aomori, 3. Iwate, 4. Miyagi, 5. Akita, 6. Yamagata, 7. Fukushima, 8. Ibaraki, 9. Tochigi, 10. Gunma, 11. Saitama, 12. Chiba, 13. Tokyo, 14. Kanagawa, 15. Niigata, 16. Toyama, 17. Ishikawa, 18. Fukui, 19. Yamanashi, 20. Nagano, 21. Gifu, 22. Shizuoka, 23. Aichi, 24. Mie, 25. Shiga, 26. Kyoto, 27. Osaka, 28. Hyogo, 29. Nara, 30. Wakayama, 31. Tottori, 32. Shimane, 33. Okayama, 34. Hiroshima, 35. Yamaguchi, 36. Tokushima, 37. Kagawa, 38. Ehime, 39. Kochi, 40. Fukuoka, 41. Saga, 42. Nagasaki, 43. Kumamoto, 44. Oita, 45. Miyazaki, 46. Kagoshima, 47. Okinawa.

Figura 86 - Paso 1 búsqueda de viviendas

Paso 2

En el segundo paso el usuario debe especificar una serie de características opcionales que formarán los parámetros de la petición a la API REST.

INFO JAPÓN

ARTÍCULOS ESTUDIAR TRABAJAR VIVIR PROYECTO IJ

1 Localización 2 Características 3 Buscar viviendas

Características optionales de la búsqueda

Elige el rango de precios sobre el que se buscarán las casas

20,000¥ 60,000¥

BAJOS COSTES INICIALES SIN AVAL

Elige el tamaño de las casas

Metros cuadrados Habitaciones

SIN SEÑAL INICIAL ADMITEN MASCOTAS

Elige el tipo de construcción y su antigüedad así como la distancia hasta la estación más cercana

Tipo de construcción Distancia a estaciór Antigüedad

SIN DEPOSITO PERIODICO DE ESTANCIA CORTO

SIN GASTO DE AGENCIA FURNISHED

INTERNET WIFI

PAGO CON TARJETA

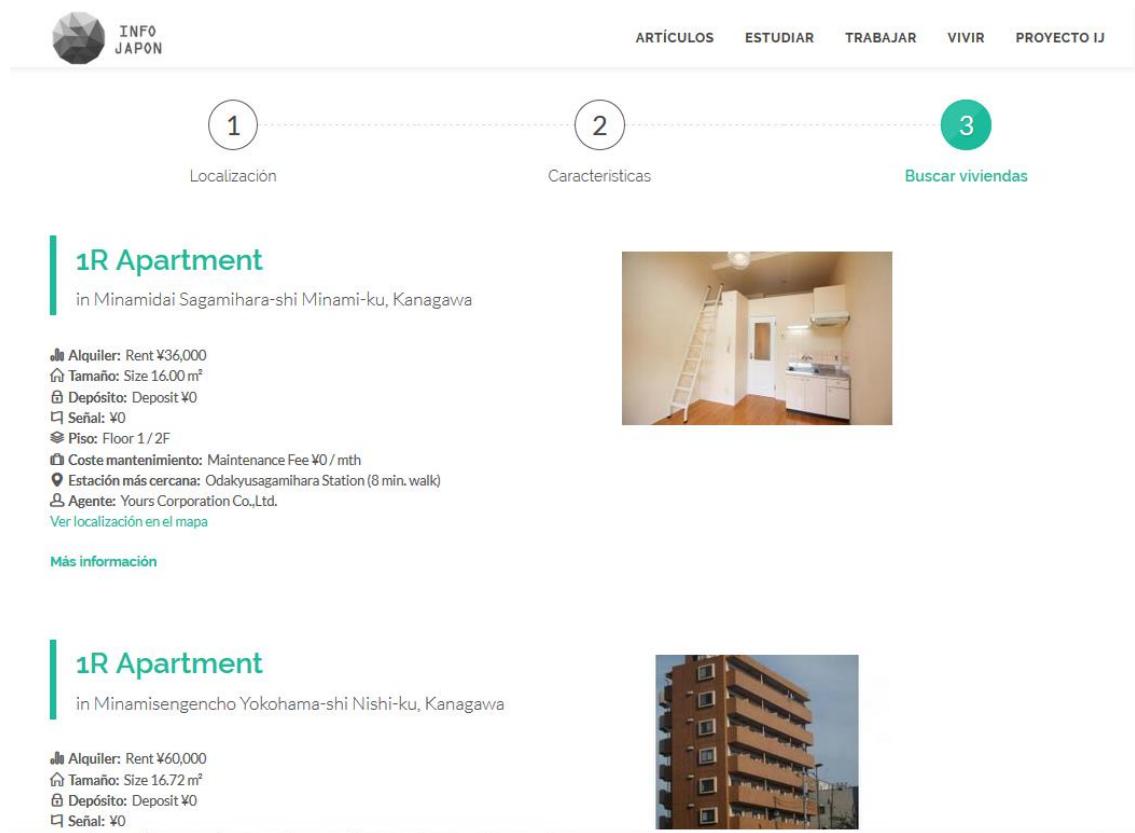
Figura 87 - Paso 2 búsqueda de viviendas

Paso 3

Una vez enviada la petición al hacer click en el paso 3 se mostrará una serie de viviendas en función de los parámetros de búsqueda rellenados.

Petición asíncrona de ejemplo (AJAX):

http://www.infojapon.com:8081/vivir/Kanagawa?page=1&min_price=20000&max_price=60000&min_meter=&rooms=&distance_station=&building_type=&building_age=&low_initial_costs=&no_guarantor=&no_key_money=&pets=&no_deposit=&short_term_ok=&no_agency_fee=&furnished=&internet=&wifi=&credit_card=1



INFO JAPÓN

ARTÍCULOS ESTUDIAR TRABAJAR VIVIR PROYECTO IJ

1 Localización 2 Características 3 Buscar viviendas

1R Apartment
in Minamidai Sagamihara-shi Minami-ku, Kanagawa

Alquiler: Rent ¥36,000
Tamaño: Size 16.00 m²
Depósito: Deposit ¥0
Señal: ¥0
Piso: Floor 1 / 2F
Coste mantenimiento: Maintenance Fee ¥0 / mth
Estación más cercana: Odakyusagamihara Station (8 min. walk)
Agente: Yours Corporation Co.,Ltd.
[Ver localización en el mapa](#)

[Más información](#)

1R Apartment
in Minamisengencho Yokohama-shi Nishi-ku, Kanagawa

Alquiler: Rent ¥60,000
Tamaño: Size 16.72 m²
Depósito: Deposit ¥0
Señal: ¥0

Figura 88 - Paso 3 listado de viviendas

4.2.3.8 CONTACTO

La página de contacto.html contiene información referente al proyecto así como información de contacto para resolución de dudas e incidencias.

The screenshot shows the contact page of the InfoJapon website. At the top, there's a header with the 'INFO JAPÓN' logo and navigation links for ARTÍCULOS, ESTUDIAR, TRABAJAR, VIVIR, and PROYECTO IJ. The main content area has a heading 'Proyecto InfoJapón' followed by a paragraph about the project's goal. Below that is another paragraph about the current state of the project. To the right of the text is a photograph of a traditional Japanese garden with cherry blossoms and a stone bridge over water. The contact form section includes fields for name, email, message, and a checkbox for sending feedback. There's also a note about filling out a survey.

INFO JAPÓN

ARTÍCULOS ESTUDIAR TRABAJAR VIVIR PROYECTO IJ

Proyecto InfoJapón

InfoJapón nace con el objetivo de ser una plataforma que permita conectar al usuario con información actualizada sobre Japón de manera sencilla haciendo uso de las últimas tecnologías. De esta manera el proyecto pretende centralizar información de interés referente Japón y crecer en base a la demanda de los propios usuarios.

En la actualidad hay muchísima información en la red cuyo acceso se torna complejo si no sale en los primeros resultados de Google. En InfoJapón el usuario podrá acceder a esta información de manera simplificada.

Estado actual del proyecto

Actualmente el proyecto se encuentra en fase beta. Siendo un proyecto ambicioso que pretende abarcar varios ámbitos de información (blogs, videoblogs, estudios...) el crecimiento es progresivo y tiene en cuenta las opiniones de los propios usuarios para su mejora por lo que estaríamos encantados de conocer tu opinión (ver información de contacto).

Contacto

Siquieres que tu blog aparezca en infoJapón, quieres saber más sobre el proyecto o quieres proporcionar retroalimentación en base a errores que has podido observar durante tu experiencia en infoJapón(*) (tanto en el cliente móvil como en el cliente web) escríbenos a infojaponproject@gmail.com

*Si vais a proporcionar retroalimentación es imprescindible rellenes y envíeis la siguiente plantilla:

- Dispositivo utilizado: (Ordenador / tablet / móvil / Otro (indicar))
- Cliente utilizado: (web / aplicación android)
- Navegador utilizado y su versión (en caso de web): (ej: Google Chrome v2.3.4)
- Descripción del problema: (Intentar explicar los pasos que habéis tomado hasta la aparición del problema)

Figura 89 - Pantalla de contacto

4.2.4 CAMBIOS PARA ADAPTAR A LA APLICACIÓN MÓVIL

Hubo que realizar una serie de cambios para exportar la aplicación a la herramienta software Intel XDK y poder generar el .apk correspondiente.

- Añadir formato a las enlaces (.html) ya que el .htaccess forma parte del servidor el propio paquete no lo sabría interpretar.
- Minimizar css y js así como comprimir imágenes y eliminar páginas como la de contacto o el botón de google play para reducir el tamaño de la aplicación.

Una vez en el Intel XDK se exportó el proyecto web a proyecto APP lo que generó nuevo código en la carpeta del proyecto de la app. A partir de aquí para generar la aplicación simplemente fue necesario generar un certificado de desarrollador y llenar la información respectiva a la APP.

Capítulo 5. PRUEBAS Y EVALUACIÓN

5.1 PRUEBAS MODELO

Se ha realizado una plantilla de Test para evaluar el crawler de las universidades. Esta plantilla permite determinar si la aplicación ha podido generar la URL correctamente y si los resultados coinciden con los de la URL de búsqueda real. Sirve por tanto para los crawlers que operan sobre la misma página tales como el de búsqueda de universidades, de colegios de posgrado y de colegios de formación profesional.

Ejemplo:

```
Running com.puntojapon.test.CrawlerTest
Running university crawler test...
urlStudies -> &u%5Bac%5D=101-102-103-104-105-106-107-111-151-160-166
Crawleando...
http://www.jpss.jp/en/search/?tb=1&a%5Bnm%5D=&a%5Bfw%5D=&u%5Bfc%5D=&u%5Bdp%5D=&u%5But%5D%5B%5D=1&u%5Bej%5D%5B%5D=1&u%5Ben%5D%5B%5D=2&u%5Bac%5D=101-102-103-104-105-106-107-111-151-160-166&a%5Bpf%5D=&search.x=177&search.y=11&search=search

Universidades: 4
Got: 4 through
http://www.jpss.jp/en/search/?tb=1&a%5Bnm%5D=&a%5Bfw%5D=&u%5Bfc%5D=&u%5Bdp%5D=&u%5But%5D%5B%5D=1&u%5Bej%5D%5B%5D=1&u%5Ben%5D%5B%5D=2&u%5Bac%5D=101-102-103-104-105-106-107-111-151-160-166&a%5Bpf%5D=&search.x=177&search.y=11&search=search

Expected: 4 through
http://www.jpss.jp/en/search/?tb=1&a%5Bnm%5D=&a%5Bfw%5D=&u%5Bfc%5D=&u%5Bdp%5D=&u%5But%5D%5B%5D=1&u%5Bej%5D%5B%5D=1&u%5Ben%5D%5B%5D=2&u%5Bac%5D=101-102-103-104-105-106-107-111-151-160-166&a%5Bpf%5D=&search.x=123&search.y=9&search=search
```

Como se puede observar se puede especificar la página web original y, en base a los parámetros de entrada, obtener la información de la página web cuya URL ha sido armada por el crawler para cotejarlas y determinar si poseen la misma información. En caso de que esta batería de test fallara daría lugar a que algún tramo del crawler ha cambiado y por tanto habría que especificar otra vez que información extraer y de qué parte en base a los cambios de la web origen.

5.2 PRUEBAS DE CONTROLADOR Y SERVIDOR

Se han efectuado varios test de rendimiento contra la API REST mediante la herramienta Apache JMeter para observar el rendimiento del servidor dependiendo del volumen de visitas y el correcto funcionamiento de la API [30].

5.2.1 TEST 1: /artículos

Considerando que la petición GET /artículos es la que se va a ejecutar cada vez que se accede al index de la página web y que posee los artículos relacionados con los diferentes blogs que es la funcionalidad que se estima va a ser la más utilizada por los usuarios, se considera entonces que la afluencia de usuarios se concentrará en esta misma por lo que se han ejecutado 3 tests.

5.2.1.1 *Mejor caso*

5.2.1.1.1 Descripción

- Acción: GET /artículos
- Parámetros:
- Número de hilos (peticiones): 50 peticiones
- Periodo de subida: 10 segundos

5.2.1.1.2 Resultados

- Pico máximo CPU: 14,4%
- Media: 231 ms
- Mediana: 223 ms
- Min: 211 ms
- Max: 390 ms
- % Error: 0,00%
- Rendimiento: 5,0/sec
- Kb/sec: 370,1

5.2.1.1.3 Conclusión

Siendo 50 usuarios una media adecuada de peticiones concurrentes dado el tamaño de la aplicación el resultado es satisfactorio. El tiempo medio que el usuario necesita para cargar los artículos es de 231ms mientras que en el peor de los casos se puede demorar hasta los 390ms. Con un pico de 14,4% de carga de procesador se puede determinar que el test es satisfactorio.

5.2.1.2 *Caso medio*

5.2.1.2.1 Descripción

- Acción: GET /artículos
- Parámetros:
- Número de hilos (peticiones): 120 peticiones
- Periodo de subida: 10 segundos

5.2.1.2.2 Resultados

- Pico máximo CPU: 26,7%
- Media: 235 ms
- Mediana 235 ms
- Min: 210 ms
- Máx: 260 ms
- % Error: 0,00%
- Rendimiento: 14,8/sec
- Kb/sec: 1092,2

5.2.1.2.3 Conclusión

Siendo 120 usuarios una afluencia considerable de peticiones concurrentes dado el tamaño de la aplicación el resultado es satisfactorio. El tiempo medio que el usuario necesita para cargar los artículos es de 235ms mientras que en el peor de los casos se puede demorar hasta los

260ms. En comparación con el test 1.1 el comportamiento del servidor es positivo llegando un pico de carga de procesador del 26,7%.

5.2.1.3 *Peor caso*

5.2.1.3.1 Descripción

- Acción: GET /artículos
- Parámetros:
- Número de hilos (peticiones): 500 peticiones
- Periodo de subida: 10 segundos

5.2.1.3.2 Resultados

- Pico máximo CPU: 62%
- Media: 5811 ms
- Mediana: 4822 ms
- Min: 257 ms
- Máx: 52685 ms
- % Error: 0,00%
- Rendimiento 9,4/sec
- Kb/sec: 691,5

5.2.1.3.3 Conclusión

Siendo 500 peticiones concurrentes podemos observar que, aunque el procesador no se satura y puede responderlas a todas (pico de 62% de carga de CPU), habrá muchas peticiones que se demoren bastante en el tiempo. Aun así, siendo la media 5811 ms se puede determinar que el servidor va a responder bien frente a una fuerte demanda en la página principal.

5.2.2 TEST 2: /artículos/buscar

Se considera que el recurso GET /artículos/buscar va a tener un nivel de uso medio. El nivel de carga de procesador se espera que sea bajo ya que requiere de hacer consultas al servicio de ElasticSearch sin necesidad de aplicar ninguna transformación.

5.2.2.1 *Caso medio*

5.2.2.1.1 Descripción

- Acción: GET /artículos/buscar
- Parámetros: título=Festivales
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.2.1.2 Resultados

- Pico máximo CPU: 2,7%
- Media: 88 ms
- Mediana: 85 ms
- Min: 77 ms
- Máx: 104 ms
- % Error: 0,00%

- Rendimiento 2,1/sec
- Kb/sec: 24,9

5.2.2.1.3 Conclusión

Con 20 peticiones concurrentes de búsquedas de artículos en el repositorio de documentos de ElasticSearch el procesador no sube de 2,7% proporcionando un rendimiento máximo en la respuesta de 104 ms por lo que se puede concluir que, aun duplicando el número de peticiones el sistema no se va a resentir.

5.2.3 TEST 3: /universidades/{prefecture}/{typeStudies}

Se considera que el recurso GET /artículos/prefecture/typeStudies va a tener un nivel de uso medio. El nivel de carga de procesador se espera que sea alto ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.3.1 Caso medio

5.2.3.1.1 Descripción

- Acción: GET /universidades/Kanagawa/Literature
- Parámetros:
admisionMonth=admisionUni=deadLine=eju=engExam=nameUni=typeUni=
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.3.1.2 Resultados

- Pico máximo CPU: 25,7%
- Media: 1972ms
- Mediana: 1909 ms
- Min: 1861 ms
- Máx: 2262ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 19,6

5.2.3.1.3 Conclusión

Con 20 peticiones concurrentes de búsquedas de universidades y teniendo en cuenta el proceso que conlleva para el procesador, un máximo de 25,7% de uso de procesador y un tiempo máximo de consulta de 2262ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.4 TEST 4: /universidades/id/{id}

Se considera que el recurso GET /universidades/id/{id} va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea media ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.4.1 Caso medio

5.2.4.1.1 Descripción

- Acción: GET /universidades/id/271
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.4.1.2 Resultados

- Pico máximo CPU: 22,6%
- Media: 1633 ms
- Mediana: 1573 ms
- Min: 1543 ms
- Máx: 1874ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 6,0

5.2.4.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de universidades y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 22,6% de uso de procesador y un tiempo máximo de consulta de 1874ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.5 TEST 5: /universidades/id/{idUniversity}/{idFaculty}/access

Se considera que el recurso GET /universidades/id/{idUniversity}/{idFaculty}/access va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas aunque dichos procesos, específicamente, no requieren grandes extracciones de datos.

5.2.5.1 Caso medio

5.2.5.1.1 Descripción

- Acción: GET /universidades/id/357/3507/ access
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.5.1.2 Resultados

- Pico máximo CPU: 17,1%
- Media: 1636ms
- Mediana: 1571ms

- Min: 1558 ms
- Máx: 1858ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 2,2

5.2.5.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de información sobre el acceso a una universidad determinada teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un pico de 17,1% de uso y un tiempo de respuesta de como máximo 1888ms es un rendimiento muy satisfactorio.

5.2.6 TEST 6: /universidades/id/{idUniversity}/{idFaculty}/admissions

Se considera que el recurso GET /universidades/id/{idUniversity}/{idFaculty}/admissions va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas aunque dichos procesos, específicamente, no requieren grandes extracciones de datos.

5.2.6.1 Caso medio

5.2.6.1.1 Descripción

- Acción: GET /universidades/id/357/3507/admissions
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.6.1.2 Resultados

- Pico máximo CPU: 11,7%
- Media: 1633ms
- Mediana: 1580ms
- Min: 1562 ms
- Máx: 1888ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 3,5

5.2.6.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de información sobre las admisiones de una universidad determinada teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un pico de 11,7% de uso y un tiempo de respuesta de como máximo 1888ms es un rendimiento muy satisfactorio.

5.2.7 TEST 7: /universidades/id/{idUniversity}/{idFaculty}/facilities

Se considera que el recurso GET /universidades/id/{idUniversity}/{idFaculty}/facilities va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas aunque dichos procesos, específicamente, no requieren grandes extracciones de datos.

5.2.7.1 Caso medio

5.2.7.1.1 Descripción

- Acción: GET /universidades/id/357/3507/facilities
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.7.1.2 Resultados

- Pico máximo CPU: 12,1%
- Media: 1629ms
- Mediana: 1580ms
- Min: 1566 ms
- Máx: 1871ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 12,3

5.2.7.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de información sobre las instalaciones de una universidad determinada teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un pico de 12,1% de uso y un tiempo de respuesta de como máximo 1871ms es un rendimiento muy satisfactorio.

5.2.8 TEST 8:: /universidades/id/{idUniversity}/{idFaculty}/info

Se considera que el recurso GET /universidades/id/{idUniversity}/{idFaculty}/info va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea alto ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.8.1 Caso medio

5.2.8.1.1 Descripción

- Acción: GET /universidades/id/357/3507/info
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.8.1.2 Resultados

- Pico máximo CPU: 12,7%

- Media: 1537ms
- Mediana: 1382ms
- Min: 1351 ms
- Máx: 1901ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 12,3

5.2.8.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de información básica de una universidad determinada teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un pico de 12,7% de uso y un tiempo de respuesta de como máximo 1901ms es un rendimiento muy satisfactorio.

5.2.9 TEST 9: /universidades/{id}/{idUniversity}/{idFaculty}/support

Se considera que el recurso GET /universidades{id}/{idUniversity}/{idFaculty}/support va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas aunque dichos procesos, específicamente, no requieren grandes extracciones de datos.

5.2.9.1 Caso medio

5.2.9.1.1 Descripción

- Acción: GET /universidades{id}/357/3507/support
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.9.1.2 Resultados

- Pico máximo CPU: 18,2%
- Media: 1533ms
- Mediana: 1372ms
- Min: 1335 ms
- Máx: 1906ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 9,2

5.2.9.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de información de apoyo al estudiante de una universidad determinada teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un pico de 18,2% de uso y un tiempo de respuesta de como máximo 1906ms es un rendimiento satisfactorio.

5.2.10 TEST 10: /posgrado/{prefecture}/{typeStudies}

Se considera que el recurso GET /posgrado/{prefecture}/{typeStudies} va a tener un nivel de uso medio. El nivel de carga de procesador se espera que sea alto ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.10.1 Caso medio

5.2.10.1.1 Descripción

- Acción: GET /posgrado/Tokyo
- Parámetros: nameGrad=&typeGrad=master&typeCourse=&englishCourse=
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.10.1.2 Resultados

- Pico máximo CPU: 31,7%
- Media: 27348ms
- Mediana: 28020 ms
- Min: 23537 ms
- Máx: 29836ms
- % Error: 0,00%
- Rendimiento 32,7/min
- Kb/sec: 62,9

5.2.10.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de listas de colegios de posgrado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 31,7% de uso de procesador. Debido a que la profundidad de la búsqueda se aprecia una demora en el tiempo llegando a tomar como máximo 29836ms lo que indica que la función debe ser optimizada.

5.2.11 TEST 11: /posgrado/id/{id}

Se considera que el recurso GET /posgrado/id/{id} va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea media ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.11.1 Caso medio

5.2.11.1.1 Descripción

- Acción: GET /posgrado/id/271
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.11.1.2 Resultados

- Pico máximo CPU: 24,6%
- Media: 1598 ms
- Mediana: 1544 ms

- Min: 1527 ms
- Máx: 1848ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 1,4

5.2.11.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de colegios de posgrado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 24,6% de uso de procesador y un tiempo máximo de consulta de 1874ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.12 TEST 12: /posgrado/id/{idGraduate}/{idFaculty}/access

Se considera que el recurso GET /posgrado/id/{idGraduate}/{idFaculty}/access va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.12.1 Caso medio

5.2.12.1.1 Descripción

- Acción: GET /posgrado/id/240/1765/access
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.12.1.2 Resultados

- Pico máximo CPU: 22,3%
- Media: 7735ms
- Mediana: 7625ms
- Min: 3764ms
- Máx: 11702ms
- % Error: 0,00%
- Rendimiento 1,5/sec
- Kb/sec: 2,8

5.2.12.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información de accesos de un colegio de posgrado determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 22,3% de uso de procesador y un tiempo máximo de consulta de 1887ms es un resultado normal teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones ni grandes tiempos de espera ya que debe cargar imágenes externas a la aplicación.

5.2.13 TEST 13: /posgrado/id/{idGraduate}/{idFaculty}/admissions

Se considera que el recurso GET /posgrado/id/{idGraduate}/{idFaculty}/admissions va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.13.1 Caso medio

5.2.13.1.1 Descripción

- Acción: GET /posgrado/id/240/1765/ admissions
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.13.1.2 Resultados

- Pico máximo CPU: 15,9%
- Media: 1676ms
- Mediana: 1620ms
- Min: 1594ms
- Máx: 1887ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 27,8

5.2.13.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información de admisiones de un colegio de posgrado determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 17,1% de uso de procesador y un tiempo máximo de consulta de 1887ms es satisfactorio teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.14 TEST 14: /posgrado/id/{idGraduate}/{idFaculty}/facilities

Se considera que el recurso GET /posgrado/id/{idGraduate}/{idFaculty}/facilities va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.14.1 Caso medio

5.2.14.1.1 Descripción

- Acción: GET /posgrado/id/240/1765/facilities
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.14.1.2 Resultados

- Pico máximo CPU: 17,1%

- Media: 1629ms
- Mediana: 1572ms
- Min: 1536ms
- Máx: 1887ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 4,4

5.2.14.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información de las instalaciones de un colegio de posgrado determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 17,1% de uso de procesador y un tiempo máximo de consulta de 1887ms es satisfactorio teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.15 TEST 15: /posgrado/id/{idGraduate}/{idFaculty}/info

Se considera que el recurso GET /posgrado/id/{idGraduate}/{idFaculty}/info va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.15.1 Caso medio

5.2.15.1.1 Descripción

- Acción: GET /posgrado/id/240/1765/info
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.15.1.2 Resultados

- Pico máximo CPU: 14,8%
- Media: 1621ms
- Mediana: 1556ms
- Min: 1541 ms
- Máx: 1873ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 15,5

5.2.15.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información básica de un colegio de posgrado determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 14,8% de uso de procesador y un tiempo máximo de consulta de 1873ms es satisfactorio teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.16 TEST 16: /posgrado/id/{idGraduate}/{idFaculty}/support

Se considera que el recurso GET /posgrado/id/{idGraduate}/{idFaculty}/support va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.16.1 Caso medio

5.2.16.1.1 Descripción

- Acción: GET /posgrado/id/240/1765/support
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.16.1.2 Resultados

- Pico máximo CPU: 14,5%
- Media: 1637ms
- Mediana: 1588ms
- Min: 1551 ms
- Máx: 1902ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 18,5

5.2.16.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información de apoyo al estudiante de un colegio de posgrado determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 12,5% de uso de procesador y un tiempo máximo de consulta de 1902ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.17 TEST 17: /fp/{prefecture}

Se considera que el recurso GET /fp/{prefecture} va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea medio ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.17.1 Caso medio

5.2.17.1.1 Descripción

- Acción: GET /fp/Tokyo
- Parámetros: nameTech=
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.17.1.2 Resultados

- Pico máximo CPU: 17,6%
- Media: 1174 ms

- Mediana: 1113 ms
- Min: 1112 ms
- Máx: 1398ms
- % Error: 0,00%
- Rendimiento 1,9/sec
- Kb/sec: 23,5

5.2.17.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la lista de colegios de formación profesional y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 17,6% de uso de procesador y un tiempo máximo de consulta de 1398ms es un rendimiento muy satisfactorio teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.18 TEST 18: /fp/{id}

Se considera que el recurso GET /fp/{id} va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea media ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.18.1 Caso medio

5.2.18.1.1 Descripción

- Acción: GET /fp/{id}/3
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.18.1.2 Resultados

- Pico máximo CPU: 29,6%
- Media: 1616 ms
- Mediana: 1556 ms
- Min: 1530 ms
- Máx: 1824ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 9,7

5.2.18.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de colegios de formación profesional y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 29,6% de uso de procesador y un tiempo máximo de consulta de 1824ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.19 TEST 20: /fp/id/{idTechSchool}/{idFaculty}/admissions

Se considera que el recurso GET /fp/id/{idTechSchool}/{idFaculty}/admissions va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.19.1 Caso medio

5.2.19.1.1 Descripción

- Acción: GET /fp/id/6/admissions
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.19.1.2 Resultados

- Pico máximo CPU: 21,2%
- Media: 1610 ms
- Mediana: 1543 ms
- Min: 1528 ms
- Máx: 1848ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 3,0

5.2.19.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información de admisiones de un colegio de formación profesional determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 29,6% de uso de procesador y un tiempo máximo de consulta de 1824ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.20 TEST 22: /fp/id/{idTechSchool}/{idFaculty}/info

Se considera que el recurso GET /fp/id/{idTechSchool}/{idFaculty}/info va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.20.1 Caso medio

5.2.20.1.1 Descripción

- Acción: GET /fp/id/6/info
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.20.1.2 Resultados

- Pico máximo CPU: 31,7%
- Media: 1626ms
- Mediana: 1568 ms
- Min: 1535 ms
- Máx: 1862ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 19,7

5.2.20.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información básica de un colegio de formación profesional determinado y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 29,6% de uso de procesador y un tiempo máximo de consulta de 1824ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.21 TEST 23: /fp/{id}/{idTechSchool}/{idFaculty}/support

Se considera que el recurso GET /fp/{id}/{idTechSchool}/{idFaculty}/support va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea bajo ya que aunque requiere de los procesos ETL para la extracción y carga de datos de páginas web externas la información a extraer es poca.

5.2.21.1 Caso medio

5.2.21.1.1 Descripción

- Acción: GET /fp/{id}/support
- Parámetros:
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.21.1.2 Resultados

- Pico máximo CPU: 16,2%
- Media: 1601ms
- Mediana: 1556 ms
- Min: 1506 ms
- Máx: 1864ms
- % Error: 0,00%
- Rendimiento 1,8/sec
- Kb/sec: 1,8

5.2.21.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de la información de apoyo al estudiante de un colegio de formación profesional determinado y teniendo en cuenta el proceso que

conlleva para el procesador debido al proceso ETL, un máximo de 29,6% de uso de procesador y un tiempo máximo de consulta de 1824ms es asumible teniendo en cuenta que incluso duplicando las peticiones el sistema podrá soportar sin ralentizaciones.

5.2.22 TEST 24: /fp/{ prefecture}

Se considera que el recurso GET /fp/{ prefecture} va a tener un nivel de uso bajo. El nivel de carga de procesador se espera que sea medio ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.22.1 Caso medio

5.2.22.1.1 Descripción

- Acción: GET /fp/Tokyo
- Parámetros: nameTech=
- Número de hilos (peticiones): 20 peticiones
- Periodo de subida: 10 segundos

5.2.22.1.2 Resultados

- Pico máximo CPU: 9,9%
- Media: 1164 ms
- Mediana: 1130 ms
- Min: 1105 ms
- Máx: 1394ms
- % Error: 0,00%
- Rendimiento 1,9/sec
- Kb/sec: 23,4

5.2.22.1.3 Conclusión

Con 20 peticiones concurrentes de visualización de colegios de formación profesional y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, aunque de menor calibre, un máximo de 9,9% de uso de procesador y un tiempo máximo de consulta de 1105ms es un rendimiento más que satisfactorio.

5.2.23 TEST 25: /escuelaidiomas/id/{idSchool}

Se considera que el recurso GET /escuelaidiomas/id/{id} va a tener un nivel de uso alto. El nivel de carga de procesador se espera que sea media ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas.

5.2.23.1 Caso medio

5.2.23.1.1 Descripción

- Acción: GET /escuelaidiomas/id/88
- Parámetros:
- Número de hilos (peticiones): 40 peticiones

- Periodo de subida: 10 segundos

5.2.23.1.2 Resultados

- Pico máximo CPU: 20,01%
- Media: 1331 ms
- Mediana: 1326 ms
- Min: 1273 ms
- Máx: 1424ms
- % Error: 0,00%
- Rendimiento 3,6/sec
- Kb/sec: 19,3

5.2.23.1.3 Conclusión

Con 40 peticiones concurrentes para la visualización de información de colegios de japonés y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, un máximo de 20,01% de uso de procesador y un tiempo máximo de consulta de 1424ms es un rendimiento satisfactorio para la afluencia esperada.

5.2.24 TEST 26: /escuelaldiomas/{area}

Se considera que el recurso GET /escuelaldiomas/{área} va a tener un nivel de uso alto. El nivel de carga de procesador se espera que sea media ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas aunque este proceso específicamente requiere de una carga baja del mismo debido a la poca extracción de datos a realizar.

5.2.24.1 Caso medio

5.2.24.1.1 Descripción

- Acción: GET /escuelaldiomas/tokyo
- Parámetros:
- Número de hilos (peticiones): 40 peticiones
- Periodo de subida: 10 segundos

5.2.24.1.2 Resultados

- Pico máximo CPU: 19,6%
- Media: 1836 ms
- Mediana: 1875 ms
- Min: 1577 ms
- Máx: 2160ms
- % Error: 0,00%
- Rendimiento 3,4/sec
- Kb/sec: 62,4

5.2.24.1.3 Conclusión

Con 40 peticiones concurrentes de visualización de la lista de colegios de japonés y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, aunque de menor

calibre, un máximo de 19,6% de uso de procesador y un tiempo máximo de consulta de 2160ms es un rendimiento satisfactorio.

5.2.25 TEST 27: /trabajo/{prefecture}/{specialty}

Se considera que el recurso GET /trabajo/{prefecture}/{specialty} va a tener un nivel de uso alto. El nivel de carga de procesador se espera que sea baja ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web externas aunque este proceso específicamente requiere de una carga baja del mismo debido a la poca extracción de datos a realizar.

5.2.25.1 Caso medio

5.2.25.1.1 Descripción

- Acción: GET /trabajo/Kanagawa/accounting
- Parámetros: page=0
- Número de hilos (peticiones): 40 peticiones
- Periodo de subida: 10 segundos

5.2.25.1.2 Resultados

- Pico máximo CPU: 14,3%
- Media: 1356 ms
- Mediana: 1372 ms
- Min: 1211 ms
- Máx: 1654ms
- % Error: 0,00%
- Rendimiento 3,4/sec
- Kb/sec: 62,4

5.2.25.1.3 Conclusión

Con 40 peticiones concurrentes para listar ofertas de trabajo y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, aunque de menor calibre, un máximo de 14,3% de uso de procesador y un tiempo máximo de consulta de 1654ms es un rendimiento satisfactorio.

5.2.26 TEST 28: /vivir/{prefecture}

Se considera que el recurso GET /vivir/{prefecture} va a tener un nivel de uso alto. El nivel de carga de procesador se espera que sea media ya que requiere de los procesos ETL para la extracción y carga de datos de páginas web.

5.2.26.1 Caso medio

5.2.26.1.1 Descripción

- Acción: GET /trabajo/Kanagawa/accounting

- Parámetros:
page=1&min_price=20000&max_price=30000&min_meter=&rooms=&distance_statio
n=&building_type=&building_age=&low_initial_costs=&no_guarantor=&no_key_mon
ey=&pets=&no_deposit=&short_term_ok=&no_agency_fee=&furnished=&internet=&
wifi=&credit_card=
- Número de hilos (peticiones): 40 peticiones
- Periodo de subida: 10 segundos

5.2.26.1.2 Resultados

- Pico máximo CPU: 29,1%
- Media: 8957 ms
- Mediana: 9035 ms
- Min: 5035 ms
- Máx: 12424ms
- % Error: 0,00%
- Rendimiento 3,4/sec
- Kb/sec: 62,4

5.2.26.1.3 Conclusión

Con 40 peticiones concurrentes para listar viviendas y teniendo en cuenta el proceso que conlleva para el procesador debido al proceso ETL, aunque de menor calibre, un máximo de 29,1% de uso de procesador es un resultado de rendimiento satisfactorio, por otra parte un tiempo máximo de 12424ms y 8957ms de media es un conjunto de tiempos reales debido la naturaleza de la búsqueda y la carga de imágenes ajenas al servidor pero optimizables.

5.3 CONCLUSIÓN DE PRUEBAS

La batería de pruebas del módulo de estudios se lleva a cabo siempre que se actualice el código fuente mediante Apache Maven lo que permite prevenir cambios que estropeen la funcionalidad de la aplicación.

Según los resultados de los test de rendimiento las búsquedas que más se demoran en el tiempo son las búsquedas en profundidad referentes al listado de universidades, colegios de posgrado y colegios de formación profesional debido al número de páginas que tienen que crawlear. Respecto al uso del procesador, para una afluencia de usuarios media se ha demostrado que el sistema puede hacer frente sin generar cuellos de botella.

Capítulo 6. DESPLIEGUE

6.1 DESPLIEGUE EN SERVIDOR

Debido a los requisitos de la aplicación la misma se va a alojar en un servidor virtual en la nube el cual proporcione un rendimiento equilibrado para que no genere cuellos de botella en las peticiones al cliente web y la API REST.

Tras la batería de test de hardware en el servidor virtual de pruebas se determinó la contratación de un servidor con las mismas características de hardware, localizado en Gravelines (Francia) con la empresa OVH:

- Sistema de virtualización: KVM OpenStack
- Procesador: 1 vCore Xeon E5v3 @2,4 GHz
- Memoria: 2GB DDR3
- Almacenamiento: SSD 10GB (RAID 10 local)
- Acuerdo de nivel de servicio: 99,95% de disponibilidad del servidor
- Escalabilidad ascendente
 - Procesador: hasta 2 vCore Xeon E5v3 @2,4 GHz
 - Memoria: hasta 8GB DDR3
 - Almacenamiento: SSD hasta 40GB (RAID 10 local)
- Ancho de banda: 100Mbps – Tráfico ilimitado
- Protección Anti-DDoS incluida

El software que se ha instalado es el siguiente:

- Sistema operativo: Ubuntu server 16.04 64 bits
- Control de versiones: Git
- JVM: OpenJDK 64-Bit (build 25.45-b02, mixed mode)
- Elastic Search
- Logstash
- Kibana
- Apache web server (Apache2) y PHP7.0
- Logwatch
- Apache maven
- Haveged

6.1.1 SECURIZACIÓN DEL SERVIDOR

6.1.1.1 CAMBIO DE CONTRASEÑAS ROOT Y ACTUALIZACIÓN DE PAQUETES

Tras contratar el servidor se recibió la IP del servidor y las credenciales para su acceso como root. Accediendo por putty (cliente SSH para Windows) lo primero que hizo fue cambiar la contraseña del usuario root:

```
# passwd
```

```
root@vps11:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Figura 90 - Cambio de contraseña en servidor

A continuación fue necesario actualizar los repositorios y actualizar el sistema aplicando los últimos parches.

```
apt-get update
apt-get upgrade
```

6.1.1.2 CREACIÓN DE USUARIO Y OTORGACIÓN DE PERMISOS SUDO

Tras ello se creó el usuario con el que implementaríamos todo el despliegue (ya que, por seguridad, es mejor no tratar con el usuario root siempre). Le asignamos como shell bash y le establecimos una contraseña.

```
useradd deploy
mkdir /home/deploy
mkdir /home/deploy/.ssh
chmod 700 /home/deploy/.ssh
usermod -s /bin/bash deploy
passwd deploy
```

A continuación, para evitar tener que iniciar sesión como root metimos al usuario deploy en el grupo sudo. A este grupo le dimos poder para poder ascender con permisos de sudo. Para ello realizamos lo siguiente:

```
visudo
```

Y dentro, quedó de la siguiente manera:

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Figura 91 - Fichero de configuración sudoers

A continuación introducimos al usuario deploy en el grupo sudo:

```
usermod -aG sudo deploy
```

6.1.1.3 SECURIZACIÓN SSH

Para añadir un plus a la securización en las conexiones SSH cambiamos que en vez de escuchar por el puerto 22, el servicio SSH escuchara en el puerto 5555.

```
sudo vi /etc/ssh/sshd_config
```

```
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 5555
```

Figura 92 - Fichero de configuración SSHD

```
sudo /etc/init.d/ssh restart
```

6.1.1.4 INSTALACIÓN DEL FIREWALL

Por defecto Ubuntu contiene UFW. UFW deniega todas las conexiones entrantes y permite todas las conexiones salientes, sin embargo por defecto está desactivado.

En primer lugar nos aseguramos que soportara IPv6. Para ello en el archivo de configuración ubicado en /etc/default/ufw cambiamos el campo IPV6 a yes.

```
sudo vi /etc/default/ufw
```

```
# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
# accepted). You will need to 'disable' and then 'enable' the firewall for
# the changes to take affect.
IPV6=yes
```

Figura 93 - Fichero de configuración UFW

A continuación pasmos a configurar los puertos.

Dado que la API REST funciona sobre el puerto 8081 permitiremos las conexiones a ese puerto. Además del puerto 80 y 443 para conexiones HTTP y HTTPS, el puerto 9200 para Elastic Search y el puerto 5601 para Kibana (Accesibles solo mediante el servidor de pruebas o el ordenador del desarrollador). Además del puerto 5555 que utilizaremos para conexiones SSH (accesible sólo desde el servidor de pruebas y ordenador del desarrollador) y el puerto 5666 para conexiones NRPE de nagios.

```
sudo ufw allow 5555
sudo ufw allow 8081
sudo ufw allow 80
sudo ufw allow 443
sudo ufw allow 9200 to 51.255.202.84
sudo ufw allow 5601 to 51.255.202.84
sudo ufw allow 5555 to 51.255.202.84
```

```
sudo ufw disable  
sudo ufw enable
```

To	Action	From
--	-----	-----
8081	ALLOW	Anywhere
80	ALLOW	Anywhere
443	ALLOW	Anywhere
5555	ALLOW	88.10.35.53
9200	ALLOW	88.10.35.53
5601	ALLOW	88.10.35.53
5555	ALLOW	51.255.202.84
9200	ALLOW	51.255.202.84
5601	ALLOW	51.255.202.84
8081 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)

Figura 94 - Reglas Firewall servidor deploy

De esta manera solo es posible acceder por SSH al servidor o a kibana/elasticSearch mediante el túnel del servidor de pruebas o el ordenador del administrador del sistema.

6.1.1.5 INSTALACIÓN DE LOGWATCH

Logwatch permite monitorizar los logs del sistema y enviar información relacionada al rendimiento de los mismos mediante email al administrador del servidor cronológicamente. Se instaló y añadió a una tarea cron para que el administrador esté informado en todo momento del estado de los logs

```
Sudo apt-get install logwatch
vim /etc/cron.daily/00logwatch
sudo /usr/sbin/logwatch --output mail --mailto
jdecastrocabello@gmail.com --detail high
```

----- httpd Begin -----

```
9.09 MB transferred in 498 responses (1xx 0, 2xx 442, 3xx 13, 4xx 43, 5xx 0)
 153 Images (3.14 MB),
 260 Content pages (3.24 MB),
 22 Fonts (2.42 MB),
 63 Other (0.29 MB)
```

Requests with error response codes

```
404 Not Found
  /robots.txt: 10 Time(s)
  http://check.proxyradar.com/azenv.php?auth... 1535545168&p=80: 3 Time(s)
  /manual: 2 Time(s)
  /!: 1 Time(s)
  /about: 1 Time(s)
  /about-us: 1 Time(s)
  /about.html: 1 Time(s)
  /about_us: 1 Time(s)
  /aboutus: 1 Time(s)
  /blog/: 1 Time(s)
  /blogs/: 1 Time(s)
  /company: 1 Time(s)
  /contact: 1 Time(s)
  /contact-us: 1 Time(s)
  /contact.html: 1 Time(s)
  /contact_us: 1 Time(s)
  /contactus: 1 Time(s)
  /drupal: 1 Time(s)
  /help: 1 Time(s)
  /joom!: 1 Time(s)
  /joomla!: 1 Time(s)
  /new/: 1 Time(s)
  /old/: 1 Time(s)
  /portal/: 1 Time(s)
  /shop/: 1 Time(s)
  /site/: 1 Time(s)
  /support: 1 Time(s)
  /test/: 1 Time(s)
  /wordpress/: 1 Time(s)
  /wp/: 1 Time(s)
405 Method Not Allowed
  /webdav: 1 Time(s)
```

A total of 4 ROBOTS were logged

```
Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html) 3 Time(s)
- 1 Time(s)
Mozilla/5.0 (compatible; MJ12bot/v1.4.5; http://www.majestic12.co.uk/bot.php?+) 4 Time(s)
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:28.0) Gecko/20100101 Firefox/28.0 (FlipboardProxy/1.1; +http://flipboard.com/browserproxy) 2 Time(s)
```

----- httpd End -----

Figura 95 - Extracto de correo logwatch

6.1.2 INSTALACIÓN DE LA JVM (JAVA VIRTUAL MACHINE)

Fué necesario instalar la Java Virtual Machine ya que es la plataforma de programación utilizada en la programación de la API REST y el modelo además de ser la base de kettle y elastic Search.

```
sudo apt-get install openjdk-8-jdk
```

6.1.3 INSTALACIÓN DE GIT (CONTROL DE VERSIONES)

Toda la aplicación se ha desarrollado utilizando los servidores de Bitbucket y utilizando como control de versiones Git. Para poder desplegar la aplicación en el servidor fue necesario instalar el cliente git en el servidor y descargar el proyecto.

```
sudo apt-get update
```

```
sudo apt-get install git
```

Para configurar el usuario git del servidor se efectuó de la siguiente manera:

```
Sudo git config --global user.name "infojapon"
```

```
Sudo git config --global user.email jdecastrocabello@gmail.com
```

6.1.4 INSTALACIÓN DE APACHE MAVEN

Apache Maven ha sido utilizado junto a spring-boot en el proyecto para:

- Validación del proyecto
- Compilación
- Empaquetamiento
- Pruebas de integración
- Descarga y gestión de dependencias
- Despliegue de proyecto

```
Sudo apt-get install maven
```

Se ejecuta cada vez que se inicia el servicio backend.

6.1.5 CONFIGURACIONES PREVIAS A LA EJECUCIÓN DEL SERVICIO TOMCAT DE LA API REST

Es importante remarcar que el servidor contratado corre en un contenedor virtual. En estos entornos, los contenedores tienen que compartir entropía entre ellos y el host padre que los contiene. En Linux la JVM tiene por defecto leer del dispositivo /dev/random que es un generador de números aleatorios con fines de securización y criptográfica bloqueante. La fuente de entropía usa este generador de números así que cuando el dispositivo /dev/random es leído la fuente de entropía es drenada. Si la fuente de entropía está vacía /dev/random se bloquea.

Al ser contenedores que “solitarios” (no poseen teclado, ni mouse) tienen tasas bajas adquiriendo entropía. Esto provoca que, como la JVM lee de /dev/random, se generen cuellos

de botella ya que Apache Tomcat utiliza sub-sistemas de seguridad usando una API para generar números aleatorios que utiliza directamente el dispositivo /dev/random drenándolo y generando un rendimiento negativo.

Para solventar este problema existe un paquete que fuerza a los subsistemas de seguridad de Apache Tomcat a utilizar otra fuente (/dev/urandom) eliminando así los cuellos de botella. [31]

```
Apt-get install haveged -y
```

6.1.6 INSTALACIÓN DE APACHE2 + PHP 7.0

Para instalar el servidor web apache2 que contendrá la aplicación cliente ejecutamos el siguiente comando:

```
sudo apt-get install apache2
```

Una vez instalado apache 2 cambiamos la configuración para, como bien especifiqué anteriormente, poder acceder a los recursos web sin necesidad de poner la terminación (formato):

```
sudo vi /etc/apache2/apache2.conf
```

En la linea 166: AllowOverride None lo cambiamos por AllowOverride All

A continuación ejecutaremos los siguientes comandos para habilitar el módulo de sustitución del formato y reiniciamos el servicio apache2 para que los cambios surtieran efecto.

```
sudo a2enmod rewrite
```

```
sudo /etc/init.d/apache2 restart
```

Para la instalación de PHP 7.0 (Necesario para algunos módulos de apache2) introdujimos el siguiente código que instalaba también las librerías necesarias para sus módulos aplicado en apache2:

```
sudo apt-get install php7.0 libapache2-mod-php7.0 php7.0-mcrypt
```

6.2 DESPLIEGUE BACKEND

Para desplegar la aplicación backend que contiene tanto el modelo como el controlador RESTful fue necesario conectar con el servidor de control de versiones Bitbucket y descargar el programa mediante el comando git [32].

Para la primera descarga se ejecutó:

```
sudo git clone https://jdecastroc@bitbucket.org/jdecastroc/punto-japon.git
```

Y para el primer despliegue, en la carpeta raíz de la aplicación descargada:

```
sudo mvn clean package
```

```
sudo mvn spring-boot:run
```

6.2.1 CONFIGURACIÓN PARA LA CORRECTA ACTUALIZACIÓN DEL REPOSITORIO DE ARTÍCULOS

Para que los artículos se actualicen cada hora añadimos una tarea crontab que llamara a la API para actualizar el repositorio.

```
crontab -e
```

Y añadimos al final

```
0 * * * * curl -X POST  
http://www.infojapon.com:8081/articulos/actualizarRepositorio
```

A continuación para que la transformación se hiciera correctamente fue necesario cambiar la localización del servidor, por defecto inglesa, a española. Sobre todo para evitar problemas con el formato de las fechas que se van a leer de los RSS. Para ello cambiamos en /etc/default/locale y pusimos:

```
LC_ALL=es_ES.UTF-8
```

```
LANG=es_ES.UTF-8
```

A continuación instalamos el paquete de idioma español con apt y se fijó en /src/main/resources/indexBlogs.sh la nueva IP.

6.3 DESPLIEGUE FRONTEND (Cliente web)

En las versiones desarrolladas el cliente web se encuentra dentro del mismo proyecto que el backend concretamente en la ruta /src/main/web.

Primero de todo necesitamos configurar un servidor virtual en el servicio web apache2 para poder acceder a la aplicación. En nuestro caso creamos un nuevo servidor virtual ubicado en /etc/apache2/mod-available/infojapon.conf con la siguiente información.

```
<VirtualHost *:80>

    ServerName www.infojapon.com
    ServerAlias infojapon
    ServerAdmin jdecastrocabello@gmail.com
    DocumentRoot /var/www/infojapon
    <Directory /var/www/infojapon>
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    <Directory /var/www/>
        Options -Indexes
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

De esta manera cualquier conexión al puerto HTTP 80 redireccionaría al cliente a la carpeta /var/www/infojapon donde se encontraría la aplicación web cliente. Las opciones embebidas entre las etiquetas <Directory> son opciones que permiten el cambio de los formatos de las páginas y el acceso sin restricciones a los usuarios finales en ese directorio.

La configuración de los log nos permitirá hacer un seguimiento de los errores del servicio para una mejor prevención y anticipación.

Para desplegar y activar el cliente web copiaremos el contenido del cliente web ubicado en el proyecto en la carpeta que se va a servir a través del puerto 80 (/var/www/infojapon).

```
sudo mkdir /var/www/infojapon
sudo cp -R /home/deploy/punto-japon/src/web/Puntojapon/*
/var/www/infojapon/
sudo a2ensite infoJapon.conf
```

6.4 GOOGLE PLAY (Cliente app)

Para poder subir la aplicación móvil al mercado de aplicaciones Google Play ha sido necesario el pago de una licencia de desarrollador Google. Una vez efectuado el pago fue necesario llenar una serie de formularios referentes a la naturaleza de la aplicación así como subir un conjunto de imágenes que la identificarían más adelante. Por último paso se subió mediante el panel de desarrollador la última .apk generada con Intel XDK.

Capítulo 7. ANÁLISIS DE MÉTRICAS

7.1.1 MÉTRICAS DE APLICACIÓN

Mediante la inclusión de un fichero de registros propios (app.log) la aplicación backend es capaz de registrar todos los accesos a los recursos así como los errores en los diferentes procesos. Este fichero es un fichero log persistente que almacena los registros en tiempo real en ElasticSearch gracias al uso de Logstash [33] [34].

```
input {
  file {
    path => "/home/deploy/punto-japon-deploy/app.log"
    start_position => "beginning"
    type => "logs"
  }
}
filter {
  grok{
    match=>{
      "message" => "\A\[{\LOGLEVEL:loglevel}\] \{TIMESTAMP_ISO8601:timestamp\} com\\.puntojapon\\.common\\.appLogger logInfo \- \(\{IP:clientip\}\) \{NOTSPACE:request\}\{GREEDYDATA:additionalOutput\}"
    }
  }
}
output {
  elasticsearch {
    hosts => ["91.134.143.80:9200"]
    index => "logpuntojapon-\{+YYYY.MM.dd\}"
}
```

Figura 96 - Fichero de configuración logStash

Mediante el uso de Kibana se configuró que analizara el registro del fichero app.log almacenado en ElasticSearch. Esta configuración dotó al sistema del control de las siguientes métricas:

- Nivel de alerta de registros
- Fecha de acceso de usuarios
- IP de usuarios
- Operación efectuada por los usuarios
- Información extra proporcionada en las operaciones

De esta manera es posible determinar la demanda de los usuarios tanto móviles como web y actuar en consecuencia a la misma.



Figura 97 - Gráfica de peticiones a la API REST

```
[INFO] 2016-07-04 16:56:27,547 com.puntojapon.common.appLogger logInfo - (83.51.43.209) [JOB_SEARCH_REQUEST] - Prefecture: Kanagawa Speci
[INFO] 2016-07-04 12:10:14,778 com.puntojapon.common.appLogger logInfo - (91.134.143.80) [REPOSITORY_UPDATED]
[INFO] 2016-07-04 12:15:00,132 com.puntojapon.common.appLogger logInfo - (51.255.202.84) [REPOSITORY_UPDATED]
[INFO] 2016-07-04 16:50:41,252 com.puntojapon.common.appLogger logInfo - (83.51.43.209) [LANGUAGE SCHOOL LIST REQUEST] - Area: tokyo
[INFO] 2016-07-05 00:19:53,806 com.puntojapon.common.appLogger logInfo - (83.51.43.209) [ARTICLES_REQUEST]
```

Figura 98 - Ejemplo de registros almacenados por el sistema

7.1.2 GOOGLE ANALYTICS

Google Analytics permite, mediante la inclusión de un script en el código HTML de la página principal, analizar el tráfico de la aplicación web para generar gráficas con los datos obtenidos.

Estas gráficas permiten realizar un análisis de situación del posicionamiento de la aplicación así como determinar de factores entre otros muchos, como:

- Origen de los visitantes
- Duración de la visita
- Número de páginas servidas
- Porcentaje de visitantes que acuden de rebote
- Sesiones

De esta manera se pretende adoptar una posición proactiva que permita tomar decisiones para el crecimiento de la aplicación.

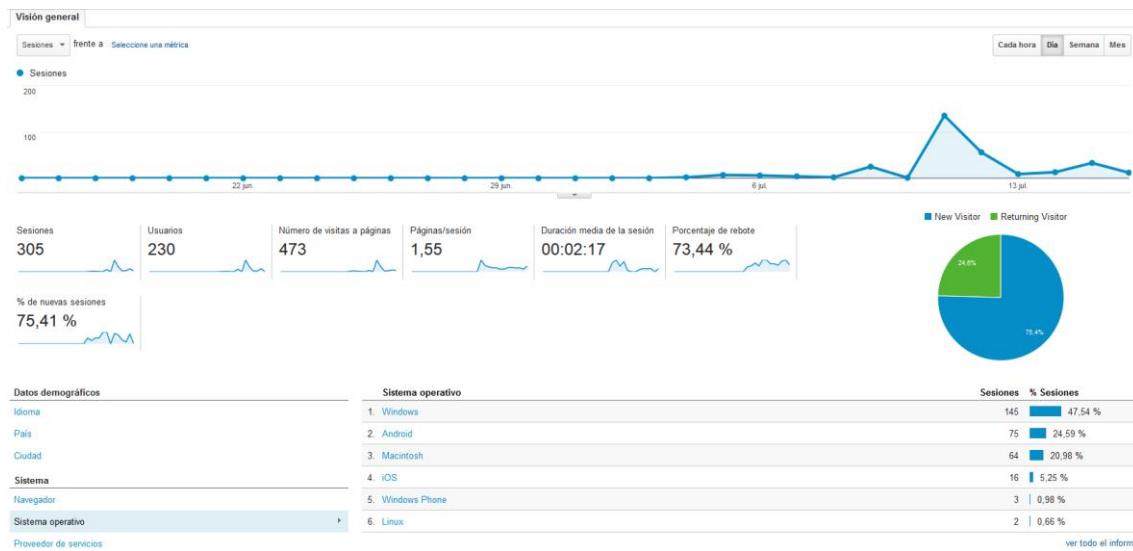


Figura 99 - Panel de control Google Analytics

Capítulo 8. CONCLUSIONES

El proyecto de InfoJapón ha sido un éxito completo a nivel personal y profesional.

Personalmente, el proyecto lo que más me ha aportado a parte de buenas prácticas y constancia es confianza. Como profesional siento que estoy al servicio de la sociedad y el simple hecho de poder haber llevado un proyecto que ayuda a la gente a estar un poco más cerca de donde quiere, desde su inicio hasta su despegue sentando unas bases para futuros desarrollos me ha hecho creer que puedo lograr cualquier cosa pese a las dificultades que se presenten. Han sido duros meses de trabajo y constancia luchando contra el tiempo y contra problemas que a priori no tenían solución pero al final siento que ha sido el broche a mis estudios del grado en ingeniería informática.

En cuanto al apartado profesional siento que estoy más cerca que nunca de mi sueño. El trabajo en este proyecto me ha hecho darme cuenta de lo rápido que avanza este sector y que esa velocidad realmente no significa nada malo, al contrario, significa que podemos elegir nuestro nicho y disfrutar sin dejar de aprender pese al paso de los años. Gracias a InfoJapón he podido aunar mi hobby y mi profesión en pos de ayudar a la gente y a su vez he podido determinar el siguiente paso en mi carrera profesional: llegar a trabajar de desarrollador FullStack. He podido comprobar que el abanico de tecnologías y aplicaciones software utilizadas en el proyecto son demandadas en la actualidad por lo que debo apostar fuerte por seguir en esta línea.

En cuanto a los primeros resultados del proyecto en sí son muy satisfactorios. La primera versión tanto web como móvil fue liberada el 11 de Julio alcanzando el mismo día una suma de 135 sesiones tras publicar la aplicación en dos grupos diferentes de Facebook. La poca gente que lo pudo ver apoyó el anuncio y me animó a continuar con la aplicación dándome las gracias por el buen trabajo. Durante todo el desarrollo del proyecto no esperaba ni felicitaciones, ni agradecimientos, ni ánimos por parte de ningún usuario final más allá de amigos o familiares, el mismo día de la liberación de la aplicación, cuando vi a la gente apoyar el proyecto... esa sensación no se me olvidará nunca y es que tras un duro trabajo el reconocimiento muchas veces significa la victoria.

A día 17 de Julio ha habido 349 sesiones de las cuales un 26,1% ha vuelto en menos de 1 semana, 518 visitas con una duración media de 02:15 minutos, 2221 accesos a los recursos de la API REST y un total de 114 reacciones positivas en Facebook.

The screenshot shows a Facebook post with several comments:

- User 1:** asombroso muchas gracias! sigue así! en verdad esto se ve muy util 😊
Replies:
 - Me gusta · Responder · 11 de julio a las 23:02 · Editado
- User 2:** Excelente trabajo y muy útil. Gracias
Replies:
 - Ya no me gusta · Responder · 11 de julio a las 11:11
 - 1 respuesta
- User 3:** Felicitaciones por la iniciativa!!
Replies:
 - Ya no me gusta · Responder · 11 de julio a las 15:51
 - 1 respuesta
- User 4:** Felicitaciones por este gran proyecto por ayudar a conocer mas y orientar a los que deseamos ir a este gran país un gran trabajo gracias saludos
Replies:
 - Ya no me gusta · Responder · 11 de julio a las 15:55
 - 1 respuesta
- User 5:** Felicitaciones, un gran aporte.
Replies:
 - Me gusta · Responder · 12 de julio a las 1:30 · Editado

Capítulo 9. FUTURAS LÍNEAS DE TRABAJO

El proyecto InfoJapón no ha hecho más que comenzar. InfoJapón es un proyecto ambicioso que pretende abarcar varios ámbitos de información (blogs, videoblogs, estudios, trabajo...). El crecimiento es progresivo y tiene en cuenta las opiniones y demandas de los usuarios.

Los siguientes pasos son:

- Introducción de nuevas fuentes para la búsqueda de empleo.
- Introducción de nuevas fuentes para la búsqueda de vivienda.
- Creación de estadísticas para los colegios de lengua japonesa y un sistema de búsquedas apropiado al mismo ya que es el campo de estudios más demandado por los extranjeros hispanohablantes.
- Creación de un apartado para la visualización de video blogs asociados al proyecto ordenados de forma cronológica. Este apartado permitirá las búsquedas sobre los video blogs.
- Creación de crawlers para la indexación completa de los blogs asociados al proyecto. De esta manera las búsquedas sobre información de Japón contarán con un gran conjunto de datos.
- Creación de un apartado para organización y visualización de encuentros multi-región utilizando la API de meetup.

PRESUPUESTO

Proyecto InfoJapón

Universidad Europea - Grado en Ingeniería Informática

Líder del Proyecto: Jorge de Castro Cabello

Presupuesto total

		Horas empleadas:	501			12.727,60
Nombre	Elemento	Concepto	Tipo de Unidades	Horas/Ud	Tasa/Mes	Presupuesto
1 Estudio previo				16		320,00
Jorge de Castro Cabello	Consultoría	Tasa	Jornadas	16	20,00	320,00
2 Ejecución				475		12.280,00
2.1 Diseño funcional				90		2.250,00
Jorge de Castro Cabello	Analista funcional	Tasa	Jornadas	90	25,00	2.250,00
2.2 Desarrollo				276		6.960,00
Jorge de Castro Cabello	Analista programador	Tasa	Jornadas	264	25,00	6.600,00
Mao Misaki	Diseñadora	Tasa	Jornadas	12	30,00	360,00
2.3 Pruebas de usuario				20		400,00
Alberto García	Tester	Tasa	Jornadas	5	15,00	75,00
Jorge de Castro Cabello	Analista programador	Tasa	Jornadas	10	25,00	250,00
Santiago Gualda	Tester	Tasa	Jornadas	2	15,00	30,00
Pablo Bordons	Tester	Tasa	Jornadas	3	15,00	45,00
2.4 Producción				59		1.770,00
Jorge de Castro Cabello	Administrador de sistemas	Tasa	Jornadas	59	30,00	1.770,00
2.5 Mantenimiento y documentación				30		900,00
Jorge de Castro Cabello	Administrador de sistemas	Tasa	Jornadas	30	30,00	900,00
3 Varios				10		127,60
	Servidor virtual	Materiales	Cantidad	2	7	66,50
	Licencia desarrollador Google	Materiales	Cantidad	1		26,00
	Dominio .com	Materiales	Cantidad	1		14,30
	Dominio .ovh	Materiales	Cantidad	1		2,40
	Plantilla CSS	Materiales	Cantidad	1		18,40

Figura 100 - Presupuesto del proyecto

BIBLIOGRAFÍA.

- [1] Statistics Bureau. Inmigration statistics. [Online].
<http://www.stat.go.jp/english/data/idou/3.htm>
- [2] Canal Estrategia. En 2020 el universo digital se multiplicará por diez. [Online].
<http://www.confirmasistemas.es/es/contenidos/canal-estrategia/en-2020-el-universo-digital-se-multiplicara-por-diez>
- [3] Universidad Nacional de Educación a Distancia. Breve historia de la inteligencia artificial. [Online]. <http://www2.uned.es/pfp-internet-y-educacion/historia.html>
- [4] Seyed M. Mirtaheri, Mustafa Emre Dinçtürk, Salman Hooshmand, Gregor V. Bochmann, and Guy-Vincent Jourdan. A Brief History of Web Crawlers. [Online].
<http://ssrg.eecs.uottawa.ca/docs/CASCON2013.pdf>
- [5] Neil Aggarwal, Anthony Van Geest, and Vijay Girija. Is scraping and crawling to collect data illegal? [Online]. <https://www.quora.com/Is-scraping-and-crawling-to-collect-data-illegal>
- [6] Andrearss. Formatos para intercambio de información. [Online].
<http://www.hipertextual.com/archivo/2014/05/xml-json-yaml/>
- [7] Wikipedia. Extract, transform and load. [Online].
https://es.wikipedia.org/wiki/Extract,_transform_and_load
- [8] Pentaho Community. Pentaho Data Integration tool. [Online].
<http://community.pentaho.com/projects/data-integration/>
- [9] Wikipedia. Bases de datos documentales. [Online].
https://es.wikipedia.org/wiki/NoSQL#Bases_de_datos_documentales
- [10] Elastic. Elasticsearch. [Online]. <https://www.elastic.co/products/elasticsearch>
- [11] Elastic. Kibana. [Online]. <https://www.elastic.co/products/kibana>
- [12] Elastic. Logstash. [Online]. <https://www.elastic.co/products/logstash>
- [13] Wikipedia. Representational State Transfer. [Online].
https://es.wikipedia.org/wiki/Representational_State_Transfer
- [14] Asier Marqués. Conceptos sobre apis rest. [Online].
<http://asiermarques.com/2013/conceptos-sobre-apis-rest/>
- [15] Wikipedia. Rich Internet Application. [Online].
https://es.wikipedia.org/wiki/Rich_Internet_application

-
- [16] Wikipedia. AJAX. [Online]. <https://es.wikipedia.org/wiki/AJAX>
- [17] Ravi Kiran Perumalla. Ruby on rails or Spring for a startup. [Online].
<https://www.quora.com/Ruby-on-rails-or-Spring-for-a-startup>
- [18] Elastic. Indexing a document. [Online].
<https://www.elastic.co/guide/en/elasticsearch/guide/current/index-doc.html>
- [19] Elastic. Querying a Nested Object. [Online].
<https://www.elastic.co/guide/en/elasticsearch/guide/current/nested-query.html>
- [20] Siteworx. Chapter 6: Object and Nested Types, and Multifields | Dev Focus: Elasticsearch (Tutorial / Demo). [Online]. <https://www.youtube.com/watch?v=yqAKfwGZpE0>
- [21] Elastic. Inner hits. [Online].
<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-inner-hits.html>
- [22] Mdewit. Inner hits not working with nested filter. [Online].
<http://stackoverflow.com/questions/29251994/inner-hits-not-working-with-nested-filter>
- [23] Oracle. Process Builder. [Online].
<https://docs.oracle.com/javase/7/docs/api/java/lang/ProcessBuilder.html>
- [24] Jsoup. jsoup 1.9.2 API. [Online]. <https://jsoup.org/apidocs/>
- [25] Japan Study Support. Information for foreign students. [Online]. <http://www.jpss.jp/>
- [26] Wikipedia. Divide y vencerás. [Online].
https://es.wikipedia.org/wiki/Algoritmo_divide_y_vencer%C3%A1s
- [27] Realestate. Find a home in Japan. [Online]. <https://www.realestate.co.jp/>
- [28] Jon Brisbin, Oliver Gierke, and Greg Turnquist. Spring Data REST - Reference Documentation. [Online]. <http://docs.spring.io/spring-data/rest/docs/current/reference/html/>
- [29] apsillers. How does access control allow origin header work. [Online].
<http://stackoverflow.com/questions/10636611/how-does-access-control-allow-origin-header-work>
- [30] Mitchell Anicas. How to use apache Jmeter to perform load testing on a web server. [Online]. <https://www.digitalocean.com/community/tutorials/how-to-use-apache-jmeter-to-perform-load-testing-on-a-web-server>
- [31] Henry. Slow startup on Tomcat 7.0.57 because of SecureRandom. [Online].
<http://stackoverflow.com/questions/28201794/slow-startup-on-tomcat-7-0-57-because->

[of-securerandom](#)

[32] Tobias Günther. Git cheat sheet. [Online]. <https://www.git-tower.com/blog/git-cheat-sheet/>

[33] Elastic. Logstash configuration examples. [Online].
<https://www.elastic.co/guide/en/logstash/current/config-examples.html>

[34] Robert A. Heinlein. Grok Debugger. [Online]. <http://grokdebug.herokuapp.com/>