



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS

Tema:

DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN EN
PLATAFORMA MÓVIL PARA FACILITAR EL PROCESAMIENTO DE
BASES DE DATOS DE PUBLICACIONES CIENTÍFICAS.

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

LÍNEA DE INVESTIGACIÓN

Desarrollo de Software

AUTOR: Daniel Omar Robayo Orozco

TUTOR: Ing. PhD. Félix Oscar Fernández Peña, MSc.

Ambato – Ecuador

Marzo, 2021

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: “DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN EN PLATAFORMA MÓVIL PARA FACILITAR EL PROCESAMIENTO DE BASES DE DATOS DE PUBLICACIONES CIENTÍFICAS”, desarrollado bajo la modalidad Proyecto de Investigación por el señor Daniel Omar Robayo Orozco, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2021

Ing. PhD. Félix Oscar Fernández Peña, MSc.

TUTOR

AUTORÍA

El presente trabajo de investigación titulado: “DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN EN PLATAFORMA MÓVIL PARA FACILITAR EL PROCESAMIENTO DE BASES DE DATOS DE PUBLICACIONES CIENTÍFICAS.” es absolutamente original, auténtico y personal. En tal virtud, el contenido, a efectos legales y académicos que se despenden del mismo, son de exclusiva responsabilidad del autor.

Ambato, marzo de 2021

Daniel Omar Robayo Orozco

CC: 0503787293

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo de 2021.

Daniel Omar Robayo Orozco

CC: 0503787293

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Daniel Omar Robayo Orozco, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, “DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN EN PLATAFORMA MÓVIL PARA FACILITAR EL PROCESAMIENTO DE BASES DE DATOS DE PUBLICACIONES CIENTÍFICAS”, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, marzo de 2021

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL

Dr. PhD. Víctor Guachimosa
PROFESOR CALIFICADOR

Dr. PhD. Julio Balarezo
PROFESOR CALIFICADOR

DEDICATORIA

A mi madre que cuando he pensado en dejarlo todo me ha llenado de valentía, motivación y esperanza en continuar y no rendirme para luchar.

A mi padre que me ha enseñado la valía en trabajar por conseguir algo y no parar; pero, sobre todo, a ver el lado positivo de la vida.

A mi hermano el cual me ha enseñado que indirectamente me convertí en una figura a seguir y he tenido que llenar huecos que una vez en mí quedaron; obteniendo aciertos y fracasos en el intento, pero sintiéndome feliz al ver en las formas en las cuales te has adaptado y las has resuelto.

A aquellos que ya no están dejando conocimiento y lecciones ayudándome a resolver gran parte de las dudas que he adquirido con el paso del tiempo.

AGRADECIMIENTO

A todos aquellos que han estado en las buenas, las malas y cuando más los he necesitado; sin tener que mencionarlos sabrán que este mensaje es para ustedes en cuando lean este fragmento.

A mi tutor. Gracias a usted y a sus palabras de aliento entendí la importancia de la confianza en uno mismo, luchando así por continuar con aquello que me apasiona.

Pero, especialmente agradezco a mis padres demostrándome que mientras se tenga aspiraciones y la voluntad de progresar se podrá conseguirlo con o sin apoyo y; cada cual, a su manera, siempre pensar en la familia.

NADA ES VERDAD, TODO ESTÁ PERMITIDO.

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN EJECUTIVO	xviii
ABSTRACT	xix
INTRODUCCIÓN	xx
CAPÍTULO I.- MARCO TEÓRICO	1
1.1 Tema de investigación	1
1.2 Antecedentes investigativos	1
1.2.1 Contextualización del problema	1
1.2.2 Fundamentación teórica	2
1.2.2.1 Sistema de recomendación (SR)	2
1.2.2.2 Análisis de metodologías de desarrollo de software	3
1.2.2.3 Análisis de tecnologías de desarrollo móvil	10
1.2.2.4 Análisis de lenguajes para el análisis de datos	16
1.2.2.5 Definición de arquitecturas	18
1.2.2.6 El estado del ecosistema de desarrolladores	20
1.2.2.7 Web Scraping	21
1.2.2.8 API-REST	21
1.2.2.9 Procesamiento de Bases de Datos de Bibliotecas Científicas	24
1.2.2.10 Escala de usabilidad del sistema	39
1.3 Objetivos	40
1.3.1 General	40
1.3.2 Específicos	40
CAPÍTULO II.- METODOLOGÍA	41
2.1 Materiales	41
2.2 Métodos	42
2.2.1 Modalidad básica de la investigación	42
2.2.2 Metodología seleccionada	42

2.2.3 Framework seleccionado	42
2.2.4 Lenguaje seleccionado para el análisis de datos	42
2.2.5 Metodología de desarrollo del proyecto	43
2.2.6 Población y muestra	44
2.2.7 Uso de SUS para evaluar el impacto del sistema de recomendación.....	44
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN	45
3.1 Análisis y discusión de los resultados	45
3.1.1 Aplicación de metodología Mobile-D.....	45
3.1.2 Fase de exploración.....	45
3.1.2.1 Levantamiento de requerimientos.....	45
3.1.2.2 Establecer grupo de interés.....	46
3.1.2.3 Planificación del proyecto.....	46
3.1.2.4 Búsqueda de bibliotecas virtuales que ofrezcan APIs de consulta de artículos científicos.....	47
3.1.2.5 Identificación de páginas habilitadas para realizar web scraping.....	48
3.1.3 Fase de iniciación	52
3.1.3.1 Capacitación de personal en el uso de las APIs encontradas.....	52
3.1.3.2 Optimización en el envío de consultas a las APIs obteniendo únicamente la información necesaria.....	55
3.1.3.3 Establecer los datos a utilizar en las páginas web a realizar web scraping.....	57
3.1.3.4 Creación del servicio de consulta.....	59
3.1.3.5 Planificación de la arquitectura	71
3.1.3.6 Diagramas de actividades del proyecto	72
3.1.3.7 Diagramas de secuencias del proyecto.....	79
3.1.3.8 Configuración de proyecto móvil con paquetes necesarios	84
3.1.4 Fase de producción	85
3.1.4.1 Iteración 1.- Navigation Bar.....	85
3.1.4.2 Iteración 2.- Búsqueda de artículos	89
3.1.4.3 Iteración 3.- Búsqueda de revistas	98
3.1.4.4 Iteración 4.- Configuración de parámetros	107
3.1.4.5 Iteración 5.- Consulta de revista.....	115

3.1.4.6 Iteración 6.- Búsqueda de artículos por issn o eissn .	121
3.1.5 Fase de estabilización	127
3.1.6 Fase de prueba y reparación.....	128
3.1.7 Recolección de información	129
3.1.8 Procesamiento y análisis de datos	132
3.1.9 Muestra del funcionamiento de la aplicación	137
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES	141
4.1 Conclusiones.....	141
4.2 Recomendaciones	142
BIBLIOGRAFÍA	143
ANEXOS	147
Anexo 1.- Código HTML Cuartiles	147
Anexo 2.- Código HTML SJR.....	147
Anexo 3.- Código HTML Citaciones por documento	147
Anexo 4.- Código HTML Total citas	148
Anexo 5.- Código HTML Citas externas por documento	149
Anexo 6.- Código HTML Documentos citables	149
Anexo 7.- Código HTML Documentos citados	150

ÍNDICE DE FIGURAS

Figura 1.1: Ciclo de desarrollo Extreme Programming (XP)	7
Figura 1.2: Procesos de fases de Extreme Programming (XP)	8
Figura 1.3: Ciclo de desarrollo Mobile-D	9
Figura 1.4: Código representativo de React Native	13
Figura 1.5: Ejecución de proyecto en motor Javascript	13
Figura 1.6: Comunicación puente nativo	14
Figura 1.7: Conexión puente nativo con dispositivo	14
Figura 1.8: Arquitectura React Native	15
Figura 1.9: Arquitectura Flutter	16
Figura 1.10: Arquitectura del aplicativo móvil	18
Figura 1.11: Arquitectura del aplicativo de análisis de datos	19
Figura 1.12: Arquitectura de API-REST	19
Figura 1.13: Resultado de frameworks móviles multiplataforma utilizados	20
Figura 3.1: Código para realizar web scraping	48
Figura 3.2: Web scraping- HTTP Error 403: Forbidden	49
Figura 3.3: Web scraping- HTTP Error 401: Unauthorized	50
Figura 3.4: Web scraping- Datos dinámicos	51
Figura 3.5: Web scraping- Datos correctos	51
Figura 3.6: Obtención de api key en Springer	52
Figura 3.7: Consulta API Springer- Open access	53
Figura 3.8: Consulta API Springer- Metadata v2	53
Figura 3.9: Obtención de api key en Elsevier	54
Figura 3.10: Consulta API ScienceDirect	54
Figura 3.11: Consulta API Scopus	55
Figura 3.12: Consulta API Scopus (Optimizada)	56
Figura 3.13: Consulta publisher de API Autor	56
Figura 3.14: Búsqueda de parámetro enviado desde el url	57
Figura 3.15: Campos requeridos para uso en el aplicativo	58
Figura 3.16: Instalación de BeautifulSoup	59
Figura 3.17: Instalación de Flask	59
Figura 3.18: Modelo de artículos	61
Figura 3.19: Modelo de revistas	62

Figura 3.20: Método para consultar revistas en Elsevier	63
Figura 3.21: Método para consultar artículos en Elsevier	63
Figura 3.22: Método para consultar artículos por issn en Elsevier	63
Figura 3.23: Método para consultar revistas en Springer	63
Figura 3.24: Método para consultar artículos en Springer.....	64
Figura 3.25: Método para consultar artículos por issn en Springer	64
Figura 3.26: Formato JSON de Cuartil.....	64
Figura 3.27: Formato JSON de Citaciones por documento	65
Figura 3.28: Formato JSON de Citas totales	65
Figura 3.29: Formato JSON de Citas externas por documento	65
Figura 3.30: Formato JSON de Documentos citables	65
Figura 3.31: Formato JSON de Documentos citados	65
Figura 3.32: Formato JSON de SJR.....	66
Figura 3.33: Modelo dos columnas	66
Figura 3.34: Modelo tres columnas.....	66
Figura 3.35: Web scraping Scimago Journal & Country Rank.....	66
Figura 3.36: Vista- Búsqueda de revistas.....	68
Figura 3.37: Vista- Búsqueda de artículos	69
Figura 3.38: Vista- Búsqueda de artículos por issn en Elsevier	70
Figura 3.39: Vista- Búsqueda de artículos por issn en Springer.....	70
Figura 3.40: Vista- Web scraping	71
Figura 3.41: Arquitectura del proyecto	71
Figura 3.42: Diagrama de actividad.- Buscar artículos en inicio de pestaña	72
Figura 3.43: Diagrama de actividad.- Buscar artículos al final de la lista.....	72
Figura 3.44: Diagrama de actividad.- Buscar artículos al final de la lista.....	73
Figura 3.45: Diagrama de actividad.- Redirigir sitio web del artículo	73
Figura 3.46: Diagrama de actividad.- Buscar revistas en el inicio de pestaña	74
Figura 3.47: Diagrama de actividad.- Buscar revistas con cambio de término por usuario	74
Figura 3.48: Diagrama de actividad.- Consultar datos de revista	75
Figura 3.49: Diagrama de actividad.- Buscar artículos por revista	76
Figura 3.50: Diagrama de actividad.- Cambiar término de búsqueda	77
Figura 3.51: Diagrama de actividad.- Cambiar número de artículos a buscar y agrupar	78

Figura 3.52: Diagrama de actividad.- Cambiar área de búsqueda de biblioteca virtual	78
Figura 3.53: Diagrama de secuencia.- Inicio pestaña Artículos.....	79
Figura 3.54: Diagrama de secuencia.- Consulta de artículos con parámetros del usuario	79
Figura 3.55: Diagrama de secuencia.- Scroll al final de la lista	80
Figura 3.56: Diagrama de secuencia.- Redirigir a biblioteca virtual con el artículo	80
Figura 3.57: Diagrama de secuencia.- Inicio pestaña Revistas	81
Figura 3.58: Diagrama de secuencia.- Consulta de revistas con parámetros del usuario	81
Figura 3.59: Diagrama de secuencia.- Consultar por revista seleccionada	82
Figura 3.60: Diagrama de secuencia.- Consulta de artículos por revista.....	82
Figura 3.61: Diagrama de secuencia.- Cambiar término de búsqueda	83
Figura 3.62: Diagrama de secuencia.- Cambiar número de artículos a buscar y agrupar	83
Figura 3.63: Diagrama de secuencia.- Cambiar área de búsqueda de biblioteca virtual	84
Figura 3.64: Instalación de paquetes necesarios para el proyecto.....	84
Figura 3.65: Iteración 1- Trabajo(home_page)	88
Figura 3.66: Iteración 1- Liberación (Home Page)	88
Figura 3.67: Iteración 2- Trabajo (home_page)	92
Figura 3.68: Iteración 2- Trabajo (articulos_page)	93
Figura 3.69: Iteración 2- Liberación (carga skeleton)	94
Figura 3.70: Iteración 2- Liberación (carga imágenes bibliotecas virtuales)	95
Figura 3.71: Iteración 2- Liberación (búsqueda realizada con campo de texto visible)	95
Figura 3.72: Iteración 2- Liberación (búsqueda realizada con campo de texto oculto)	96
Figura 3.73: Iteración 2- Liberación (cargando siguientes artículos con scroll)	96
Figura 3.74: Iteración 2- Liberación (búsqueda sin resultados)	97
Figura 3.75: Iteración 2- Liberación (redirección al presionar artículo)	97
Figura 3.76: Iteración 3- Trabajo (home_page)	102
Figura 3.77: Iteración 3- Trabajo (revistas_page)	103
Figura 3.78: Iteración 3- Liberación (carga skeleton)	104
Figura 3.79: Iteración 3- Liberación (carga imágenes bibliotecas virtuales).....	105

Figura 3.80: Iteración 3- Liberación (búsqueda realizada con campo de texto visible)	105
Figura 3.81: Iteración 3- Liberación (búsqueda realizada con campo de texto oculto)	106
Figura 3.82: Iteración 3- Liberación (búsqueda sin resultados)	106
Figura 3.83: Iteración 4- Trabajo (parametros_page)	111
Figura 3.84: Iteración 4- Liberación (Parámetros).....	111
Figura 3.85: Iteración 4- Liberación (Cambio de término por defecto).....	112
Figura 3.86: Iteración 4- Liberación (Mensaje al actualizar el término de búsqueda).....	112
Figura 3.87: Iteración 4- Liberación (Opciones de número de artículos).....	113
Figura 3.88: Iteración 4- Liberación (Mensaje de actualización en el número de artículos).....	113
Figura 3.89: Iteración 4- Liberación (Opciones de área de búsqueda en Elsevier)	114
Figura 3.90: Iteración 4- Liberación (Opciones de área de búsqueda en Springer)	114
Figura 3.91: Iteración 4- Liberación (Mensaje de actualización en el área de búsqueda).....	115
Figura 3.92: Iteración 5- Trabajo (revista_page)	119
Figura 3.93: Iteración 5- Liberación (Carga de skeleton)	119
Figura 3.94: Iteración 5- Liberación (Respuesta con resultados).....	120
Figura 3.95: Iteración 5- Liberación (Respuesta sin resultados).....	120
Figura 3.96: Iteración 6- Trabajo (articulos_issn_page).....	124
Figura 3.97: Iteración 6- Liberación (carga skeleton)	125
Figura 3.98: Iteración 6- Liberación (búsqueda realizada)	126
Figura 3.99: Iteración 6- Liberación (cargando siguientes artículos con scroll)	126
Figura 3.100: Iteración 6- Liberación (redirección al presionar artículo)	127
Figura 3.101: Aplicación móvil- Ruta general	127
Figura 3.102: Establecimiento de dirección en flask.....	127
Figura 3.103: Debounce en intercambio de pestañas.....	128
Figura 3.104: Pregunta 1	129
Figura 3.105: Pregunta 2	129
Figura 3.106: Pregunta 3	129
Figura 3.107: Pregunta 4	130

Figura 3.108: Pregunta 5	130
Figura 3.109: Pregunta 6	130
Figura 3.110: Pregunta 7	131
Figura 3.111: Pregunta 8	131
Figura 3.112: Pregunta 9	131
Figura 3.113: Pregunta 10.....	132
Figura 3.114: Cálculo de SUS en encuesta 1	132
Figura 3.115: Cálculo de SUS en encuesta 2	133
Figura 3.116: Cálculo de SUS en encuesta 3	133
Figura 3.117: Cálculo de SUS en encuesta 4	133
Figura 3.118: Cálculo de SUS en encuesta 5	134
Figura 3.119: Cálculo de SUS en encuesta 6	134
Figura 3.120: Cálculo de SUS en encuesta 7	134
Figura 3.121: Cálculo de SUS en encuesta 8	135
Figura 3.122: Cálculo de SUS en encuesta 9	135
Figura 3.123: Cálculo de SUS en encuesta 10.....	135
Figura 3.124: Promedio de cálculo de SUS con encuestas recolectadas	136
Figura 3.125: Funcionamiento de la aplicación.- Parámetros de búsqueda	137
Figura 3.126: Funcionamiento de la aplicación.- Búsqueda de artículos	138
Figura 3.127: Funcionamiento de la aplicación.- Selección de un artículo determinado.....	138
Figura 3.128: Funcionamiento de la aplicación.- Búsqueda de revistas.....	139
Figura 3.129: Funcionamiento de la aplicación.- Información relevante según revista seleccionada	139
Figura 3.130: Funcionamiento de la aplicación.- Artículos de revista seleccionada.....	140
Figura 3.131: Funcionamiento de la aplicación.- Selección de un artículo determinado.....	140
Figura 6.1: Código HTML Cuartiles	147
Figura 6.2: Código HTML SJR.....	147
Figura 6.3: Código HTML Citaciones por documento	148
Figura 6.4: Código HTML Total citas	148
Figura 6.5: Código HTML Citas externas por documento	149
Figura 6.6: Código HTML Documentos citables.....	149
Figura 6.7: Código HTML Documentos citados	150

ÍNDICE DE TABLAS

Tabla 1.1: Diferencias entre metodología tradicional y ágil	4
Tabla 1.2: Prácticas básicas de XP	6
Tabla 1.3: Análisis comparativo entre las metodologías Extreme Programming (XP) y Mobile-D	9
Tabla 1.4: Características de funcionalidades en dispositivos móviles.....	12
Tabla 1.5: Aspectos para realizar una aplicación móvil.....	12
Tabla 1.6: API-Springer: Colecciones válidas y tipos de respuesta	25
Tabla 1.7: API-Springer: Métodos abreviados	26
Tabla 1.8: API-Springer: Parámetros en la cadena de consulta	27
Tabla 1.9: API-Springer: Restricciones de búsqueda	29
Tabla 1.10: API-Elsevier: Métodos Scopus API	33
Tabla 1.11: API-Elsevier: Vistas de búsqueda de Scopus	36
Tabla 1.12: API-Elsevier: Métodos Autor API	37
Tabla 1.13: API-Elsevier: Vistas de búsqueda de Autor	38
Tabla 2.1: Muestra.....	44
Tabla 3.1: Definición de usuario.....	46
Tabla 3.2: Tabla de resultados de datos buscados utilizando web scraping	59
Tabla 3.3: Iteración 1- Redirigir a pestañas.....	85
Tabla 3.4: Iteración 1- Colocar título según opción seleccionada.....	85
Tabla 3.5: Iteración 1- Establecer opción de búsqueda para ser usada por las otras pestañas	85
Tabla 3.6: Iteración 2- Mostrar artículos según término de interés	89
Tabla 3.7: Iteración 2- Cargar texto por defecto almacenado	89
Tabla 3.8: Iteración 2- Cargar áreas de búsqueda almacenadas	89
Tabla 3.9: Iteración 2- Establecer término de interés mediante campo de texto	90
Tabla 3.10: Iteración 2- Realizar consulta por API-REST a servicio integrador de bases de datos	90
Tabla 3.11: Iteración 2- Mostrar skeleton	90
Tabla 3.12: Iteración 2- El servicio responde y no existe resultados.....	91
Tabla 3.13: Iteración 2- El servicio responde y existe resultados	91
Tabla 3.14: Iteración 2- Mostrar listado final al usuario	91
Tabla 3.15: Iteración 2- Consultar artículo científico	92

Tabla 3.16: Iteración 2- Cargar los siguientes artículos	92
Tabla 3.17: Iteración 3- Mostrar revistas según término de interés	98
Tabla 3.18: Iteración 3- Cargar texto por defecto almacenado	98
Tabla 3.19: Iteración 3- Cargar número de artículos almacenado	98
Tabla 3.20: Iteración 3- Cargar áreas de búsqueda almacenadas.....	99
Tabla 3.21: Iteración 3- Establecer término de interés mediante campo de texto	99
Tabla 3.22: Iteración 3- Realizar consulta por API-REST a servicio integrador de bases de datos	100
Tabla 3.23: Iteración 3- Mostrar skeleton	100
Tabla 3.24: Iteración 3- El servicio responde y no existe resultados.....	100
Tabla 3.25: Iteración 3- El servicio responde y existe resultados	101
Tabla 3.26: Iteración 3- Mostrar listado final al usuario	101
Tabla 3.27: Iteración 3- Observar revista científica.....	101
Tabla 3.28: Iteración 4- Mostrar artículos según término de interés	107
Tabla 3.29: Iteración 4- Cargar texto por defecto almacenado	107
Tabla 3.30: Iteración 4- Cargar áreas de búsqueda almacenadas.....	107
Tabla 3.31: Iteración 4- Cargar número de artículos almacenado	108
Tabla 3.32: Iteración 4- Cambiar término de interés por defecto	108
Tabla 3.33: Iteración 4- Cambiar número de revistas a buscar y agrupar.....	109
Tabla 3.34: Iteración 4- Cambiar áreas de búsqueda.....	109
Tabla 3.35: Iteración 5- Mostrar consulta de revista	115
Tabla 3.36: Iteración 5- Recibir parámetros enviados.....	115
Tabla 3.37: Iteración 5- Consultar por issn o eissn.....	116
Tabla 3.38: Iteración 5- Mostrar respuesta	116
Tabla 3.39: Iteración 5- Mostrar artículos por revista	116
Tabla 3.40: Iteración 5- Mostrar artículos por revista	117
Tabla 3.41: Iteración 6- Mostrar artículos por issn	121
Tabla 3.42: Iteración 6- Recibir parámetros enviados.....	121
Tabla 3.43: Iteración 6- Realizar consulta por API-REST a servicio integrador de bases de datos	121
Tabla 3.44: Iteración 6- Mostrar skeleton	122
Tabla 3.45: Iteración 6- El servicio responde	122
Tabla 3.46: Iteración 6- Observar artículo científico	122
Tabla 3.47: Iteración 6- Cargar los siguientes artículos	123

RESUMEN EJECUTIVO

Los docentes investigadores deben buscar artículos científicos para tener referencias, puntos de inicio o desde dónde continuar con una investigación determinada. De la misma manera, para publicar un artículo en una revista, el investigador debe identificar la mejor opción de dónde publicar. En cualquiera de los dos procesos anteriormente mencionados se consume gran cantidad de tiempo de consulta de las bases de datos especializadas disponibles en la web.

El presente proyecto se enfoca en facilitar el procesamiento de bases de datos de publicaciones científicas y sugerir artículos relacionados con un tema, y revistas científicas donde publicar. Como parte de este trabajo se desarrolló una aplicación para móvil en Flutter y se utilizó Python para el *web scraping* de las páginas web.

El uso de Flutter evitó el exceso de consultas a bases de datos de terceros y la programación Python facilitó la transformación de los datos y metadatos, de acuerdo con la necesidad de los usuarios. La metodología aplicada para el proyecto fue Mobile-D permitiendo enfocarse en la investigación, en la fase de iniciación, y realizar ciclos cortos de desarrollo iterativo.

A partir del resultado de búsqueda, el usuario puede consultar los metadatos de una publicación, según su indexación en las bases de datos científicas. Del mismo modo, puede consultar información detallada del nivel de impacto de una fuente de publicación científica. La utilidad de la aplicación fue validada exitosamente con la participación de docentes investigadores mediante una encuesta para medir la escala de usabilidad del sistema (SUS).

Palabras clave: Python, Web Scraping, API-REST, Beautifulsoup, Dart, Flutter

ABSTRACT

Research teachers must look for scientific articles to have references, starting points or from where to continue with a specific investigation. In the same way, to publish an article in a journal, the researcher must identify the best option of where to publish. In either of the two aforementioned processes, a large amount of time is consumed in consulting the specialized databases available on the web.

This project focuses on facilitating the processing of databases of scientific publications and suggesting articles related to a topic, and scientific journals to publish. As part of this work, a mobile application was developed in Flutter and Python was used for web scraping of web pages.

The use of Flutter avoided excessive queries to third-party databases and Python programming facilitated the transformation of data and metadata, according to the needs of the users. The methodology applied for the project was Mobile-D, allowing to focus on research, in the initiation phase, and to carry out short cycles of iterative development.

From the search result, the user can consult the metadata of a publication, according to its indexing in scientific databases. Similarly, you can view detailed information on the level of impact of a scientific publication source. The utility of the application was successfully validated with the participation of research professors through a survey to measure the system usability scale (SUS).

Keywords: Python, Web Scraping, API-REST, Beautifulsoup, Dart, Flutter

INTRODUCCIÓN

En los últimos años, el crecimiento exponencial de internet y la gran cantidad de información disponible ha generado un problema para los docentes investigadores en su búsqueda de publicaciones y lugares de divulgación.

A partir del problema mencionado se ha planteado desarrollar un sistema de recomendación que facilitará el procesamiento de bases de datos de publicaciones científicas optimizando tiempos utilizados en investigación, agrupando metadatos de varias bibliotecas virtuales en un solo servicio que utilizará fuentes de consumo brindadas por las plataformas (API-REST) o, al no ser proporcionado y de ser el caso, realizar Web Scraping para obtener información relevante para ser analizada y agrupada en el servicio.

La forma en la cual se plantea realizar una recomendación será que el usuario ingrese un término de interés (con la posibilidad de añadir filtros adicionales), este realizará una consulta al servicio integrador de bases de datos y mostrará los resultados para que puedan ser observados y analizados por el investigador.

Para mostrar el nivel de impacto del sistema se utilizará la Escala de Usabilidad del Sistema (SUS) que basa en realizar una encuesta de preguntas utilizando la escala de Likert, pudiendo realizarse con tamaños de muestra pequeños; además que, según Nielsen al contar con expertos recomienda realizar la evaluación entre 5 a 11, facilitando la toma de datos al encontrarnos en situación de emergencia por el COVID-19.

El presente proyecto de investigación denominado “DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN EN PLATAFORMA MÓVIL PARA FACILITAR EL PROCESAMIENTO DE BASES DE DATOS DE PUBLICACIONES CIENTÍFICAS” se encuentra dividido en los siguientes capítulos:

CAPÍTULO I. “MARCO TEÓRICO”, se plantea un problema a investigar y se establece la justificación y los objetivos que guiarán el desarrollo del proyecto.

CAPÍTULO II. “METODOLOGÍA”, se reúne todas las técnicas y herramientas necesarias para el desarrollo del proyecto, además se define las etapas de que cubrirán el desarrollo del mismo.

CAPÍTULO III. “RESULTADOS Y DISCUSIÓN”, se describe de manera precisa el desarrollo del proyecto, sus características y funcionamiento haciendo énfasis en las partes fundamentales del desarrollo.

CAPÍTULO IV. “CONCLUSIONES Y RECOMENDACIONES”, en esta sección se señalan las conclusiones y recomendaciones que se han encontrado a lo largo del desarrollo del proyecto.

CAPÍTULO I.-

MARCO TEÓRICO

1.1 Tema de investigación

Desarrollo de un sistema de recomendación en plataforma móvil para facilitar el procesamiento de bases de datos de publicaciones científicas.

1.2 Antecedentes investigativos

1.2.1 Contextualización del problema

Los sistemas de recomendación han tomado mucho interés a nivel mundial debido a que se basan en el análisis de datos recopilados y estandarizados, para luego proceder a dar sugerencias al usuario [1] [2] [3]. El resultado que se persigue es 1) obtener información de interés para el usuario a través del procesamiento semiautomático de grandes volúmenes de datos y 2) disminuir el margen de tiempo en que se obtiene dicha información, incrementando así el nivel de satisfacción de los usuarios del sistema en cuestión.

En España, las terminaciones: “Este artículo podría gustarle”, “Personas que compraron este artículo también compraron...” o “También podría interesarle” son utilizadas a menudo por los usuarios. Con ello han generado una enorme cantidad de datos en torno a sus preferencias para luego aplicar técnicas de Aprendizaje de Máquina para desarrollar un sistema que genera recomendaciones al momento de realizar una compra [4].

En Guayaquil, Ecuador, se reporta que los pacientes con diabetes y asma tienen que tener un control estricto de sus actividades y con los productos que consumen. Por ello, se están utilizando nuevas tecnologías para acortar procesos y optimizar recursos para las personas. Tal es el caso de la creación de una aplicación en la cual un médico puede obtener información importante de un paciente y su trayectoria en el cuidado de la enfermedad, para saber cómo se la podría seguir tratando basado en el análisis de casos similares registrados previamente [5].

En Morona, Ecuador, se recomiendan destinos turísticos y, para verificar la exactitud de los datos, se realizó pruebas de validación por el usuario obteniendo resultados positivos por los consejos brindados por el sistema. Para asegurar la fiabilidad de inferencia, se recomienda a la agencia turística realizar auditorías continuas a la información ingresada para verificarla con los usuarios encargados de la gestión del conocimiento [6].

En el ámbito de la colaboración de la comunidad científica, un problema que se presenta es encontrar el espacio más adecuado donde publicar los resultados de investigación alcanzados [7]. Cuando un investigador escribe un artículo científico, este procede a realizar una revisión de artículos relacionados, fundamentalmente de entre los publicados en los últimos tres años. El objetivo de esta búsqueda bibliográfica es 1) determinar la existencia de trabajos relacionados y 2) determinar dónde tendrá mejor espacio la publicación científica que se pretende.

El problema que enfrentan los investigadores está relacionado con el volumen de información que manejan las bases de datos que indexan las publicaciones científicas realizadas a escala global. Las decisiones a tomar deben tener en cuenta factores como la rigurosidad de la indexación (LATINDEX, SCOPUS, WOK), el país donde se publica, el contenido de la publicación, y otros criterios que miden su impacto, relevancia o área de interés. Al existir un gran volumen de datos para leer, analizar y decidir es de mucha dificultad encontrar el lugar en donde se pueda encontrar datos para iniciar y publicar los resultados de una investigación científica.

1.2.2 Fundamentación teórica

1.2.2.1 Sistema de recomendación (SR)

Con el crecimiento exponencial del uso del internet para la búsqueda de documentos y la gran cantidad de información disponible, se ha vuelto más laborioso realizar una consulta. El problema se ve en un exponencial aumento cuando se quiere realizar una búsqueda con un tipo de información específica para seleccionar las opciones que más se ajuste a lo investigado. Es por esto que, los Sistemas de Recomendación han aumentado considerablemente su popularidad en los últimos años ayudando a reducir el exceso de información realizado en los procesos de búsqueda de información; enviando parámetros que contribuirán a obtener mejores resultados en los ítems de información recuperados; para ello, se puede utilizar varias técnicas que ayudarán a identificar los ítems que satisfagan las preferencias o necesidades de los usuarios, y así, poder dar opciones que resultarán de interés [8].

Método de generación de recomendaciones. A continuación, se describirá los métodos de recomendación usados concurrentemente en los Sistemas de Recomendación; pero, teniendo en cuenta que estos no son recíprocamente exclusivos entre sí, más bien complementarios; es decir, un mismo Sistema de Recomendación podrá usar uno a varios de estos métodos; en un principio, se enunciarán los tres métodos más simples: [9]

La recuperación pura o recomendación nula es en la que el sistema muestra una interfaz en la cual el usuario podrá ingresar términos y realizar una consulta a una base de datos; este se trata de un sistema de búsqueda, mas no es un método de recomendación, pero es mostrado como tal para los usuarios [9].

Los sistemas aplican recomendaciones en base a selecciones por expertos. Estos identifican ítems basándose en sus preferencias, intereses y objetivos. A partir de ello poder crear una lista de ítems que se pondrán disponibles para todos los usuarios que utilizarán el sistema. Las recomendaciones usualmente irán con un comentario que ayudará a los usuarios para que puedan evaluar o entender la recomendación brindada [9].

Los sistemas que ofrecerán resúmenes estadísticos calculados teniendo en cuenta las opiniones del conjunto de usuarios, por lo que tampoco se los puede llamar personalizados [9].

1.2.2.2 Análisis de metodologías de desarrollo de software

Qué es una metodología y para qué se utiliza

Al necesitar alcanzar un objetivo que requiera habilidades y conocimientos específicos se busca aplicar un procedimiento que realice referencia a un conjunto de procedimientos racionales. La metodología es una de las partes fundamentales en un trabajo o proyecto buscando seleccionar técnicas concretas o métodos para el cumplimiento de un objetivo buscando procedimientos que se adapten a las necesidades establecidas, teniendo el fin de formalizar y optimizar actividades; determinando pasos a seguir y cómo realizarlos para cumplir con una tarea [10].

Metodología tradicional

Las metodologías tradicionales también han sido denominadas como metodologías pesadas. Esta tiene el fin de centrar la atención en llevar una documentación exhaustiva en todas las fases del proyecto, planificación y control del mismo, especificaciones de requisitos y el modelado; todo eso para buscar cumplir con un plan de trabajo, que será definido en la fase inicial del desarrollo del proyecto. Este tipo de metodología impone una disciplina rigurosa de trabajo sobre el proceso de desarrollo de software, teniendo como meta el conseguir un software más eficiente. Haciendo énfasis en la planificación del trabajo a realizar y, una vez detallado todo, comenzar el ciclo de desarrollo del producto (software). Centrándose especialmente en el control del proceso, realizando una estricta definición de roles, actividades, herramientas, artefactos y notaciones para un modelado y documentación detallada. Las metodologías tradicionales no se adaptan adecuadamente a un cambio de improviso, por lo que no es adecuado utilizarla cuando se trabaja en un entorno, en el cual los requisitos no podrán ser predichos o podrán variar. Otra característica importante es el alto costo de

implementar un cambio y la falta de flexibilidad en donde el entorno del proyecto es volátil [10].

Metodología ágil

Esta metodología nace a partir de los problemas ocasionados por las metodologías tradicionales; el cual se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. El modelo de desarrollo ágil es considerado un proceso incremental (entregas frecuentes con ciclos rápidos), también se adapta a un entorno cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación constante), sencillo (fácil de aprender y de modificación para el equipo) y adaptativo (se adapta a cambios de último momento). Una de las partes a destacar sobre esta metodología es la generación de poca documentación y no hacer uso de métodos formales. Esta metodología fue creada principalmente con el fin de tener la capacidad de respuesta ante un cambio, anteponiéndola sobre el seguimiento estricto de un plan [10].

Diferencias entre metodología tradicional y ágil

Una vez definido los conceptos sobre metodología tradicional y ágil se puede establecer un cuadro comparativo mostrando los procesos que más enfatizan de cada uno (*Tabla 1.1*).

Metodología tradicional	Metodología ágil
Documentación completa	Corta documentación
Preparado para cambios mientras el proyecto se encuentra en desarrollo	Resistencia a cambios
El cliente forma parte del equipo de desarrollo	El cliente participará en reuniones con el equipo de desarrollo
Muchas muestras de avances del proyecto	Escasas muestras de avances del proyecto
Corto número de integrantes (menor a 10)	Gran cantidad de integrantes
Impuesto por el equipo de desarrollo	Impuesto exteriormente
Poca redundancia en la arquitectura de software	Alta relevancia en establecer la arquitectura de software (expresado mediante modelos)

*Tabla 1.1: Diferencias entre metodología tradicional y ágil
Fuente: Elaboración propia a partir de [10]*

Según lo establecido en la *Tabla 1.1*, se puede considerar que las metodologías ágiles trabajan conjuntamente con el cliente durante su fase de producción; esto previene que el equipo de desarrollo cometa errores mientras se programa la fase lógica de la aplicación, además que reduce el uso de tiempo y recursos, obteniendo resultados superiores o idénticos al utilizar una metodología tradicional.

Extreme Programming (XP)

XP es una metodología ideal para su uso en equipos medianos y pequeños, cuando los requerimientos de un software son cambiantes y ambiguos. Tiene como diferencia de las metodologías tradicionales asumir el cambio como algo natural y que en alguna instancia del proyecto sucederá. XP responde a la creación de un software siguiendo los requerimientos establecidos por el usuario que pueden ser cambiados en cualquier momento. Esto se debe a que XP puede adaptarse de forma inmediata a cambios, con bajos costos y en cualquier etapa del ciclo de desarrollo. Se conoce a XP como la metodología que “abrazo” el cambio [11].

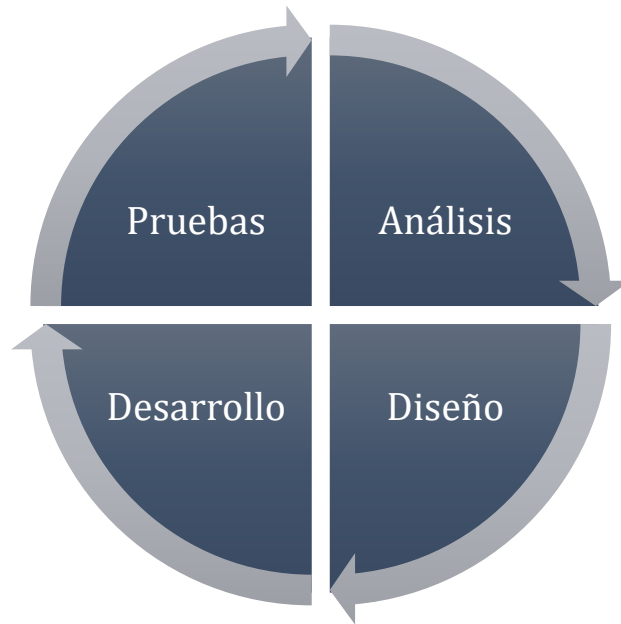
Según [12] nos plantea 12 “prácticas básicas” que se deberán seguir para que esta metodología funcione (*Tabla 1.2*).

Criterio	Descripción
Equipo completo	Forman parte del equipo todas las personas que tienen algo que ver con el proyecto, incluido el cliente y el responsable del proyecto
Planificación	Se crea las historias de usuario y se planifica en qué orden se van a hacer y las mini-versiones. La planificación se revisa continuamente
Test del cliente	El cliente, con la ayuda de los desarrolladores, propone sus propias pruebas para validar las mini-versiones
Versiones pequeñas	Las mini-versiones deben ser lo suficientemente pequeñas como para poder realizarlas una por semana. Deben ser versiones que ofrezcan algo útil al usuario final y no trozos de código que no pueda ver funcionando
Diseño simple	Hacer siempre lo mínimo imprescindible de la forma más sencilla posible. Mantener siempre sencillo el código
Pareja de programadores	Los programadores trabajan por parejas (dos delante del mismo ordenador) y se intercambian las parejas con frecuencia (un cambio diario)

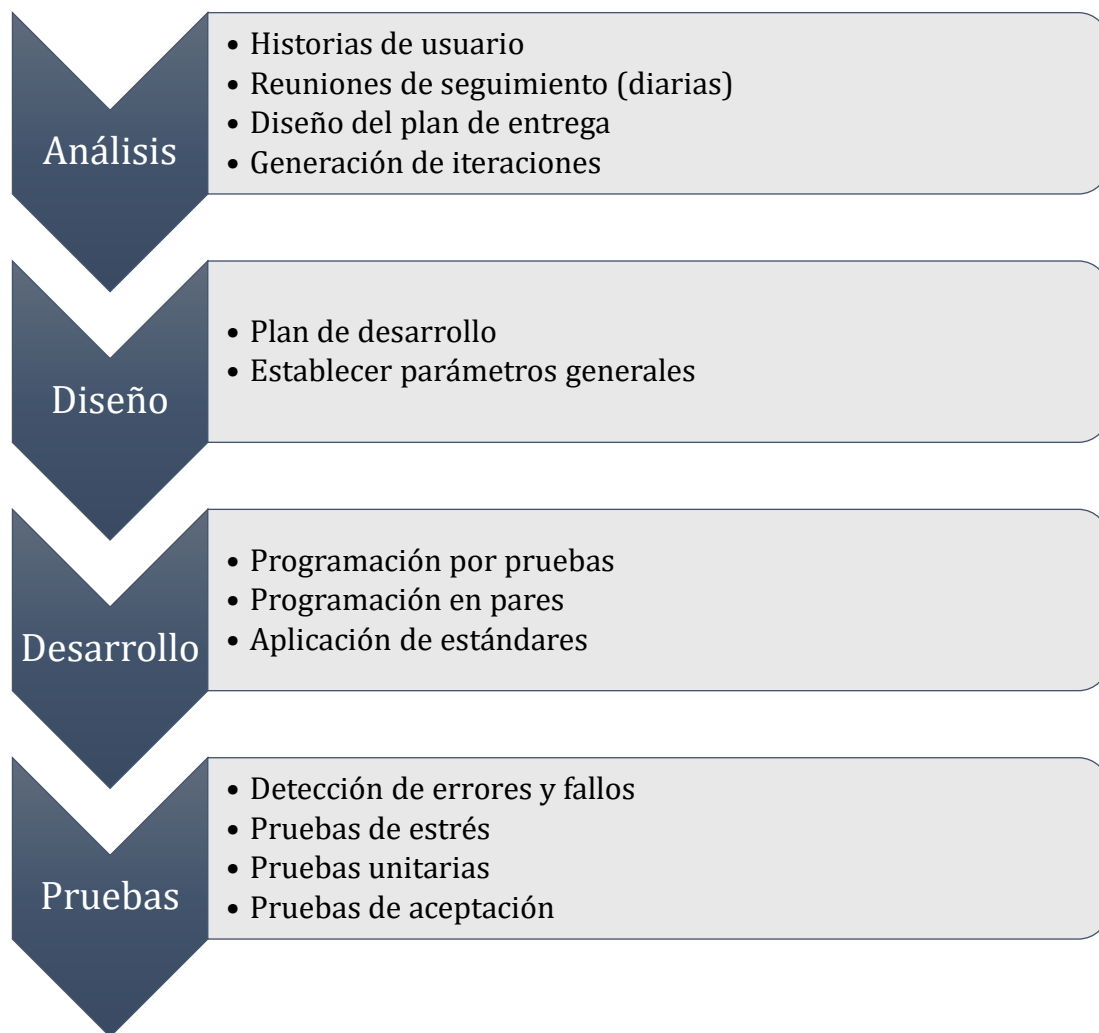
Criterio	Descripción
Desarrollo guiado por las pruebas automáticas	Se debe realizar programas de prueba automática y debe ejecutarse con mucha frecuencia. Cuantas más pruebas se realicen se minimizará el margen de errores generados
Integración continua	Debe tenerse siempre un ejecutable del proyecto que funcione y en cuanto se tenga una nueva pequeña funcionalidad, debe recompilarse y probarse. Es un error mantener una versión congelada por meses mientras se hacen mejoras para luego integrarlas al sistema. Cuando falle alguna funcionalidad, no se tendrá el conocimiento sobre cuál es el proceso que está generando problemas
El código es de todos	Cualquiera puede y debe tocar y conocer cualquier parte del código. Para ello se realizan las pruebas automáticas
Normas de codificación	Debe haber un estilo común de codificación (no importa cuál), de forma que parezca que ha sido realizado por una única persona
Metáforas	Se debe buscar unas frases o nombres que definan cómo funcionan las distintas partes del programa, de forma que sólo con los nombres se pueda uno hacer una idea de qué es lo que hace cada parte del programa. Un ejemplo claro es el "recolector de basura" de java. Ayuda a que todos los programadores (y el cliente) sepan de qué se está hablando y que no se creen mal entendidos
Ritmo sostenible	Se debe trabajar a un ritmo que se pueda mantener indefinidamente. Esto quiere decir que no debe haber días muertos en que no se sabe qué hacer y que no se deben hacer un exceso de horas otros días. Al tener claro semana a semana lo que debe hacerse, hay que trabajar duro en ello para conseguir el objetivo cercano de terminar una historia de usuario o mini-versión

Tabla 1.2: Prácticas básicas de XP
Fuente: Elaboración propia a partir de [12].

El ciclo de desarrollo está establecido en 4 fases: análisis, diseño, desarrollo y pruebas (Figura 1.1) y en cada fase tiene su proceso a realizar (Figura 1.2).



*Figura 1.1: Ciclo de desarrollo Extreme Programming (XP)
Fuente: Elaboración propia a partir de [13]*



*Figura 1.2: Procesos de fases de Extreme Programming (XP)
Fuente: Elaboración propia a partir de [13]*

Mobile-D

Es una metodología ágil que se ha apoyado en otras soluciones conocidas y con buenas bases, entre ellas tenemos: Extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP) [14].

Según [14], Mobile-D apunta a la necesidad de disponer de un ciclo de desarrollo muy rápido para equipos muy pequeños. De acuerdo con sus suposiciones, Mobile-D está pensado para grupos de no más de 10 desarrolladores colaborando en un mismo espacio físico. Si se trabaja con el ciclo de desarrollo propuesto, los proyectos deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas.

El ciclo de desarrollo está establecido en 5 fases: exploración, inicialización, producción, estabilización y prueba y reparación (*Figura 1.3*).

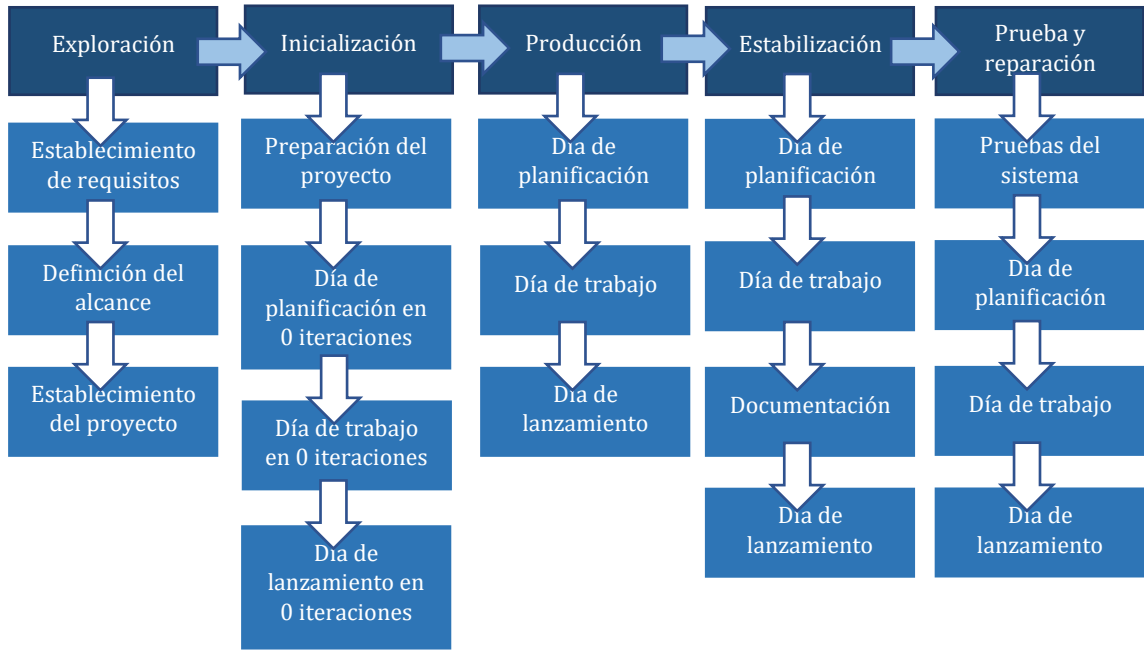


Figura 1.3: Ciclo de desarrollo Mobile-D
Fuente: Elaboración propia a partir de [14]

Análisis comparativo entre las metodologías Extreme Programming (XP) y Mobile-D

En la *Tabla 1.3* se expone un análisis que mostrará lo más importante a tener en cuenta sobre las metodologías XP y Mobile-D.

Extreme Programming (XP)	Mobile-D
Cada iteración se realiza entre una a tres semanas	Se planifica la iteración de trabajo en términos de requisitos y tareas a realizar
Grupo pequeño (máximo 12 personas)	Grupo de no más de 10 personas colaborando en un mismo espacio físico
Tiene una fase en la cual analizan los requerimientos y prepara la fase de diseño, desarrollo y pruebas	Tiene una fase en la cual preparan las siguientes fases y establecen el entorno técnico (entrenamiento del equipo de desarrollo)

Tabla 1.3: Análisis comparativo entre las metodologías Extreme Programming (XP) y Mobile-D
Fuente: Elaboración propia

1.2.2.3 Análisis de tecnologías de desarrollo móvil

Aplicaciones Nativas

Las aplicaciones nativas son aquellas que se ejecutan directamente en el sistema operativo del teléfono. Para ello existen tecnologías estables y populares y tecnologías alternativas y en crecimiento para que se tenga en cuenta en el momento de la selección en la creación del software [15], mismas que son detalladas a continuación.

Tecnologías estables y populares:

- Para iOS: Swift y Objective-C
- Para Android: Kotlin y Java

Tecnologías alternativas y en crecimiento:

- Flutter, React Native, Xamarin y Native Script

Aplicaciones Híbridas

Las aplicaciones híbridas no se ejecutan directamente en el sistema operativo del teléfono, sino que utilizan un medio intermedio en el cual poder ejecutarse, que en la mayoría de veces es el navegador web. Es por ello que ya no es necesario desarrollar dos aplicaciones independientes, sino una que esté optimizada para correr en el navegador, pero con la posibilidad de acceder a ciertas características propias del teléfono utilizando componentes externos que permiten al navegador comunicarse con ellos [15].

Entre estas tecnologías tenemos las siguientes:

- Ionic
- Apache Cordova
- Unity

Aplicaciones Web Móvil

Una aplicación Web se refiere a un software o cualquier aplicación que se puede tener acceso desde un dispositivo móvil sin la necesidad de instalarla para su uso, comúnmente se los puede encontrar en la nube, la cual es un servidor que se puede acceder para utilizar contenido ofrecido por una compañía determinado; pero, debe cumplir con elementos de uso para las diferentes plataformas, ya sea Android, iOS, Windows Phone, etc [16].

Características de funcionalidades en dispositivos móviles

La *Tabla 1.4* expone las características más importantes de cada tecnología de desarrollo móvil a tener en cuenta.

Tipo			
Característica	Web Móvil	Nativa	Híbrida
Plataforma	Navegadores webs móviles	<ul style="list-style-type: none"> • iOS • Android • Windows Phone • BlackBerry OS • Symbian 	<ul style="list-style-type: none"> • iOS • Android • Windows Phone • BlackBerry OS • Symbian
Distribución	url's	Tiendas de aplicaciones según plataforma	Tiendas de aplicaciones según plataforma
Instalación	Se realiza una sola vez y queda disponible para todas las plataformas	Más plataformas soportan la aplicación. Se eleva la dificultad en el soporte y mantenimiento	Se realiza una sola vez y queda disponible para todas las plataformas
Costos de desarrollo	Significativamente más económica	Costosas	Reducción considerable respecto al costo de las nativas
Rendimiento	HTML5 mejora la infraestructura de la web	Más rápido, especialmente si requiere procesos gráficos pesados	Se desarrolla como nativo; donde el rendimiento sea crítico
Integración hardware	Limitado	Completo	Buena integración
Acceso offline	Sólo en algunos dispositivos puede lograrse a través de HTML5	Funcionan offline	Funcionan offline

Usabilidad	Buena	Gran cantidad de efectos amigables en la interfaz, resultando ser atractivos para el usuario	Utiliza lo mejor de lo nativo y lo mejor de lo web
------------	-------	----------------------------------------------------------------------------------------------	----------------------------------------------------

Tabla 1.4: Características de funcionalidades en dispositivos móviles
Fuente: [17]

La *Tabla 1.5* nos muestra los aspectos que debemos tener en cuenta para determinar la mejor opción para realizar una aplicación móvil.

Característica	Web móvil	Híbrida	Nativa
La aplicación necesita acceso al hardware del dispositivo como la cámara o el GPS	✓	✓	✓
La aplicación debe ser funcional sin conexión	✓	✓	✓
La aplicación requiere cálculos a tiempo real o gráficos 3D de Alto Rendimiento	✓	✓	✓
Debe tener presencia en sitios como Google Play, AppStore y AppWorld	✓	✓	✓
La aplicación tendrá cambios regulares en las reglas de negocio	✓	✓	✓
El presupuesto para desarrollar la aplicación es reducido	✓	✓	✓
La Aplicación depende de constante con el servidor	✓	✓	✓

✓ Mejor opción, ✓ Opción Intermedia, ✓ Peor Intermedia

Tabla 1.5: Aspectos para realizar una aplicación móvil
Fuente: [17]

Análisis de frameworks de desarrollo móvil

React Native

React Native es una plataforma desarrollada por Facebook, de código abierto para el desarrollo de aplicaciones cross-platform nativas utilizando React y Javascript; así, pudiendo crear aplicaciones para iOS, Android, Ubuntu, etc. La gran ventaja de React Native es que utiliza el lenguaje más utilizado del mundo (Javascript) y el gestor de paquetes más grande del mundo (NPM) haciéndolo muy accesible y teniendo una curva de aprendizaje muy baja, además de ofrecer a los desarrolladores componentes del teléfono para poder añadir a la aplicación (de ser el caso) [18].

Cómo funciona React-Native

Para una mejor comprensión de cómo funciona React Native emulando vista de navegador en una aplicación se mostrará a continuación el proceso que se realiza, hasta generarse el aplicativo.

Estructura del código que representa la aplicación.



Figura 1.4: Código representativo de React Native
Fuente: [18]

Debido a que el código se encuentra escrito en JavaScript se lo debe ejecutar en un motor de Javascript (v8, el motor de Chrome) y en iOS se lo conoce como JavascriptCore [18].

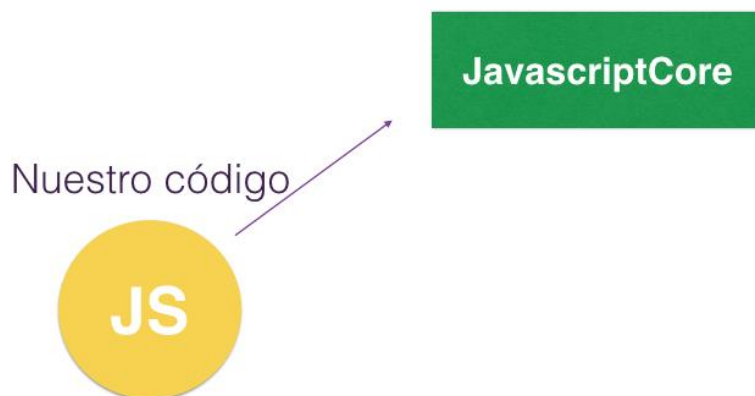


Figura 1.5: Ejecución de proyecto en motor Javascript

Fuente: [18]

Después el motor se comunica con un puente nativo que está escrito en C++ en IOS y Java en Android. Esta comunicación se hace a partir de serializar los datos (como JSON en cualquier aplicación web, por ejemplo) [18].

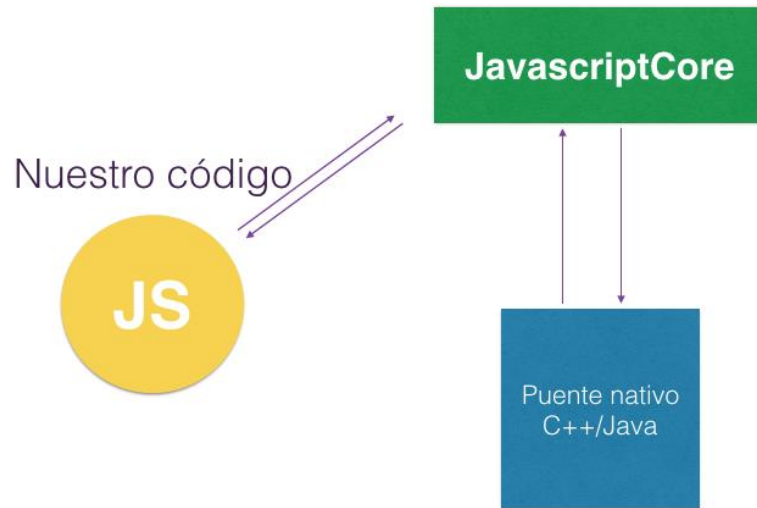


Figura 1.6: Comunicación puente nativo
Fuente: [18]

El puente nativo se conecta con el dispositivo.

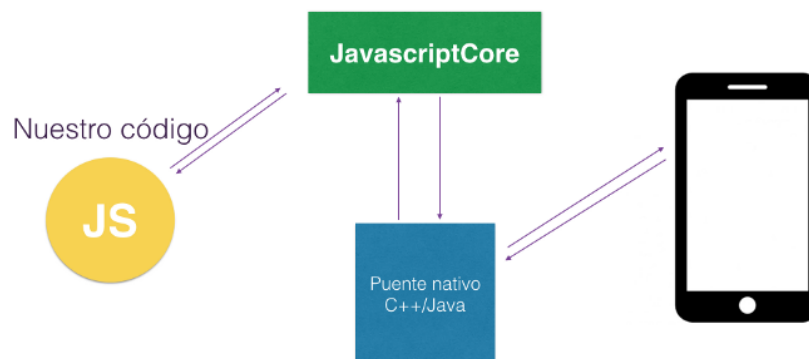


Figura 1.7: Conexión puente nativo con dispositivo
Fuente: [18]

Se debe tener en cuenta que la comunicación se la realiza bidireccionalmente.

Flutter

Flutter es el kit de herramientas de UI de Google para realizar aplicaciones, compiladas nativamente, se utiliza para el desarrollo de aplicaciones Android, iOS, Windows, Mac, Linux y Web [19].

Flutter es un framework desarrollado por Google para realizar aplicaciones que se pueden compilar nativamente, esta es utilizada para el desarrollo de aplicaciones de Android, iOS y por el momento se encuentra en versiones beta de aplicaciones para Escritorio y Web. Flutter lanzó su versión alfa el mayo de 2017 y en 11 de diciembre de 2019, anunció su versión estable. Esta herramienta es muy reciente y está teniendo un enorme éxito por las facilidades y beneficios que ha dado a los desarrolladores en la creación de programas [19].

Desarrollo de Flutter

Flutter utiliza el lenguaje de Dart, éste se compila por adelantado (AOT) en código nativo para múltiples plataformas, sin tener que pasar por un puente JavaScript, además de mostrar algo mucho más revolucionario en implementar vistas para el usuario, conocidos como Widgets [19].

Comparativa entre Flutter y React Native

Una vez establecido los frameworks de React Native y Flutter; se expone a continuación (*Figura 1.8 y Figura 1.9*), una vista gráfica del proceso que sigue cada uno para su funcionamiento.

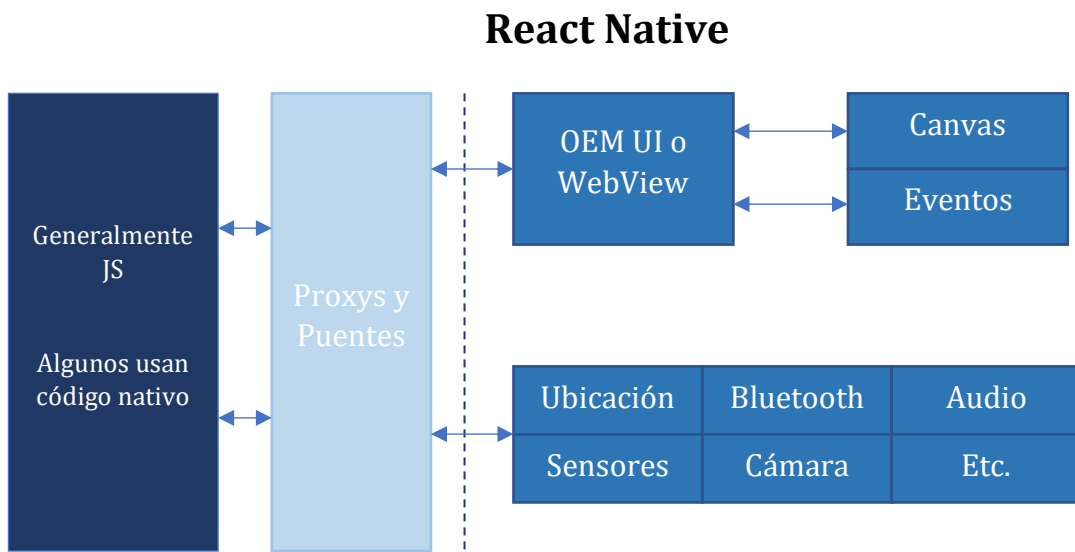


Figura 1.8: Arquitectura React Native
Fuente: Elaboración propia a partir de [20]

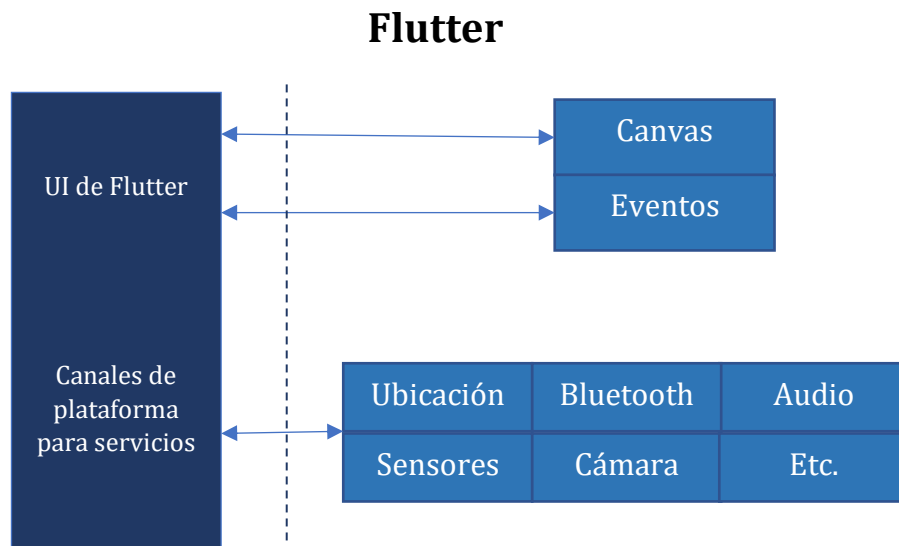


Figura 1.9: Arquitectura Flutter
Fuente: Elaboración propia a partir de [20]

Como se puede observar en la *Figura 1.8* y *Figura 1.9* Flutter no necesita de puentes o proxys, al contrario de React Native.

1.2.2.4 Análisis de lenguajes para el análisis de datos

R

Es un lenguaje de programación interpretado, es decir, ejecuta instrucciones directamente, sin una previa compilación a instrucciones en lenguaje de máquina. Se le da el término de entorno a que es un sistema planificado y coherente, en lugar de una acumulación de herramientas específicas e inflexibles, que suceden en otros softwares de análisis de datos [21].

Características de R

Entre las características a destacar por [21] tenemos:

- Manejo y almacenamiento efectivo de los datos.
- Un conjunto de operadores para la realización de cálculos con matrices.
- Una gran colección de herramientas para el análisis de datos.
- Utilidades gráficas para la visualización de datos.
- Un lenguaje de programación bien desarrollado que incluye saltos condicionales, bucles, funciones recursivas, utilidades para la entrada y salida de datos, etc.
- Tiene un formato de documentación basado en LaTeX, que se utiliza para proporcionar documentación completa tanto en formato físico como digital.

Uso de R en Big Data

R es utilizado para la manipulación, procesamiento y visualización gráfica de datos, además de [21]:

- Crear visualizaciones de datos de alta calidad.
- Crear dashboards para visualizar y analizar datos.
- Crear informes automáticos.
- Disponer de herramientas de análisis estadístico para ahondar en el conocimiento de los datos.

Python

Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de plataformas [22].

Beneficios de utilizar Python en Big Data

[23] muestra varios beneficios en la utilización de Python para el Big Data, pero entre lo que más se puede destacar es:

Simplicidad. Python es conocido por hacer que los programas funcionen en la menor cantidad de líneas de código. Identifica y asocia automáticamente los tipos de datos y, en general, resulta un lenguaje fácil de usar y toma menos tiempo en la codificación. Tampoco hay limitación para el procesamiento de datos. Puede calcular datos en cualquier tipo de equipo y entorno, básicamente en todas partes.

Facilidad de aprendizaje. En comparación con otros idiomas, Python es fácil de aprender incluso para los programadores con menos experiencia. Es un primer idioma ideal debido a tres razones principales: cuenta con amplios recursos de aprendizaje, garantiza un código legible y se rodea de una gran comunidad.

Paquetes de gran alcance. Python tiene un poderoso conjunto de paquetes para una amplia gama de necesidades de análisis y ciencia de datos. Algunos de los paquetes populares que le dan ventaja son NumPy, Pandas, Scipy, Scikit-learn, PyBrain, Tensorflow, Cython, PyMySQL, BeautifulSoup o iPython.

Visualización de datos. Python para big data ha mejorado su oferta en este espacio. Ahora existen API que pueden ofrecer buenos resultados.

Comparativa entre R y Python

Expuesto el funcionamiento de R y Python, [24] creó criterios tomando sus ventajas y desventajas de su uso, mismos que son expuestos a continuación:

- R es un lenguaje orientado al análisis estadístico que se utiliza ampliamente en el campo de la ciencia de datos.
- Python es un lenguaje de alto nivel multipropósito utilizado además en otros campos (desarrollo web, scripting, etc.).
- R es un lenguaje más lento que Python en ejecución.
- R es más potente en visualización de información y datos que Python.

1.2.2.5 Definición de arquitecturas

Arquitectura del aplicativo móvil

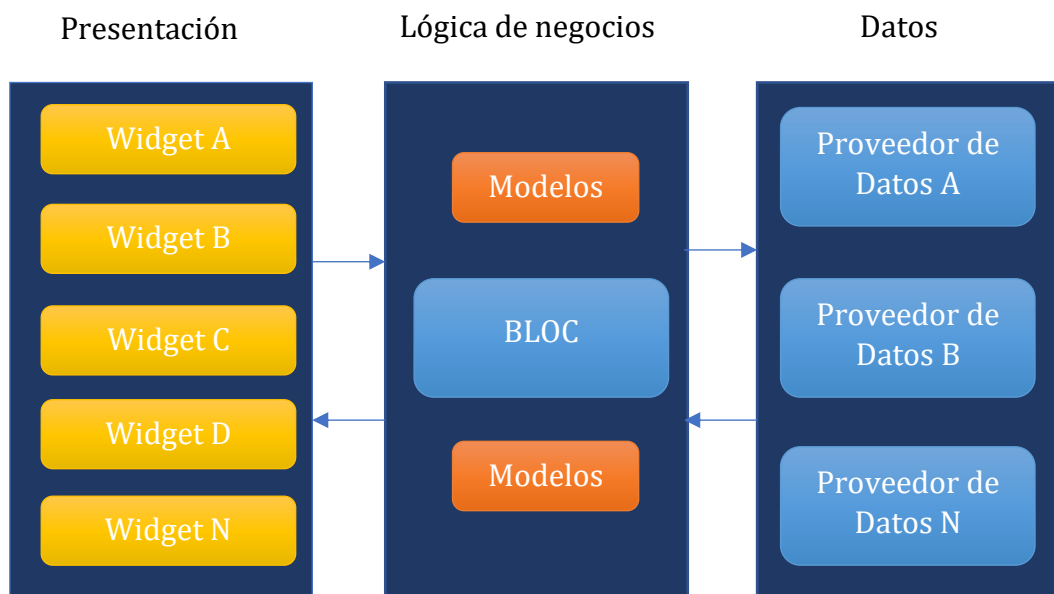


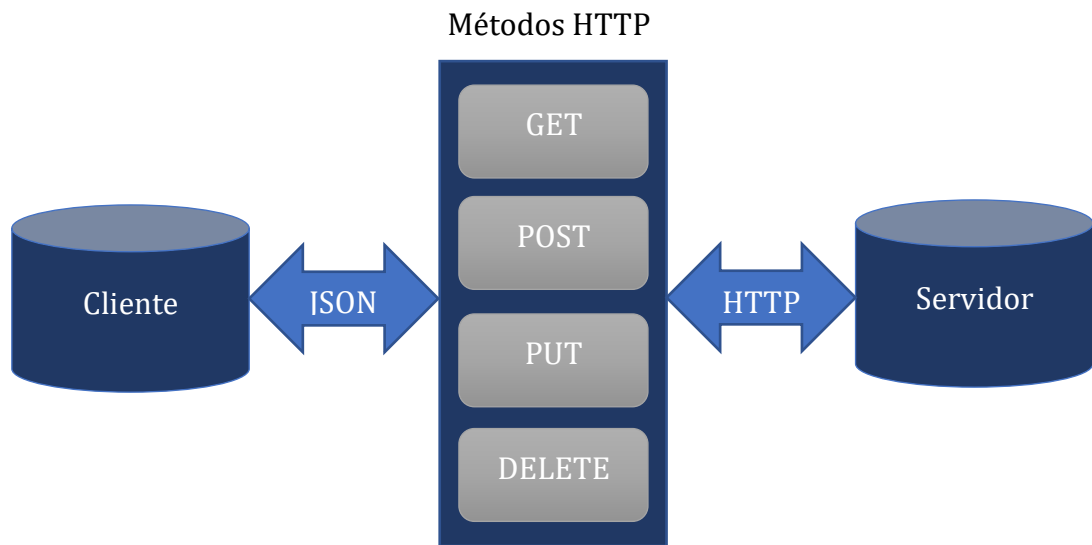
Figura 1.10: Arquitectura del aplicativo móvil
Fuente: Elaboración propia a partir de [25]

Arquitectura del aplicativo de análisis de datos



*Figura 1.11: Arquitectura del aplicativo de análisis de datos
Fuente: Elaboración propia a partir de [26]*

Arquitectura de API-REST



*Figura 1.12: Arquitectura de API-REST
Fuente: Elaboración propia a partir de [27]*

1.2.2.6 El estado del ecosistema de desarrolladores

Según el informe presentado por [28] en el año 2020 nos muestra una encuesta realizada a 19 696 desarrolladores. En la cual establecieron las últimas tendencias en torno a herramientas tecnológicas, lenguajes de programación y muchas otras facetas interesantes en el mundo del desarrollo; pero para dar énfasis en el uso del framework en este proyecto se tomó en cuenta la siguiente pregunta estipulada: “¿Qué frameworks móviles multiplataforma utilizas?”. Obteniendo como resultado que “La popularidad de Flutter ha aumentado durante el último año, ya que su participación ha aumentado en 9 puntos porcentuales. En el mismo período, las acciones de Cordova, Ionic y Xamarin disminuyeron aproximadamente 10 puntos porcentuales cada una” [28]. A continuación, en la Figura 1.13 se mostrará los resultados en base a la pregunta anteriormente mencionada.

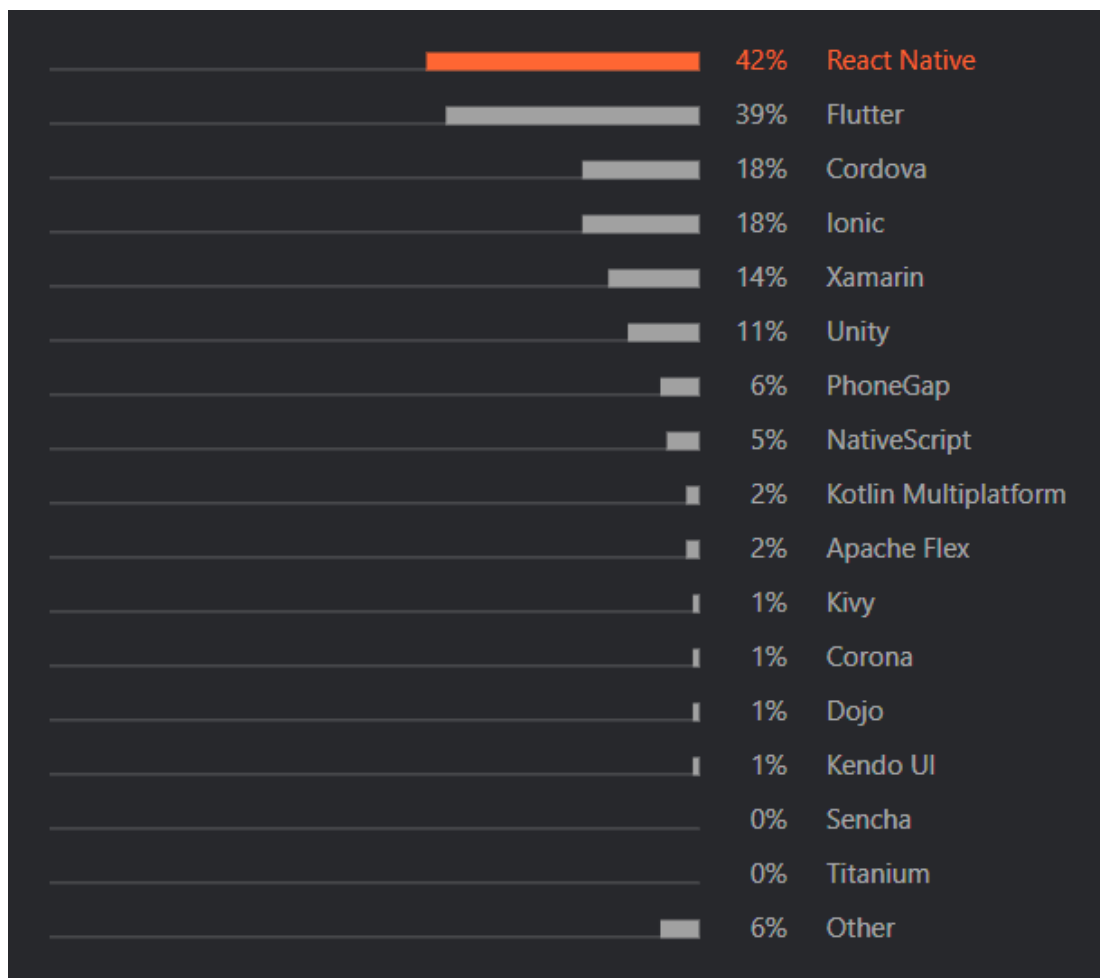


Figura 1.13: Resultado de frameworks móviles multiplataforma utilizados
Fuente: [28]

1.2.2.7 Web Scraping

Esta es una técnica que sirve para extraer información de páginas web de forma automatizada. Si se traduce del inglés, su significado vendría a ser algo así como “escarbar una web”. Es utilizado para conseguir cantidades industriales de información (Big Data) sin teclear una sola palabra, esto se realiza a través de algoritmos de búsqueda que podrán rastrear muchas páginas web para extraer solo la información que se necesite; por ello, es muy útil dominar regex (regular expressions), para delimitar las búsquedas o hacerlas más precisas y que el filtrado sea mejor [29].

El web scraping es una disciplina que debe combinar dos especialidades muy diferenciadas del conocimiento web, ambas son esenciales para poder detectar lo que se va a extraer, uno de ellos es la dominación de visualización de datos a nivel conceptual y por el otro disponer conocimientos técnicos necesarios para extraer con exactitud los datos con herramientas [29].

Web Scraping consiste en obtener datos, que se utilizarán con alguna finalidad. Es por esto que [29] muestra dos procesos claves que se realizarán a los datos una vez que estos sean obtenidos, mismos que son mostrados a continuación:

Jerarquización, ordenación y filtrado de los datos. Muchas veces cuando se extraen cantidades industriales de datos, antes de importarlos a otra plataforma se debe ‘trabajar’ con estos datos de manera precisa para depurarlos antes de utilizarlos.

Importación de los datos a otra plataforma. La importación de los datos se trata de otro proceso básico. Hay herramientas con las que se trabaja en plataformas como WordPress, como por ejemplo el plugin WP Ultimate CSV Importer, del estudio de desarrollo Smack Coders (también disponen de una versión de pago, Ultimate CSV Importer Pro).

Con un web scraper programado de forma periódica se podría acabar consiguiendo el mismo resultado para actualizar los datos de nuestra web [29].

1.2.2.8 API-REST

En los últimos años, el término API ha evolucionado con la revolución de los servicios web, a lo que hoy se lo conoce como API-REST [30].

API-REST es una biblioteca apoyada por el estándar HTTP, es decir, es un servicio que provee de funciones que dan la capacidad de hacer uso de un servicio web que no se sea propietario, dentro de una aplicación propia, de una manera segura [30].

El beneficio de utilizar un API para acceder a información del tipo que se pretende con el presente trabajo (con respecto al uso de un web scraper) es que en una API expone datos estructurados, mientras que el uso de un web scraper requiere el procesamiento de texto no estructurado, lo que implica que esta segunda variante tiene un mayor tiempo de procesamiento. Dicho de otra manera, siempre que se disponga de una API, esta será la mejor opción de integración con las fuentes de datos abiertas disponibles.

En su artículo [27] detalla las formas en la cual se puede realizar un consumo a un servicio, a través de API-REST y la respuesta que podremos tener, mismas que serán detalladas a continuación con el fin de describir el procedimiento y la respuesta que se obtendrá en el aplicativo en el consumo de servicios de terceros.

Llamadas al API

- Las llamadas al API se implementan como peticiones HTTP, en las que:
 - La URL representa el recurso
`http://www.formandome.es/api/cursos/1`

- El método (HTTP Verbs) representa la operación:
`GET http://www.formandome.es/api/cursos/1`

- El código de estado HTTP representa el resultado:
`200 OK HTTP/1.1`
`404 NOT FOUND HTTP/1.1`

Creación de recursos

- La URL estará “abierta” (el recurso todavía no existe y por tanto no tiene id)
- El método debe ser POST
`http://eventos.com/api/eventos/3/comentarios`

Respuesta a la creación de recursos

- Resultados posibles:
 - 403 (Acceso prohibido)
 - 400 (petición incorrecta, p.ej. falta un campo o su valor no es válido)
 - 500 (Error del lado del servidor al intentar crear el recurso, p.ej. se ha caído la BD)
 - 201 (Recurso creado correctamente)
- ¿Qué URL tiene el recurso recién creado?
 - La convención en REST es devolverla en la respuesta como valor de la cabecera HTTP Location

Actualización de recursos

- Método PUT
 - Según la ortodoxia REST, actualizar significaría cambiar TODOS los datos
 - PATCH es un nuevo método estándar HTTP (2010) pensado para cambiar solo ciertos datos. Muchos frameworks de programación REST todavía no lo soportan
- Resultados posibles
 - Errores ya vistos con POST
 - 201 (Recurso creado, cuando le se envía el id deseado al servidor)
 - 200 (Recurso modificado correctamente)

Eliminar recursos

- Método DELETE
- Algunos resultados posibles:
 - 200 OK
 - 404 Not found
 - 500 Server error
- Tras ejecutar el DELETE con éxito, las siguientes peticiones GET a la URL del recurso deberían devolver 404

Reglas de una arquitectura REST

- Interfaz uniforme.
- Peticiones sin estado.
- Cacheable.
- Separación de cliente y servidor.
- Sistema de Capas.
- Código bajo demanda (opcional).

1.2.2.9 Procesamiento de Bases de Datos de Bibliotecas Científicas

Para una completa explicación sobre el proceso que se realizó para el consumo de las APIs de bibliotecas virtuales se colocará los fragmentos más importantes sobre como consumirlas. Las mismas que serán presentadas a continuación.

Consumo API Springer

Colecciones válidas y tipos de respuesta

Dependiendo la colección de datos que se envíe como parámetro se obtendrá un listado de datos, estableciendo el formato que se desea obtener la respuesta (*Tabla 1.6*).

Colección	Descripción	Formato respuesta	Descripción
meta versionado	Almacena artículos de revistas y capítulos de libros en una nueva versión	pam	Devuelve resultados XML con cada registro en formato PRISM Aggregator Message (* con algunos campos adicionales específicos de revistas o libros no cubiertos por PAM en este momento [por ejemplo, openAccess])
		jats	Devuelve resultados XML con cada registro en formato JATS o BITS, un estándar estructurado en toda la industria para artículos y capítulos científicos (* "jats" usados en la ruta para cada uno / ambos).
		json	Devuelve resultados con formato JSON
		jsonp	Devuelve resultados con formato JSON con relleno

Colección	Descripción	Formato respuesta	Descripción
		jsonld	Devuelve resultados json con cada registro de nuestra oferta SN SciGraph.
metadata	Almacena artículos de revistas y capítulos de libros	pam	Devuelve resultados XML con cada registro en formato PRISM Aggregator Message (* con algunos campos adicionales específicos de revistas o libros no cubiertos por PAM en este momento [por ejemplo, openAccess])
		json	Devuelve resultados con formato JSON
		jsonp	Devuelve resultados con formato JSON con relleno
openaccess	Almacena contenido de acceso abierto	jats	Devuelve resultados xml con formato JATS (JATS para artículos, BITS para capítulos)
		json	Devuelve resultados de metadatos con formato JSON
		jsonp	Devuelve metadatos con formato JSON con relleno
integro	Almacena los journals de Springer	xml	Devuelve datos XML

Tabla 1.6: API-Springer: Colecciones válidas y tipos de respuesta

Fuente: [31]

Realizar búsquedas por métodos abreviados

Estos parámetros serán utilizados cuando se quiere aplicar un filtro exhaustivo a la consulta excluyendo búsquedas por terminaciones similares (*Tabla 1.7*).

Método	Descripción
doi	Encuentra un solo artículo basado en el DOI
issn	Encuentra un conjunto de artículos en un único ISSN de revista
isbn	Encuentra un conjunto de capítulos en un solo libro ISBN

Tabla 1.7: API-Springer: Métodos abreviados
Fuente: [31]

Parámetros en la cadena de consulta

Son requerimientos que el API pide para poder tomar esa información y enviarlas a su base de datos para realizar la consulta (*Tabla 1.8*).

Parámetro	Descripción	Requerido	Valor por defecto
q	La consulta que debe realizar el API. Admite varios filtros.	si	
s	Devuelve los resultados a partir del número especificado.	no	1
p	Número de resultados para devolver en esta solicitud. El número máximo de resultados devueltos en una sola consulta es 100 (20 en el caso de solicitudes de OpenAccess).	no	10
api_key	Key suministrada por Springer para identificar la aplicación.	si	
callback	Nombre del contenedor de devolución de llamada. Se usa solo para resultados JSONP.	no	

Tabla 1.8: API-Springer: Parámetros en la cadena de consulta
Fuente: [32]

El parámetro q tiene una amplia gama de opciones que le permiten construir consultas complejas [32], descritas en la *Tabla 1.9*.

Agregar restricciones de búsqueda

Como su nombre lo menciona, se podrán agregar restricciones en una búsqueda; pudiéndolos así filtrar aún más el resultado, tales como: país, libro, nombre de organización, etc. (Tabla 1.9).

Restricción	Descripción	Aplica a la colección	Consulta de ejemplo
vacío	Una palabra o frase que aparece entre las restricciones pero que no está precedida por un valor de restricción se tratará como el argumento de la "restricción vacía". Las solicitudes de este tipo ubicarán documentos que contengan la palabra o frase especificada en cualquier elemento.	M, O, M2	<i>http://api.springernature.com/metadata/json?q=%22user%20experience%22&api_key=...</i>
title:	Localizar documentos que contengan una palabra o frase en el elemento "título del artículo / capítulo"	M, O, M2	<i>http://api.springernature.com/meta/v2/pam?q=title:%22game%20theory%22&api_key=..</i>
orgname:	Localizar documentos que contengan una palabra o frase en el elemento "nombre de la organización"	M, O, M2	<i>http://api.springernature.com/metadata/json?q=orgname:Paris&api_key=..</i>

Restricción	Descripción	Aplica a la colección	Consulta de ejemplo
journal:	Localizar documentos que contengan una palabra o frase en el elemento "título del journal"	M, O, M2	http://api.springernature.com/openaccess/jats?q=journal:informatics&api_key=..
book:	Localizar documentos que contengan una palabra o frase en el elemento "título del libro"	M, O, M2	http://api.springernature.com/meta/v2/jats?q=book:%22social%20network%22&api_key=..
name:	Localizar documentos que contengan una palabra o frase en el elemento "nombre del autor"	M, O, M2	http://api.springernature.com/openaccess/jats?q=name:Salvador&api_key=..

Tabla 1.9: API-Springer: Restricciones de búsqueda
Fuente: [33]

Usar múltiples restricciones

En el siguiente fragmento [33] muestra las formas en la cual se podrá agregar múltiples restricciones en una consulta:

Cada restricción que aparece en su solicitud será automáticamente AND con todas las demás. Por ejemplo, una solicitud que incluye restricciones: *"title:bone+name:Jones"* es el equivalente a la solicitud que contiene restricciones concatenadas por el operador AND: *"title:bone%20AND%20name:Jones"*.

Para modificar este comportamiento, puede incluir el operador OR para ampliar los resultados.

Ejemplo. Una solicitud que contiene las restricciones: *"title:bone%20OR%20name:Jones"*, devolvería resultados donde el título contenía la cadena *"bone"* o el nombre del autor contenía la cadena *"Jones"*.

Los corchetes *"[]"* también se pueden utilizar para ayudar a estructurar estas construcciones booleanas.

Por ejemplo, aquí hay una solicitud válida que usa corchetes para agrupar restricciones:

```
http://api.springernature.com/meta/v2/pam?q=(title:"game theory" OR title:"perfect information")&api_key=.
```

Esta solicitud devuelve documentos donde el título incluye la frase *"teoría de juegos"* o incluye la frase *"perfect information"*.

Consejos útiles

Springer ofrece una serie de consejos que resultarán útiles en el momento de su consumo. Entre los cuales destaca: "Dado que el carácter {espacio} (representado como "%20" o "+") se utiliza para separar los pares "restricción: argumento", siempre que el argumento de una restricción sea una frase de varias palabras (p. Ej., "Sustituto óseo"), el símbolo de comillas (representado por %22 en una URL) debe usarse alrededor de la frase para que el {espacio} incluido sea reconocido como parte del argumento" [33].

Consumo API Elsevier

API de búsqueda de Scopus

Actualmente, Scopus posee la mayor base de datos de resúmenes y citas de literatura de investigación y fuentes web de calidad, cubriendo 50 millones de resúmenes de más de 20 500 títulos revisados por pares de más de 5 000 editores, ofreciendo a través de un recurso en el cual se podrá recuperar metadatos de resultados relevantes en formatos de texto específicos del usuario [34].

Scopus ofrece los siguientes parámetros que podrán ser utilizados para filtrar la información (*Tabla 1.10*).

Parámetro	Tipo	Descripción
apiKey	xsd:string	Anulación para el encabezado HTTP X-ELS-APIKey, esto representa una clave de desarrollador de aplicaciones única que proporciona acceso a los recursos.
query	xsd:string (requerido)	Esto representa la búsqueda booleana que se ejecutará contra el clúster SCOPUS. Hay información adicional sobre los consejos de búsqueda. ej. query=corazón+ataque%20AND%20text (hígado)
view	xsd:string defecto: STANDARD options: STANDARD, COMPLETE	Este alias representa la lista de elementos que se devolverán en la respuesta. La <i>Tabla 1.10</i> muestra las vistas de búsqueda de Scopus.
field	xsd:string	Este alias representa el nombre de campos específicos que deben devolverse. La lista de campos incluye todos los campos devueltos en la carga útil de respuesta (ver <i>Tabla 1.10</i>). Se pueden especificar varios campos, delimitados por comas. Tenga en cuenta que especificar este parámetro anula el parámetro de vista. ej. field=url,identifier,description
start	xsd:string	Valor numérico que representa el desplazamiento de los resultados (es decir, la posición de inicio de los resultados de búsqueda). El máximo para este

Parámetro	Tipo	Descripción
		valor es un valor predeterminado a nivel de sistema (varía con el grupo de búsqueda) menos el número de resultados solicitados. Si no se especifica, el desplazamiento se establecerá en cero (es decir, primer resultado de búsqueda) ex. start=5
count	xsd:string	Valor numérico que representa el número máximo de resultados que se devolverán para la búsqueda. Si no se proporciona, se establecerá en un sistema predeterminado según el nivel de servicio. Además, el número no puede exceder el valor máximo predeterminado del sistema; si lo hace, se devolverá un error. ex. count=10
subj	xsd:string	Representa el código de área temático asociado con la categoría de contenido deseada. Tenga en cuenta que estos mapeos de códigos de materias varían según el entorno en el que se ejecuta la solicitud. El siguiente recurso muestra todas las clasificaciones de materias disponibles. Las opciones disponibles incluyen: AGRI - Ciencias agrícolas y biológicas ARTES - Artes y Humanidades BIOC - Bioquímica, Genética y Biología Molecular BUSI - Negocios, Gestión y Contabilidad CENG - Ingeniería química CHEM - Química COMP - Ciencias de la Computación DECI - Ciencias de la decisión DENT - Odontología EART - Ciencias de la Tierra y Planetarias ECON - Economía, Econometría y Finanzas ENER - Energía ENGI - Ingeniería ENVI - Ciencias ambientales HEAL - Profesiones de la salud IMMU - Inmunología y Microbiología MATE - Ciencia de los materiales MATEMÁTICAS - Matemáticas MEDI - Medicina

Parámetro	Tipo	Descripción
		NEUR - Neurociencia NURS - Enfermería PHAR - Farmacología, Toxicología y Farmacia PHYS - Física y Astronomía PSYC - Psicología SOCI - Ciencias Sociales VETE - Veterinario MULT - Multidisciplinario

Tabla 1.10: API-Elsevier: Métodos Scopus API
Fuente: [34]

Vistas de búsqueda de Scopus

En la respuesta del listado de datos consultados se podrá seleccionar específicamente las columnas que deseamos obtener; así, reduciendo el consumo de datos en la respuesta (Tabla 1.11).

Campo	Descripción	STANDARD	COMPLETE
link ref=self	URI de la API de recuperación de resumen de contenido	X	X
link ref=scopus	URL de la página de detalles del resumen de Scopus	X	X
link ref=scopus-citedby	Scopus Citado por URL de resultados	X	X
prism:url	URI de la API de recuperación de resumen de contenido	X	X
dc:identifier	ID de Scopus	X	X
eid	ID Electrónico	X	X
dc:title	Título del artículo	X	X
prism:aggregationType	Tipo de fuente	X	X

Campo	Descripción	STANDARD	COMPLETE
subtype	Código de tipo de documento	X	X
subtypeDescription	Descripción del tipo de documento	X	X
citedby-count	Recuento de citas	X	X
prism:publicationName	Título de la fuente	X	X
prism:isbn	Identificador de origen	X	X
prism:issn	Identificador de origen	X	X
prism:volume	Volumen	X	X
prism:issueIdentifier	Problema	X	X
prism:pageRange	Página	X	X
prism:coverDate	Fecha de publicación (YYYY-MM-DD)	X	X
prism:coverDisplayDate	Fecha de publicación (texto original)	X	X
prism:doi	Identificador de objeto de documento	X	X
pii	Identificador de artículo de publicación	X	X
pubmed-id	Identificador de MEDLINE	X	X
orcid	ORCID	X	X
dc:creator	Primer autor (primera entrada de autenticación)	X	X
openaccess	Estado de acceso abierto	X	X

Campo	Descripción	STANDARD	COMPLETE
openaccessFlag			
affiliation affilname	Nombre de afiliación	X	X
affiliation affiliation-city	Ciudad de afiliación	X	X
affiliation affiliation-country	País de afiliación	X	X
affiliation afid	ID de afiliación		X
affiliation affiliation-url	URI de API de recuperación de afiliación de contenido que hace referencia al perfil de afiliación		X
affiliation name-variant	Nombre alternativo de afiliación		X
author author-url authid orcid authname given-name surname initials afid	Lista completa de autores (incluye ID de autor) La URL del autor contiene el URI de la API de recuperación de autor de contenido que hace referencia al perfil del autor.		X
dc:description	Resumen		X
authkeywords	Palabras clave del autor		X
article-number	Número de artículo		X
fund-acr	Acrónimo de la agencia financiadora		X

Campo	Descripción	STANDARD	COMPLETE
fund-no	Identificación de la agencia financiadora		X
fund-sponsor	Nombre de la agencia financiadora		X

Tabla 1.11: API-Elsevier: Vistas de búsqueda de Scopus
Fuente: [35]

API de búsqueda de autor

Son los parámetros que pide el API para tomarlos y enviarlos a consultar en su base de datos y así obtener información acerca de un autor (Tabla 1.12).

Parámetro	Tipo	Descripción
apiKey	xsd:string	Anulación para el encabezado HTTP X-ELS-APIKey, esto representa una clave de desarrollador de aplicaciones única que proporciona acceso a los recursos.
query	xsd:string	Esto representa la búsqueda booleana que se ejecutará en el grupo Autor. Hay información adicional sobre los consejos de búsqueda. ej. query=afil(universidad)
view	xsd:string defecto: STANDARD options: STANDARD, COMPLETE	Este alias representa la lista de elementos que se devolverán en la respuesta. La Tabla 1.13 muestra las vistas de búsqueda de autor.
field	xsd:string	Este alias representa el nombre de campos específicos que deben devolverse. La lista de campos incluye todos los campos devueltos en la carga útil de respuesta (ver Tabla 1.13). Se pueden especificar varios campos, delimitados por comas. Tenga en cuenta que especificar este parámetro anula el parámetro de vista. ej. field=url,identifier,description
start	xsd:string	Valor numérico que representa el desplazamiento de los resultados (es decir, la posición de inicio de los resultados de búsqueda). El máximo para este valor es un valor predeterminado a nivel de

Parámetro	Tipo	Descripción
		sistema (varía con el grupo de búsqueda) menos el número de resultados solicitados. Si no se especifica, el desplazamiento se establecerá en cero (es decir, primer resultado de búsqueda) ex. start=5
count	xsd:string	Valor numérico que representa el número máximo de resultados que se devolverán para la búsqueda. Si no se proporciona, se establecerá en un sistema predeterminado según el nivel de servicio. Además, el número no puede exceder el valor máximo predeterminado del sistema; si lo hace, se devolverá un error. ex. count=10

Tabla 1.12: API-Elsevier: Métodos Autor API
Fuente: [36]

Vistas de búsqueda de autor

De la misma manera, se podrá seleccionar específicamente cuáles serán las columnas que se utilizarán, reduciendo el consumo de datos en la petición (Tabla 1.13).

Campo	Descripción	STANDARD
link ref=self	URI de API de recuperación de autor de contenido	X
link ref=scopus-author	URL de detalles del autor de Scopus, no habilitada para PREVIEW	X
link ref=scopus-citedby	Autor de Scopus citado por URL de resultados	X
link ref=search	Contenido URL de la API de búsqueda de Scopus para los resúmenes asociados con el autor (es decir, una búsqueda de documentos de Scopus por ID de autor) http://api.elsevier.com/content/search/scopus?query=authid({authid})	X
prism:url	URI de API de recuperación de contenido del autor	X
dc:identifier	ID de autor	X
eid	ID Electrónico	X
orcid	Identificador del autor	X

Campo	Descripción	STANDARD
document-count	Numero de documentos	X
subject-area	Áreas temáticas (Máximo de 3)	X
preferred-name-surname	Apellido del autor preferido	X
preferred-name-given-name	Nombre del autor preferido	X
preferred-name-initials	Iniciales del autor	X
name-variant	Variantes de nombre de autor (Máximo de 3)	X
affiliation-current-affiliation-name	Nombre de afiliaciones actuales	X
affiliation-current-affiliation-city	Ciudad	X
affiliation-current-affiliation-country	País	X
affiliation-current-affiliation-id	ID de afiliación	X
affiliation-current-affiliation-url	URI de API de recuperación de contenido de afiliación	X

*Tabla 1.13: API-Elsevier: Vistas de búsqueda de Autor
Fuente: [37]*

1.2.2.10 Escala de usabilidad del sistema

La escala de usabilidad del sistema (SUS) proporciona una herramienta que servirá para medir la usabilidad de un programa. El cual consiste en un cuestionario de 10 preguntas; las cuales tendrán cinco opciones de respuesta, las mismas que irán desde totalmente de acuerdo hasta totalmente en desacuerdo [38].

Beneficios de usar un SUS

Como nos muestra [38], al tener múltiples referencias en artículos y publicaciones. SUS se ha convertido en un estándar en la industria, pudiéndonos describir beneficios observados en su uso, tales como:

- Una escala muy fácil de administrar a los participantes
- Se puede utilizar en tamaños de muestra pequeños con resultados fiables
- Es válido: Puede diferenciar eficazmente entre sistemas utilizables e inútiles

Consideraciones al utilizar un SUS

Las consideraciones que [38] nos indica en el momento de su uso son las siguientes:

- El sistema de puntuación es algo complejo
- Hay una tentación cuando se fija en las puntuaciones, ya que están en una escala de 0-100, que se los puede interpretar como porcentajes, pero no lo son
- La mejor manera de interpretar sus resultados consiste en "normalizar" las puntuaciones para producir una clasificación de percentiles
- SUS no es diagnóstico - su uso es en la clasificación de la facilidad de uso del sitio, aplicación o entorno que se está probando

Interpretar partituras

Acorde a lo descrito por [38] y [39] se logró interpretar de la siguiente manera la calificación de las evaluaciones:

Se sumará los resultados promediados de los cuestionarios realizados, tendiendo como consideración lo siguiente: las preguntas impares tomarán el valor asignado por el usuario y se le restará uno y para las preguntas pares, será de cinco menos el valor asignado por el entrevistado; una vez con la sumatoria de los resultados de cada pregunta se multiplicará por 2.5. Con este resultado promediado se podrá establecer el nivel de usabilidad del sistema considerando que si se tiene una puntuación por encima de 68 se consideraría por encima de la media y cualquier cosa por debajo de 68 está por debajo de la media.

1.3 Objetivos

1.3.1 General

- Desarrollar un sistema de recomendación en plataforma móvil para facilitar el procesamiento de bases de datos de publicaciones científicas.

1.3.2 Específicos

- Identificar las tecnologías para el acceso y procesamiento de la información contenida en bases de datos de publicaciones científicas.
- Implementar un mecanismo para la recomendación de publicaciones científicas y espacios de publicación científica basado en temas de interés del usuario.
- Evaluar el impacto del sistema de recomendación implementado.

CAPÍTULO II.-

METODOLOGÍA

2.1 Materiales

Para la recolección de información se realizó una entrevista con el tutor del proyecto que a su vez es docente investigador en la cual se trató sobre el proceso realizado para buscar información relevante basado en los términos de interés del usuario en las bibliotecas virtuales y tener un punto de partida para el inicio de su investigación. Una vez terminado su artículo científico procede a buscar revistas en base a las áreas pertenecientes; para luego investigar en Scimago Journal & Country Rank los cuartiles, SJR y sus citas y así decidir la mejor opción para su publicación.

Para la evaluación del impacto del programa se realizó una encuesta de escala de usabilidad del sistema, basado en [38], de la siguiente manera:

1. Creo que me gustaría utilizar este sistema con frecuencia.
2. Encontré el sistema innecesariamente complejo.
3. Pensé que el sistema era fácil de usar.
4. Creo que necesitaría el apoyo de una persona con conocimientos técnicos para poder utilizar este sistema.
5. Encontré que las diversas funciones de este sistema estaban bien integradas.
6. Pensé que había demasiada incoherencia en este sistema.
7. Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.
8. Me pareció que el sistema es muy difícil de usar.
9. Me sentí muy seguro de usar el sistema.
10. Necesitaba aprender muchas cosas antes de poder seguir este sistema.

Con el objetivo de evaluar la opinión y tener resultados cerrados se optó por usar la escala de Likert con las siguientes opciones:

- Totalmente de acuerdo
- Algo de acuerdo
- Ni de acuerdo ni en desacuerdo
- Algo en desacuerdo
- Totalmente en desacuerdo

2.2 Métodos

2.2.1 Modalidad básica de la investigación

Investigación de campo: La investigación será de campo debido a que se podrá observar el procedimiento realizado por el investigador cuando realiza una consulta de un artículo o revista científica y de acuerdo a los resultados, tomar bases para los inicios de su investigación o seleccionar una revista que le resulte favorable para la divulgación de su artículo.

Investigación bibliográfica: La investigación será bibliográfica debido a que se tomarán como fuentes documentos, artículos, libros, revistas y otras fuentes en internet; sobre los procesos realizados por los investigadores para la publicación de artículos científicos.

2.2.2 Metodología seleccionada

La metodología utilizada para el presente proyecto fue Mobile-D, ya que cada proceso de desarrollo a realizar fue corto y se hizo más énfasis en la investigación de APIs de búsqueda de artículos científicos facilitadas por varias bibliotecas virtuales y páginas a las cuales se pudo realizar web scraping para luego crear un servicio el cual brinde información filtrada al aplicativo móvil. Esta decisión se sustenta en la comparativa de metodologías de desarrollo llevada a cabo, que se documenta en la sección 1.2.2.2.

2.2.3 Framework seleccionado

Se seleccionó el framework de Flutter debido a que nos ofrece la posibilidad de realizar *hot reload* (ver los efectos reflejados en algún cambio de forma inmediata, sin tener que recompilar la aplicación y perder el contexto en el que se encontraba). Además, nos permite realizar cambios y validaciones rápidamente (tanto en lo visual como en la parte lógica) debido a la conservación de los datos. Todo esto se documenta en la sección 1.2.2.3 y muestra la popularidad del aumento en el uso del framework en la sección 1.2.2.6.

2.2.4 Lenguaje seleccionado para el análisis de datos

Para obtener datos que no están disponibles por medio de APIs, sino a través de páginas web, se utilizó web scraping implementado con el paquete BeautifulSoup en lenguaje Python. Dicho paquete permitió extraer la información del html, y unificar los resultados con las APIs que sí están disponibles. Del mismo modo, se decidió utilizar Flask para, a partir de esta información, generar una nueva API que facilita el acceso a los datos analizados/filtrados, tal y como se documenta en la sección 1.2.2.4.

2.2.5 Metodología de desarrollo del proyecto

Siguiendo las fases detalladas por la metodología Mobile-D se muestra a continuación los procesos establecidos para la realización del proyecto:

- Fase de exploración
 - Levantamiento de requerimientos
 - Establecer grupo de interés
 - Planificación del proyecto
 - Búsqueda de bibliotecas virtuales que ofrezcan APIs de consulta de artículos científicos
 - Identificación de páginas habilitadas para realizar web scraping.
- Fase de inicialización
 - Capacitación de personal en el uso de las APIs encontradas
 - Optimización en el envío de consultas a las APIs obteniendo únicamente la información necesaria
 - Establecer los datos a utilizar en las páginas web a realizar web scraping
 - Creación del servicio de consulta
 - Planificación de la arquitectura
 - Configuración de proyecto móvil con paquetes necesarios
- Fase de producción
 - Iteración 1.- Navigation Bar
 - Iteración 2.- Búsqueda de artículos
 - Iteración 3.- Búsqueda de revistas
 - Iteración 4.- Configuración de parámetros
 - Iteración 5.- Consulta de revista
 - Iteración 6.- Búsqueda de artículos por issn o eissn
- Fase de estabilización
 - Generar un archivo en la aplicación de ruta general.
 - Establecer un parámetro en el servidor para cambiar de dirección.
- Fase de prueba y reparación del sistema
 - Prueba del producto integrado y terminado con los requisitos del cliente.
 - Eliminar defectos encontrados en la aplicación.

2.2.6 Población y muestra

Población: En el presente proyecto se ha tomado como caso de estudio/aplicación de los resultados a la comunidad científica de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. En este escenario, se considera que la población son los 64 docentes-investigadores que laboran en dicha entidad, en las carreras de Sistemas, Electrónica, Industrial, Tecnologías de la Información, Telecomunicaciones.

Según Nielsen [40] al contar con expertos en el tema es adecuado tomar una muestra entre 5 y 11; si la muestra es mayor, se puede generar problemas en la validación.

Muestra: Para este caso, con el fin de obtener el nivel de aceptación del aplicativo, se toma de manera representativa como muestra lo siguiente:

Descripción de la muestra	Cantidad	Porcentaje
Docentes de la FISEI	10	100%
Total	10	100%

*Tabla 2.1: Muestra
Fuente: Elaboración propia*

2.2.7 Uso de SUS para evaluar el impacto del sistema de recomendación

Al tratarse de una muestra de 10 docentes se medirá la usabilidad de la aplicación utilizando el instrumento descrito en la sección 1.2.2.10. Esta evaluación nos dará resultados fiables con tamaños de muestra pequeños para diferenciar eficientemente si el sistema es funcional, necesita agregarse más utilidades o es ineficaz.

CAPÍTULO III.-

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

3.1.1 Aplicación de metodología Mobile-D

Para la aplicación de Mobile-D en la creación del proyecto es necesario destacar, esta metodología utiliza la fase de inicialización para planificar y establecer las fases siguientes, procesos y sobre todo a la capacitación del personal. Para la realización de este proyecto se utilizó la fase mencionada anteriormente para enfocarse en la lectura de la documentación y entender el uso de las APIs para realizar consultar a las bases de datos de las bibliotecas virtuales para su posterior uso en la aplicación móvil integradora de registros de publicaciones científicas.

3.1.2 Fase de exploración

3.1.2.1 Levantamiento de requerimientos.

Tomando en cuenta el proceso de búsqueda de publicaciones científicas, establecido por el docente investigador (entrevista documentada en la sección 2.1), se estableció un listado de requerimientos para el desarrollo del aplicativo. A continuación, se listan estos requisitos:

- Consultar artículos científicos según término de interés y áreas de búsqueda específicas.
- Mostrar datos preliminares del artículo como: título, biblioteca, doi, tipo de publicación y contenido.
- Consultar revistas científicas según término de interés y áreas de búsqueda específicas.
- Mostrar datos preliminares de la revista como: título, biblioteca, tipo de publicación, número de artículos agrupados en la búsqueda, issn y eissn.
- Almacenar las preferencias del usuario como: término por defecto, número de artículos a buscar y agrupar, área de búsqueda para su uso en las consultas a las bases de datos de cada biblioteca virtual.
- Mostrar cuartiles, SJR y citas por revista.
- Redirigir a la página de la biblioteca virtual con el artículo/revista científica donde se encuentre alojado.

3.1.2.2 Establecer grupo de interés.

Se estableció como grupo de interés a los docentes investigadores que realizan el proceso de búsqueda y creación de artículos científicos para su divulgación.

3.1.2.3 Planificación del proyecto.

Definición de usuario

Tipo de usuario	Tareas	Módulos
Usuario	<ul style="list-style-type: none">• Búsqueda de artículos científicos por término de interés o revista seleccionada.• Búsqueda de revistas por término de interés.• Búsqueda de cuartiles, SJR y citaciones de revista seleccionada.• Redirigir a biblioteca virtual del artículo/ revista científica.• Definir y almacenar parámetros de búsqueda como:<ul style="list-style-type: none">○ Término de interés por defecto○ Cantidad de artículos a agrupar para buscar revistas.○ Áreas de búsqueda de artículos/revistas.	<ul style="list-style-type: none">• Artículos• Revistas• Revista• Parámetros

Tabla 3.1: Definición de usuario
Fuente: Elaboración propia

Selección de entorno

- Postman para obtener la información en bruto de APIs de bibliotecas virtuales y planificar el uso de los campos a utilizar.
- Uso de Python para el consumo de API-REST de las bases de datos de las bibliotecas virtuales, BeautifulSoup para realizar web scraping y obtener información de Scimago Journal & Country Rank y creación de API-REST en Flask para mostrar los datos analizados/filtrados.
- Flutter para el desarrollo del aplicativo móvil multiplataforma.
- Laptop Asus, procesador Intel Core i7 con 16GB RAM.
- Celular Xiaomi MI 8 Lite con Android.
- Android Studio para la creación de máquinas virtuales.

3.1.2.4 Búsqueda de bibliotecas virtuales que ofrezcan APIs de consulta de artículos científicos.

Para identificar las tecnologías para el acceso y procesamiento de la información contenida en bases de datos de publicaciones científicas se ingresó en los sitios web de las bibliotecas virtuales de Elsevier, Scimago Journal & Country Rank, Springer, Proquest, Lexis Finder, Wiley Online Library, Digitalia, Taylor & Francis y Latindex para ver si ofrecían APIs para realizar consultas de revistas y artículos científicos, pero muchos de ellos tienen su información privada, además solo se puede realizar búsquedas solo desde su plataforma y en otros casos se debe tener una suscripción activa para realizar consultas.

Las bibliotecas virtuales encontradas que brindan sus datos desde API y tienen documentación clara, bien estructurada y para uso público fueron:

- Elsevier (https://dev.elsevier.com/api_docs.html)
- Springer (<https://dev.springernature.com/restfuloperations>)

Cabe recalcar que estas bibliotecas ofrecieron datos open-access y privados, teniendo como diferencia la restricción en los resultados expuestos dependiendo su caso.

3.1.2.5 Identificación de páginas habilitadas para realizar web scraping.

Se tomó las bibliotecas virtuales anteriormente mencionadas para observar si se podía recabar datos u obtener datos adicionales a los ya adquiridos a través de API (Elsevier y Springer) por su restricción de datos y estos se los puede visualizar desde su plataforma.

Estableciendo el código de la Figura 3.1 se pudo observar a las páginas que se podría realizar web scraping.

```
import urllib
from bs4 import BeautifulSoup
pagina_web =
'https://www.scimagojr.com/journalsearch.php?q=66411&tip=sid&clean=0
'
codigo = urllib.request.urlopen(pagina_web)
bea = BeautifulSoup(codigo)
print(bea)
```

Figura 3.1: Código para realizar web scraping

Fuente: Elaboración propia

Los resultados obtenidos fueron los siguientes:

- Respuesta *Error 403: Forbidden*. Esto ocurre debido a que los sitios tienen activos bots que detectan y bloquean a otros bots (creado con web scraping) debido a que lo consideran malicioso y bloquea el acceso al sitio. Por lo tanto, antes de realizar este proceso en cada sitio es recomendable leer atentamente los términos y condiciones de uso del servicio debido a que podrían tomar medidas legales por utilizar bots porque se considera piratería.

```

-----
HTTPError Traceback (most recent call last)
<ipython-input-1-0dc7b02255c7> in <module>
     9 #pagina_web = 'https://www.acm.org/search#stq=artificial%20intelligence&stp=1'
    10 pagina_web = 'https://www.tandfonline.com/action/doSearch?AllField=artificial+in
telligence'
--> 11 codigo = urllib.request.urlopen(pagina_web)
     12 bea = BeautifulSoup(codigo)
     13 print(bea)

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, ca
file, capath, cadefault, context)
     220     else:
     221         opener = _opener
--> 222     return opener.open(url, data, timeout)
     223
     224 def install_opener(opener):

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in open(self, fullurl, data, time
out)
     529         for processor in self.process_response.get(protocol, []):
     530             meth = getattr(processor, meth_name)
--> 531             response = meth(req, response)
     532
     533         return response

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in http_response(self, request, r
esponse)
     639         if not (200 <= code < 300):
     640             response = self.parent.error(
--> 641                 'http', request, response, code, msg, hdrs)
     642
     643         return response

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in error(self, proto, *args)
     567         if http_err:
     568             args = (dict, 'default', 'http_error_default') + orig_args
--> 569         return self._call_chain(*args)
     570
     571 # XXX probably also want an abstract factory that knows when it makes

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in _call_chain(self, chain, kind,
meth_name, *args)
     501         for handler in handlers:
     502             func = getattr(handler, meth_name)
--> 503             result = func(*args)
     504             if result is not None:
     505                 return result

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in http_error_default(self, req,
fp, code, msg, hdrs)
     647 class HTTPDefaultErrorHandler(BaseHandler):
     648     def http_error_default(self, req, fp, code, msg, hdrs):
--> 649         raise HTTPError(req.full_url, code, msg, hdrs, fp)
     650
     651 class HTTPRedirectHandler(BaseHandler):

HTTPError: HTTP Error 403: Forbidden

```

Figura 3.2: Web scraping- HTTP Error 403: Forbidden

Fuente: Elaboración propia

- Respuesta HTTP Error 401: Unauthorized. Se debe a que pide una autenticación de usuario para poder acceder a la página y consumir la información.

```

-----
HTTPError                                Traceback (most recent call last)
<ipython-input-3-e940d3ccb063> in <module>
    10 #pagina_web = 'https://www.tandfonline.com/action/doSearch?AllField=artificial+i
ntelligence'
    11 pagina_web = 'https://www.scopus.com/sourceid/24141'
--> 12 codigo = urllib.request.urlopen(pagina_web)
    13 bea = BeautifulSoup(codigo)
    14 print(bea)

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in urlopen(url, data, timeout, ca
file, capath, cadefault, context)
    220     else:
    221         opener = _opener
--> 222     return opener.open(url, data, timeout)
    223
    224 def install_opener(opener):

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in open(self, fullurl, data, time
out)
    529         for processor in self.process_response.get(protocol, []):
    530             meth = getattr(processor, meth_name)
--> 531             response = meth(req, response)
    532
    533     return response

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in http_response(self, request, r
esponse)
    639         if not (200 <= code < 300):
    640             response = self.parent.error(
--> 641                 'http', request, response, code, msg, hdrs)
    642
    643     return response

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in error(self, proto, *args)
    567         if http_err:
    568             args = (dict, 'default', 'http_error_default') + orig_args
--> 569             return self._call_chain(*args)
    570
    571 # XXX probably also want an abstract factory that knows when it makes

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in _call_chain(self, chain, kind,
meth_name, *args)
    501         for handler in handlers:
    502             func = getattr(handler, meth_name)
--> 503             result = func(*args)
    504             if result is not None:
    505                 return result

D:\Program Files (x86)\Anaconda3\lib\urllib\request.py in http_error_default(self, req,
fp, code, msg, hdrs)
    647 class HTTPDefaultErrorHandler(BaseHandler):
    648     def http_error_default(self, req, fp, code, msg, hdrs):
--> 649         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    650
    651 class HTTPRedirectHandler(BaseHandler):

HTTPError: HTTP Error 401: Unauthorized

```

Figura 3.3: Web scraping- HTTP Error 401: Unauthorized
Fuente: Elaboración propia

- El sitio web coloca los datos dinámicamente

```

<script src="https://cdn.elsevier.com/nuxt/c9a528718228a07ee772.js" /></script><script defer=" src=" https://search-app.prod.elsevier.com/nuxt/1a134f5b193bba7cd9c6.js"></script>
<script type="text/javascript">
  //
    _uzactfeed = window._uzactfeed || [];
    _uzactfeed.push(['_setID', '6820BED4EFFBE911AA5EC9768AD49726']);
    _uzactfeed.push(['_setsID', '47C4AECEFFBE911AA5EC9768AD49726']);
    _uzactfeed.push(['_start']);
    (function() {
      var uz = document.createElement('script'); uz.type = 'text/javascript'; uz.async = true; uz.charset = 'utf-8';
      uz.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + 'cdn4.userzoom.com/feedback/js/uz_feed.js?cuid=8890341873DAE311BEDA0022196C4538';
      var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(uz, s);
    })();
  //]]&gt;
&lt;/script&gt;
&lt;div class="fixed-side-links side-bottom scroll-down side-fixed"&gt;
&lt;div class="row row-back-to-top"&gt;
&lt;div class="columns large-12"&gt;

&lt;!-- JSON Breadcrumbs End --&gt;
&lt;script class="cookie-category-4" type="text/plain"&gt;
  (function() {
    var uz = document.createElement('script'); uz.type = 'text/javascript'; uz.async = true; uz.charset = 'utf-8';
    uz.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + 'cdn4.userzoom.com/files/js/QzczN1Qx.js?t=uz_til&amp;cuid=8890341873DAE311BEDA0022196C4538';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(uz, s);
  })();
&lt;/script&gt;
&lt;script src="https://cdn.elsevier.io/matrix/ui/9.2.4/scripts-bottom.js"&gt;&lt;/script&gt;
&lt;script&gt;
  // WSYIWYG button text fix
  document.querySelectorAll('.embedded-content-cookie-message').forEach(function (message) {
    if (!message.querySelector('#ot-sdk-btn')) {
      const p = message.querySelector('p');
      if (p) {
        p.innerHTML = p.innerHTML + ' cookie settings using the link in the footer.';
      }
    }
  });
&lt;/script&gt;
</pre>
</div>
<div data-bbox="395 461 673 476" data-label="Caption">
<p>Figura 3.4: Web scraping- Datos dinámicos</p>
</div>
<div data-bbox="440 475 619 489" data-label="Text">
<p>Fuente: Elaboración propia</p>
</div>
<div data-bbox="187 491 573 508" data-label="Section-Header">
<ul>
<li>• Datos del sitio web correctos para su uso</li>
</ul>
</div>
<div data-bbox="187 514 907 859" data-label="Code-Block">
<pre>
&lt;div class="cellcontent"&gt;
&lt;div class="cellslide" id="svgquartiles"&gt;
&lt;/div&gt;
&lt;div class="cellslide"&gt;
&lt;p&gt;
  The set of journals have been ranked according to their SJR and divided into four equal groups, four quartiles. Q1 (green) comprises the quarter of the journals with the highest values, Q2 (yellow) the second highest values, Q3 (orange) the third highest values and Q4 (red) the lowest values.
&lt;/p&gt;
&lt;table&gt;&lt;thead&gt;&lt;tr&gt;&lt;th&gt;Category&lt;/th&gt;&lt;th&gt;Year&lt;/th&gt;&lt;th&gt;Quartile&lt;/th&gt;&lt;/tr&gt;&lt;/thead&gt;&lt;tbody&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;1999&lt;/td&gt;&lt;td&gt;Q4&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;2000&lt;/td&gt;&lt;td&gt;Q4&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;2001&lt;/td&gt;&lt;td&gt;Q4&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;2002&lt;/td&gt;&lt;td&gt;Q4&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;2003&lt;/td&gt;&lt;td&gt;Q4&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;2004&lt;/td&gt;&lt;td&gt;Q4&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;Earth and Planetary Sciences (miscellaneous)&lt;/td&gt;&lt;td&gt;2005&lt;/td&gt;&lt;td&gt;Q3&lt;/td&gt;&lt;/tr&gt;
&lt;/tbody&gt;&lt;/table&gt; &lt;/div&gt;
...
&lt;div class="cellslide"&gt;
&lt;p&gt;
  The SJR is a size-independent prestige indicator that ranks journals by their 'average prestige per article'. It is based on the idea that 'all citations are not created equal'. SJR is a measure of scientific influence of journals that accounts for both the number of citations received by a journal and the importance or prestige of the journals where such citations come from. It measures the scientific influence of the average article in a journal, it expresses how central to the global scientific discussion an average article of the journal is.
&lt;/p&gt;
&lt;table&gt;&lt;thead&gt;&lt;tr&gt;&lt;th&gt;Year&lt;/th&gt;&lt;th&gt;SJR&lt;/th&gt;&lt;/tr&gt;&lt;/thead&gt;&lt;tbody&gt;
&lt;tr&gt;&lt;td&gt;1999&lt;/td&gt;&lt;td&gt;0.104&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;2000&lt;/td&gt;&lt;td&gt;0.100&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;2001&lt;/td&gt;&lt;td&gt;0.103&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;2002&lt;/td&gt;&lt;td&gt;0.103&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;2003&lt;/td&gt;&lt;td&gt;0.103&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;2004&lt;/td&gt;&lt;td&gt;0.107&lt;/td&gt;&lt;/tr&gt;
&lt;tr&gt;&lt;td&gt;2005&lt;/td&gt;&lt;td&gt;0.122&lt;/td&gt;&lt;/tr&gt;
&lt;/tbody&gt;&lt;/table&gt; &lt;/div&gt;
&lt;/div&gt;
</pre>
</div>
<div data-bbox="398 862 670 878" data-label="Caption">
<p>Figura 3.5: Web scraping- Datos correctos</p>
</div>
<div data-bbox="440 876 619 892" data-label="Text">
<p>Fuente: Elaboración propia</p>
</div>
<div data-bbox="518 896 547 914" data-label="Page-Footer">
<p>51</p>
</div>
```


Al no contar con un API para su consumo, el sitio en el cual se necesitó el uso de web scraping es Scimago Journal & Country Rank.

3.1.3 Fase de iniciación

3.1.3.1 Capacitación de personal en el uso de las APIs encontradas.

Con el fin de entender y realizar un correcto uso de las APIs encontradas se estableció un tiempo en el cual el personal pudo practicar sobre el uso y la forma de respuesta en cada una de las bibliotecas detalladas a continuación, para así establecer la forma en la cual se manipularán los datos y serán enviadas a la aplicación integradora.

Springer

Para obtener un *api_key* es necesario registrarse en la página de <https://dev.springernature.com>, siguiendo los pasos y rellenando campos requeridos; una vez adquirido se tiene lo necesario para realizar consultas al API de Springer.

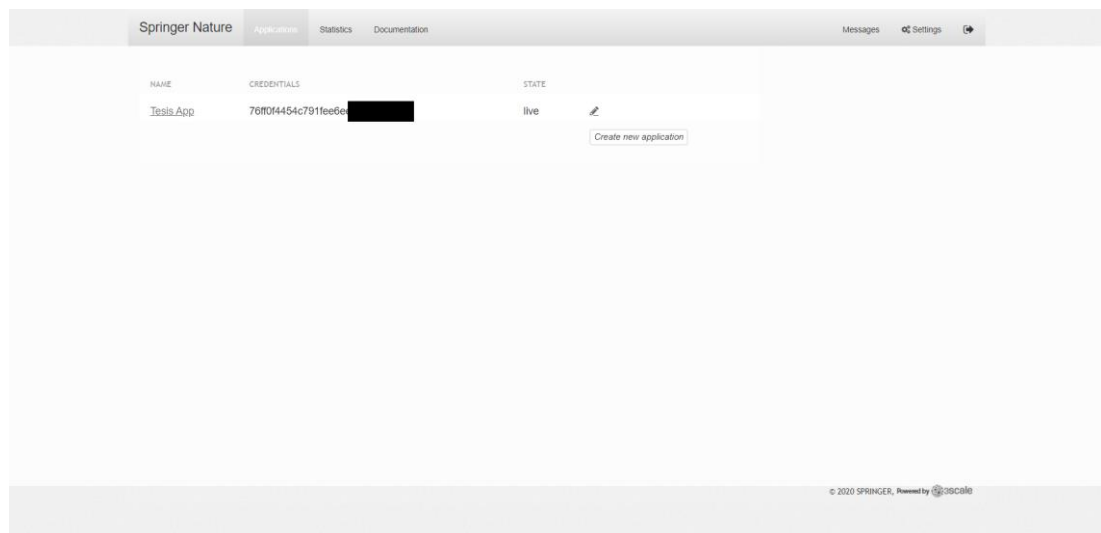


Figura 3.6: Obtención de api key en Springer

Fuente: Elaboración propia

Se realizaron pruebas con las colecciones *meta* y *openaccess*, generando una consulta compleja por palabra clave (*keyword*) y un área de búsqueda (*subject*).

Según la documentación establecida por Springer (sección 1.2.2.8), se tienen las siguientes formas de enviar peticiones.

- Consumo de API por open access

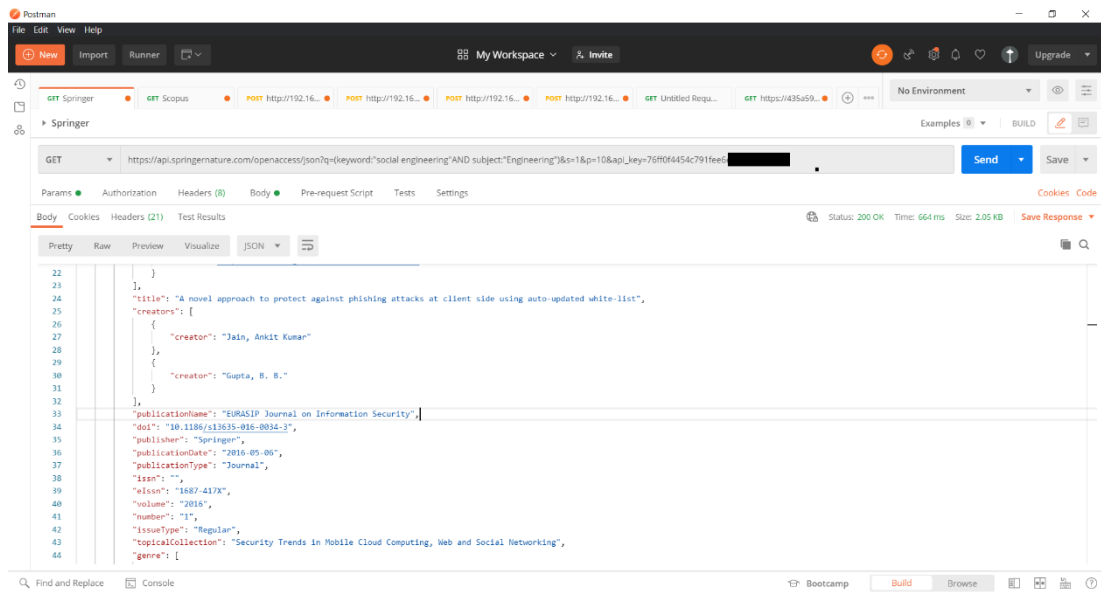


Figura 3.7: Consulta API Springer- Open access
Fuente: Elaboración propia

- Consumo de API por metadata v2

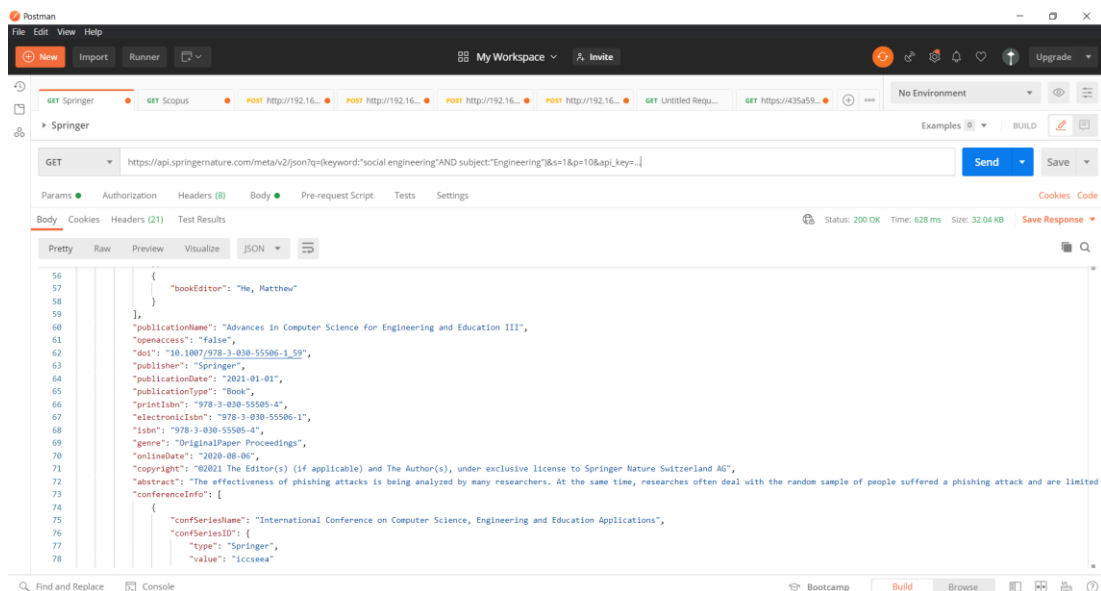


Figura 3.8: Consulta API Springer- Metadata v2
Fuente: Elaboración propia

Realizadas las consultas de la figura 3.7 y figura 3.8 se optó por utilizar la colección de openaccess debido a que se encuentran datos necesarios para el aplicativo, tales como: publisher, doi, issn y eisbn, publicationName, publicationType y contentType.

Elsevier

Para obtener un `api_key` se necesita registrarse en la página de <https://dev.elsevier.com>, siguiendo los pasos y rellenando campos requeridos; una vez adquirido se tendrá lo necesario para realizar consultas al API de Elsevier.

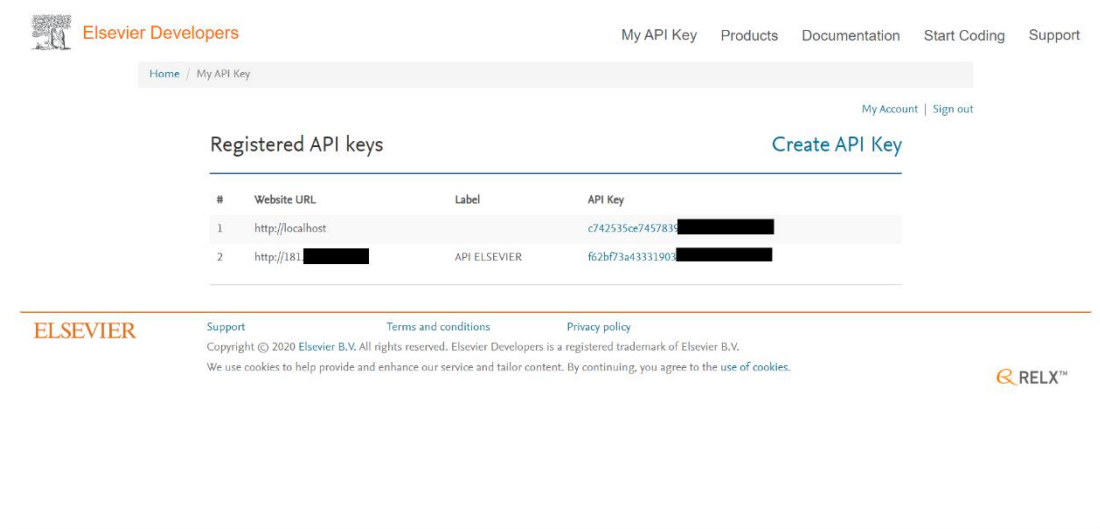


Figura 3.9: Obtención de `api key` en Elsevier

Fuente: Elaboración propia

Tal y como aparece en la documentación establecida por Elsevier (sección 1.2.2.8), es posible realizar consultas hacia las bibliotecas de ScienceDirect y Scopus, definiendo parámetros como la consulta (`query`) y el área de búsqueda (`subj`).

- Consumo de API ScienceDirect V2

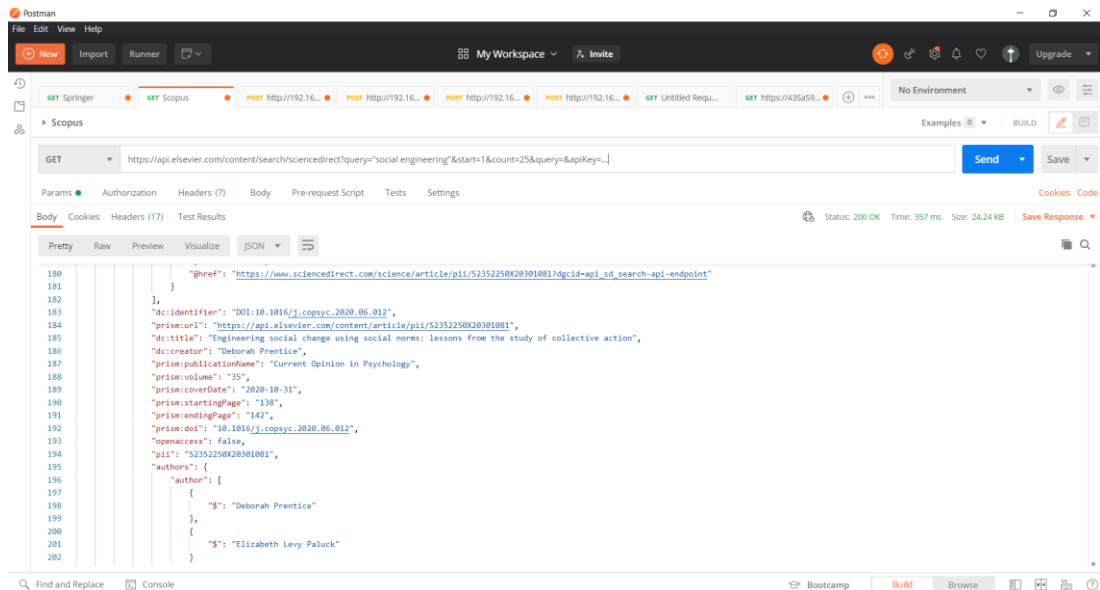


Figura 3.10: Consulta API ScienceDirect

Fuente: Elaboración propia

Resultado consulta a Scopus optimizada:

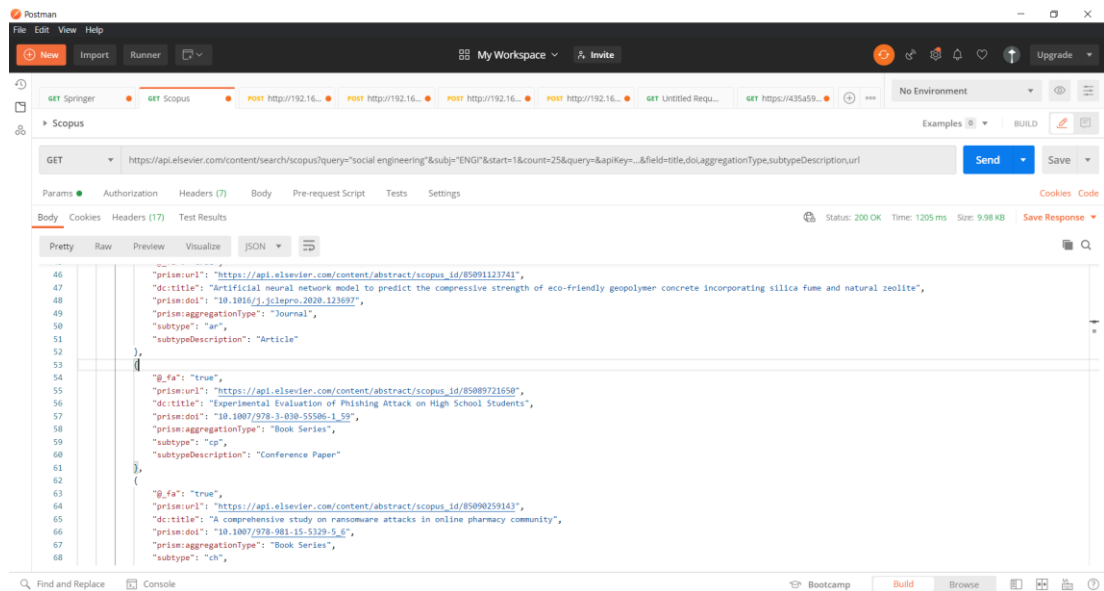


Figura 3.12: Consulta API Scopus (Optimizada)

Fuente: Elaboración propia

Obtención de publisher:

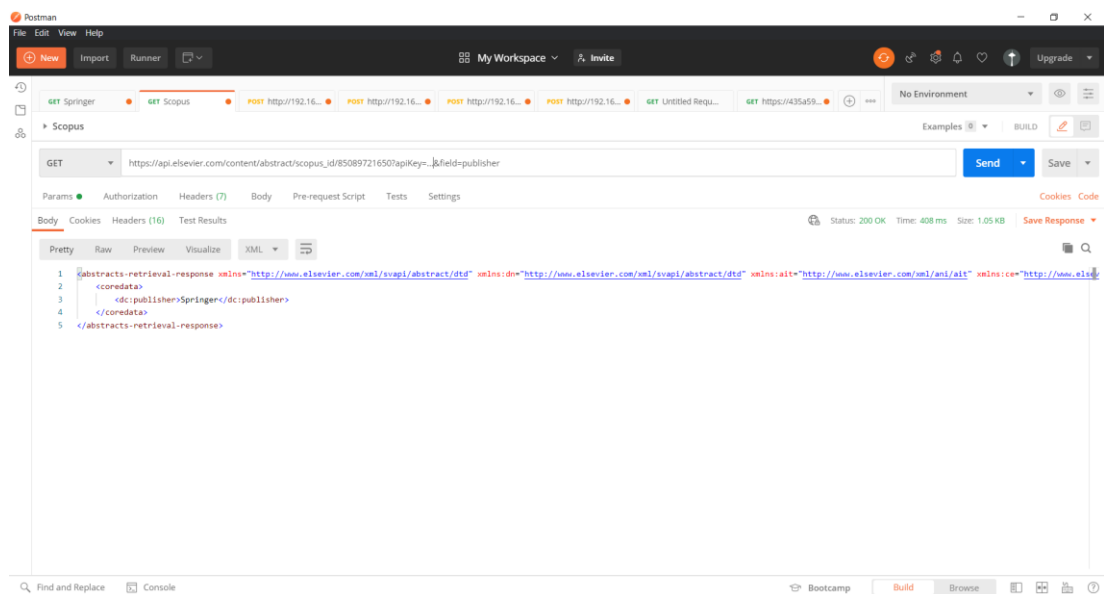


Figura 3.13: Consulta publisher de API Autor

Fuente: Elaboración propia

3.1.3.3 Establecer los datos a utilizar en las páginas web a realizar web scraping.

Url a consultar

<https://www.scimagojr.com/journalsearch.php?q={parámetro de búsqueda}>

(El parámetro de búsqueda será el issn o eissn que será enviado al seleccionar una revista en el aplicativo).

Proceso de consulta

Scimago Journal & Country Rank realiza una búsqueda todo lo relacionado al query enviado y se debe proceder a ingresar a los enlaces de resultados. Al tratarse de issn o eissn el resultado será único; por lo tanto, se deberá ingresar al primer y único enlace, como se detalla en la *Figura 3.14*.

Nota: Se tomó como parámetro principal issn y en el caso de no tener resultados se tomará como parámetro secundario el eissn.

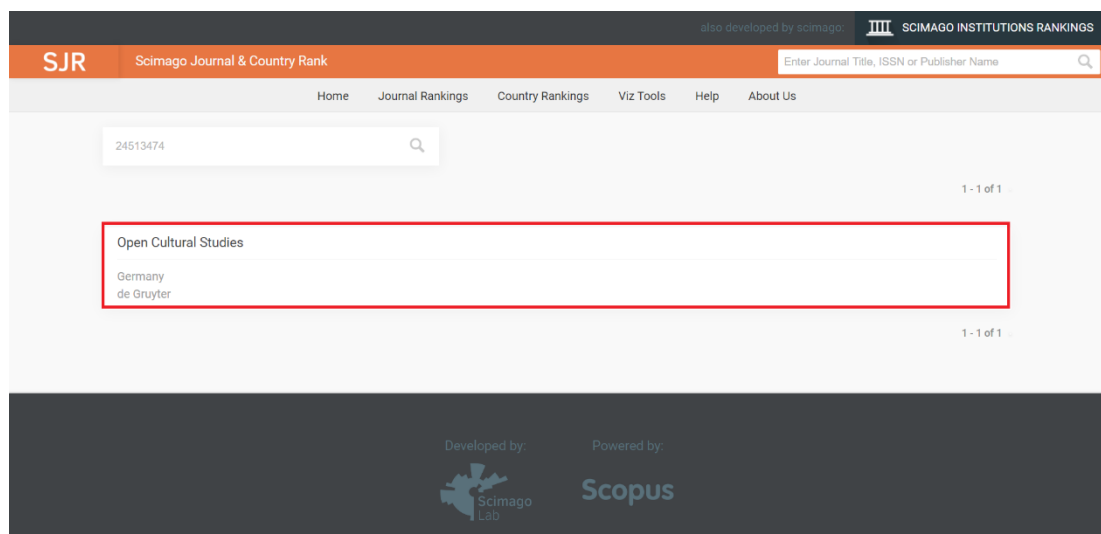


Figura 3.14: Búsqueda de parámetro enviado desde el url
Fuente: Elaboración propia

En la pestaña de la revista, los campos que se requerirán para el uso en la aplicación se muestran seleccionados en la *Figura 3.15*.

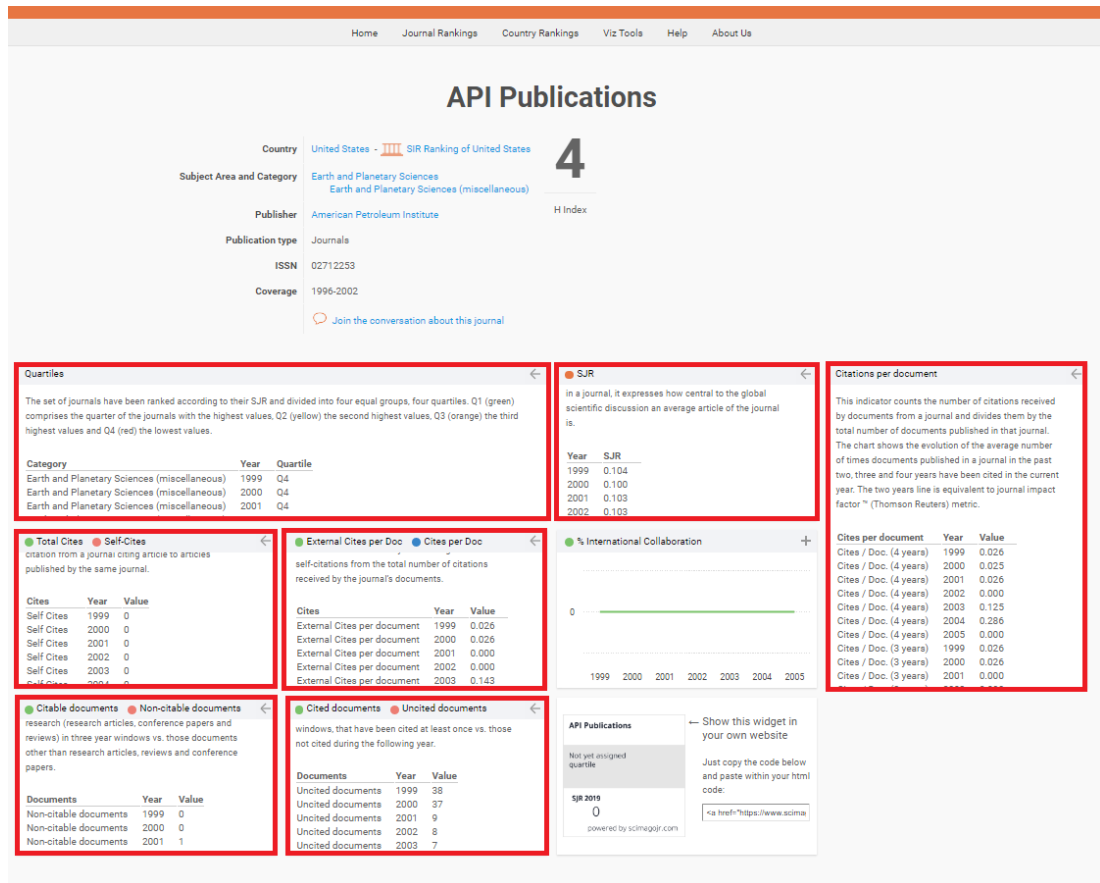


Figura 3.15: Campos requeridos para uso en el aplicativo
Fuente: Elaboración propia

Análisis de la estructura de la página que provee los datos

Un factor que facilitó la toma de datos necesarios fue el hecho de que la página fue desarrollada con buenas prácticas; al revisar la estructura se observó que todos los datos se encontraban en un *div* con clase "cellslide" y dentro tenía *table* (Anexos 1, 2, 3, 4, 5, 6, 7) con lo cual se reduce la carga al servidor al enfocarse solo en esos elementos. De igual manera, esto permite que el tiempo de respuesta sea corto.

A continuación, se muestra una tabla con los resultados de los datos buscados:

Tipo	<div class="cellslide"></div>	<table></table>	<thead></thead>	<tbody></tbody>
Cuartiles	X	X	X	X
SJR	X	X	X	X
Citaciones por documento	X	X	X	X
Total citas	X	X	X	X
Citas externas por documento	X	X	X	X
Documentos citables	X	X	X	X
Documentos citados	X	X	X	X

Tabla 3.2: Tabla de resultados de datos buscados utilizando web scraping
Fuente: Elaboración propia

3.1.3.4 Creación del servicio de consulta

Una vez creada la carpeta del servicio se debe preparar el entorno instalando los componentes a utilizar.

Beautifulsoup

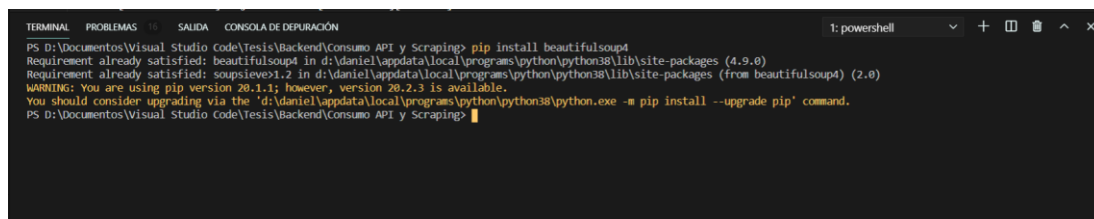


Figura 3.16: Instalación de BeautifulSoup
Fuente: Elaboración propia

Flask



Figura 3.17: Instalación de Flask
Fuente: Elaboración propia

Petición HTTP a bibliotecas virtuales

El mecanismo implementado para la recomendación de publicaciones científicas y espacios de publicación científica basado en temas de interés de usuario serán planteados en el servicio presentado a continuación:

- Realizar peticiones por API-REST a las bibliotecas virtuales (Elsevier y Springer), tomar la respuesta y generar un modelo estándar para colocar los datos requeridos (tanto para consultas de artículos como revistas con sus respectivos parámetros) y enviar la solución a través del servicio por API-REST.
- Realizar web scraping a Scimago Journal & Country Rank, ingresar con los métodos de BeautifulSoup al elemento HTML a extraer y almacenar en arreglos la respuesta para finalmente enviar la solución a través del servicio por API-REST.

Una vez expuesto los procesos a realizar se muestra a continuación el procedimiento que fue implementado:

Modelos

Una vez establecido los datos a utilizar se crearon los modelos, generando un estándar para todo tipo de respuestas consultas por API a las bibliotecas virtuales, mostradas en las *Figuras 3.18 y 3.19*.

Modelo de artículos

```
class Articulos():
    def __init__(self, start, data):
        self.data = data

    def nuevoArticuloElsevier(self, title, doi = None, aggregationType
= None, subtypeDescription = None, urlPublisher = None):
        ...

    def nuevaArticuloSpringer(self, title, doi= None, publicationType
= None, contentType = None, publisher = None):
        ...

    def getArticulosJson(self):
        ...

    def getArticulos(self):
        return self.data

    def getArticulosCount(self):
        ...

    def updateTotalArticulosElsevier(self, totalArticulos):
        ...

    def updateTotalArticulosSpringer(self, totalArticulos):
        ...
```

Figura 3.18: Modelo de artículos

Fuente: Elaboración propia

Modelo de revistas

```
class Revistas():
    def __init__(self, start, data):
        self.data = data

    def nuevaRevistaElsevier(self, publicationName, issn= None,
eissn = None, urlPublisher = None, aggregationType = ''):
        ...

    def nuevaRevistaSpringer(self, publicationName, issn= None,
eissn = None, publisher = None, publicationType = ''):
        ...

    def getRevistasJson(self):
        ...

    def getRevistas(self):
        return self.data

    def getRevistasCount(self):
        ...

    def endElsevier(self):
        ...

    def endSpringer(self):
        ...

    def updateTotalRevistasElsevier(self, totalRevistas):
        ...

    def updateTotalRevistasSpringer(self, totalRevistas):
        ...
```

Figura 3.19: Modelo de revistas

Fuente: Elaboración propia

Para la creación de una nueva revista y artículo de Elsevier se realizó una subconsulta (Figura 3.13) para así cumplir el estándar establecido del modelo. Al contrario de Springer que contaba con todos los datos necesarios por lo que no se necesitó subconsultas.

Controladores

Utilizando las rutas establecidas en las Figuras 3.7 y 3.12 se crearon los controladores para consultar revistas y artículos que recibirán parámetros como: término de interés (*query* en Elsevier y *keyword* en Springer), inicio de resultados de búsqueda (*start* en Elsevier y *s* en Springer), áreas de búsqueda (*subj* en Elsevier y *subject* en Springer) y sólo en el caso de revistas: número de artículos a buscar y agrupar.

Controlador de Consulta en Elsevier

Los métodos creados para realizar las consultas son presentados a continuación.

Método para consultar revistas

```
def consultarRevistasElsevier (busqueda, jsonReceived, data,
cantidadArticulos):
    ...
```

Figura 3.20: Método para consultar revistas en Elsevier

Fuente: Elaboración propia

Método para consultar artículos

```
def consultarArticulosElsevier (busqueda, jsonReceived, data,
cantidadArticulos):
    ...
```

Figura 3.21: Método para consultar artículos en Elsevier

Fuente: Elaboración propia

Método para consultar artículos por issn

```
def consultarIssnElsevier (busqueda, jsonReceived, data,
cantidadArticulos):
    ...
```

Figura 3.22: Método para consultar artículos por issn en Elsevier

Fuente: Elaboración propia

Consulta artículos y revistas en Springer

Los métodos creados para realizar las consultas son presentados a continuación.

Método para consultar revistas

```
def consultarRevistasSpringer (busqueda, jsonReceived, data,
cantidadArticulos):
    ...
```

Figura 3.23: Método para consultar revistas en Springer

Fuente: Elaboración propia

Método para consultar artículos

```
def consultarArticulosSpringer (busqueda, jsonReceived, data,
cantidadArticulos):
    ...
```

Figura 3.24: Método para consultar artículos en Springer

Fuente: Elaboración propia

Método para consultar artículos por issn

```
def consultarIssnSpringer (busqueda, jsonReceived, data,
cantidadArticulos):
    ...
```

Figura 3.25: Método para consultar artículos por issn en Springer

Fuente: Elaboración propia

Se optó por la creación del método de consulta de artículos por issn (Figuras 3.22, 3.25) para que la respuesta de cada biblioteca virtual no contenga todos los artículos referidos a ese término, sino que en su documentación indican la forma en la cual colocar un filtro específico.

Web scraping a Scimago Journal & Country Rank

Modelos y Formatos JSON

Con los datos encontrados en la página a realizar el web scraping se pudo establecer los siguientes modelos y formatos JSON:

Formato JSON de Cuartil

```
def quartileFormat (data, valor1, valor2, valor3):
    data['quartile'].append({
        'category' : valor1,
        'year' : valor2,
        'quartile' : valor3
    })
```

Figura 3.26: Formato JSON de Cuartil

Fuente: Elaboración propia

Formato JSON de Citaciones por documento

```
def citationsPerDocumentFormat(data, valor1, valor2, valor3):
    data['citationsPerDocument'].append({
        'citespDocument' : valor1,
        'year'           : valor2,
        'value'          : valor3
    })
```

Figura 3.27: Formato JSON de Citaciones por documento

Fuente: Elaboración propia

Formato JSON de Citas totales

```
def totalCitesFormat(data, valor1, valor2, valor3):
    data['totalCites'].append({
        'cites' : valor1,
        'year'  : valor2,
        'value' : valor3
    })
```

Figura 3.28: Formato JSON de Citas totales

Fuente: Elaboración propia

Formato JSON de Citas externas por documento

```
def externalCitesPerDocFormat(data, valor1, valor2, valor3):
    data['externalCitesPerDoc'].append({
        'cites' : valor1,
        'year'  : valor2,
        'value' : valor3
    })
```

Figura 3.29: Formato JSON de Citas externas por documento

Fuente: Elaboración propia

Formato JSON de Documentos citables

```
def citableDocumentsFormat(data, valor1, valor2, valor3):
    data['citableDocuments'].append({
        'documents' : valor1,
        'year'       : valor2,
        'value'      : valor3
    })
```

Figura 3.30: Formato JSON de Documentos citables

Fuente: Elaboración propia

Formato JSON de Documentos citados

```
def citedDocumentsFormat(data, valor1, valor2, valor3):
    data['citedDocuments'].append({
        'documents' : valor1,
        'year'       : valor2,
        'value'      : valor3
    })
```

Figura 3.31: Formato JSON de Documentos citados

Fuente: Elaboración propia

Formato JSON de SJR

```
def sjrFormat(data, valor1, valor2):
    data['sjr'].append({
        'year'      : valor1,
        'sjr'       : valor2
    })
```

Figura 3.32: Formato JSON de SJR

Fuente: Elaboración propia

Modelo dos columnas

```
def createJsonTwoColumns(filas, data):
    valor1= ''
    valor2= ''
    for f, fila in enumerate(filas):
        columnas = fila.find_all('td')
        for c, columna in enumerate(columnas):
            if f== len(filas)-1 :
                if c == 0:
                    valor1 = columna.text
                elif c == 1:
                    valor2 = columna.text
            sjrFormat(data, valor1, valor2)
```

Figura 3.33: Modelo dos columnas

Fuente: Elaboración propia

Modelo tres columnas

```
def createJsonTreeColumns(filas, data, listaCrear):
    ...
```

Figura 3.34: Modelo tres columnas

Fuente: Elaboración propia

Controlador

Identificada la ubicación de los elementos a recuperar, se implementó el web scraping a partir del issn o eissn de la revista (el que se recibe por parámetro).

```
def scraper(issn):
    ...
```

Figura 3.35: Web scraping Scimago Journal & Country Rank

Fuente: Elaboración propia

Vistas

Se establecieron cinco vistas, de las que se detalla su función a continuación:

- Consultar revistas teniendo como parámetros: término de búsqueda, número de artículos a buscar y agrupar, área de búsqueda y posición de inicio de resultados de búsqueda tanto para Elsevier como Springer.
- Consultar artículos por término de interés teniendo como parámetros: término de búsqueda, cantidad de artículos, área de búsqueda y posición de inicio de resultados de búsqueda tanto para Elsevier como Springer.
- Consultar artículos por issn o eissn en Elsevier teniendo como parámetros: término de búsqueda, cantidad de artículos, área de búsqueda y posición de inicio de resultados de búsqueda.
- Consultar artículos por issn o eissn en Springer teniendo como parámetros: término de búsqueda, cantidad de artículos, área de búsqueda y posición de inicio de resultados de búsqueda.
- Consultar cuartiles, sjr y sus citaciones de la revista teniendo como parámetro issn (en el caso de no existir resultados se deberá enviar como parámetro secundario el eissn desde la aplicación móvil).

Se establecieron dos vistas de consultas de artículos por issn o eissn para evitar enviar consultas innecesarias a una biblioteca virtual y tener resultados duplicados en la respuesta (la biblioteca a la cual se realizará la consulta estará establecida en el modelo en cada revista).

Una vez definida su estructura, se indica su implementación a continuación:

Búsqueda de revistas

```
@app.route('/apisearch', methods = ['POST'])
def busquedaRevistas():
    jsonReceived = json.loads(request.form['json'])
    busqueda=''
    # Cambio espacio por %20
    for letra in jsonReceived['search']:
        if letra == ' ':
            busqueda += "%20"
        else:
            busqueda += letra
    # Inicializo el vector
    data = {}
    data = {}
    data['revistas'] = []
    data['revistas-encontradas'] = 0
    data['end-elsevier'] = jsonReceived['elsevier']['start']-1
    data['end-springer'] = jsonReceived['springer']['start']-1
    data['total-revistas-elsevier'] = 0
    data['total-revistas-springer'] = 0
    try:
        data = consultarRevistasElsevier(busqueda,
    jsonReceived['elsevier'], data, jsonReceived['cantidadArticulos'])
        data = consultarRevistasSpringer(busqueda,
    jsonReceived['springer'], data, jsonReceived['cantidadArticulos'])
    finally:
        pass
    return jsonify({"resultado": data})
```

Figura 3.36: Vista- Búsqueda de revistas

Fuente: Elaboración propia

Búsqueda de artículos

```
@app.route('/apiarticlesearch', methods = ['POST'])
def busquedaArticulos():
    jsonReceived = json.loads(request.form['json'])
    busqueda=''
    # Cambio espacio por %20
    for letra in jsonReceived['search']:
        if letra == ' ':
            busqueda += "%20"
        else:
            busqueda += letra
    # Inicializo el vector
    data = {}
    data = {}
    data['articulos'] = []
    data['articulos-encontrados'] = 0
    data['end-articulos-elsevier'] =
jsonReceived['elsevier']['start']-1
    data['end-articulos-springer'] =
jsonReceived['springer']['start']-1
    data['total-articulos-elsevier'] = 0
    data['total-articulos-springer'] = 0
    try:
        data = consultarArticulosElsevier(busqueda,
jsonReceived['elsevier'], data, jsonReceived['cantidadArticulos'])
        data = consultarArticulosSpringer(busqueda,
jsonReceived['springer'], data, jsonReceived['cantidadArticulos'])
    finally:
        pass
    return jsonify({"resultado": data})
```

Figura 3.37: Vista- Búsqueda de artículos

Fuente: Elaboración propia

Búsqueda de artículos por issn en Elsevier

```
@app.route('/apiissnelsevier', methods = ['POST'])
def busquedaIssnElsevier():
    jsonReceived = json.loads(request.form['json'])
    busqueda=''
    # Cambio espacio por %20
    for letra in jsonReceived['search']:
        if letra == ' ':
            busqueda += "%20"
        else:
            busqueda += letra
    # Inicializo el vector
    data = {}
    data = {}
    data['articulos'] = []
    data['articulos-encontrados'] = 0
    data['end-articulos-elsevier'] =
jsonReceived['elsevier']['start']-1
    data['total-articulos-elsevier'] = 0
    try:
        data = consultarIssnElsevier(busqueda,
jsonReceived['elsevier'], data, jsonReceived['cantidadArticulos'])
    finally:
        pass
    return jsonify({"resultado": data})
```

Figura 3.38: Vista- Búsqueda de artículos por issn en Elsevier

Fuente: Elaboración propia

Búsqueda de artículos por issn en Springer

```
@app.route('/apiissnspringer', methods = ['POST'])
def busquedaIssnSpringer():
    jsonReceived = json.loads(request.form['json'])
    busqueda=''
    # Cambio espacio por %20
    for letra in jsonReceived['search']:
        if letra == ' ':
            busqueda += "%20"
        else:
            busqueda += letra
    # Inicializo el vector
    data = {}
    data = {}
    data['articulos'] = []
    data['articulos-encontrados'] = 0
    data['end-articulos-springer'] =
jsonReceived['springer']['start']-1
    data['total-articulos-springer'] = 0
    try:
        data = consultarIssnSpringer(busqueda,
jsonReceived['springer'], data, jsonReceived['cantidadArticulos'])
    finally:
        pass
    return jsonify({"resultado": data})
```

Figura 3.39: Vista- Búsqueda de artículos por issn en Springer

Fuente: Elaboración propia

Web scraping

```
@app.route('/bosistant', methods = ['POST'])  
def bosistant():  
    jsonReceived = json.loads(request.form['json'])  
    return jsonify(scraper(jsonReceived['issn']))
```

Figura 3.40: Vista- Web scraping
Fuente: Elaboración propia

3.1.3.5 Planificación de la arquitectura

A continuación, se detalla la arquitectura que se tomó para el consumo de las APIs de las bibliotecas virtuales, tomar datos del web scraping, crear un servicio con los resultados y finalmente enviar y utilizar los datos en el aplicativo móvil.

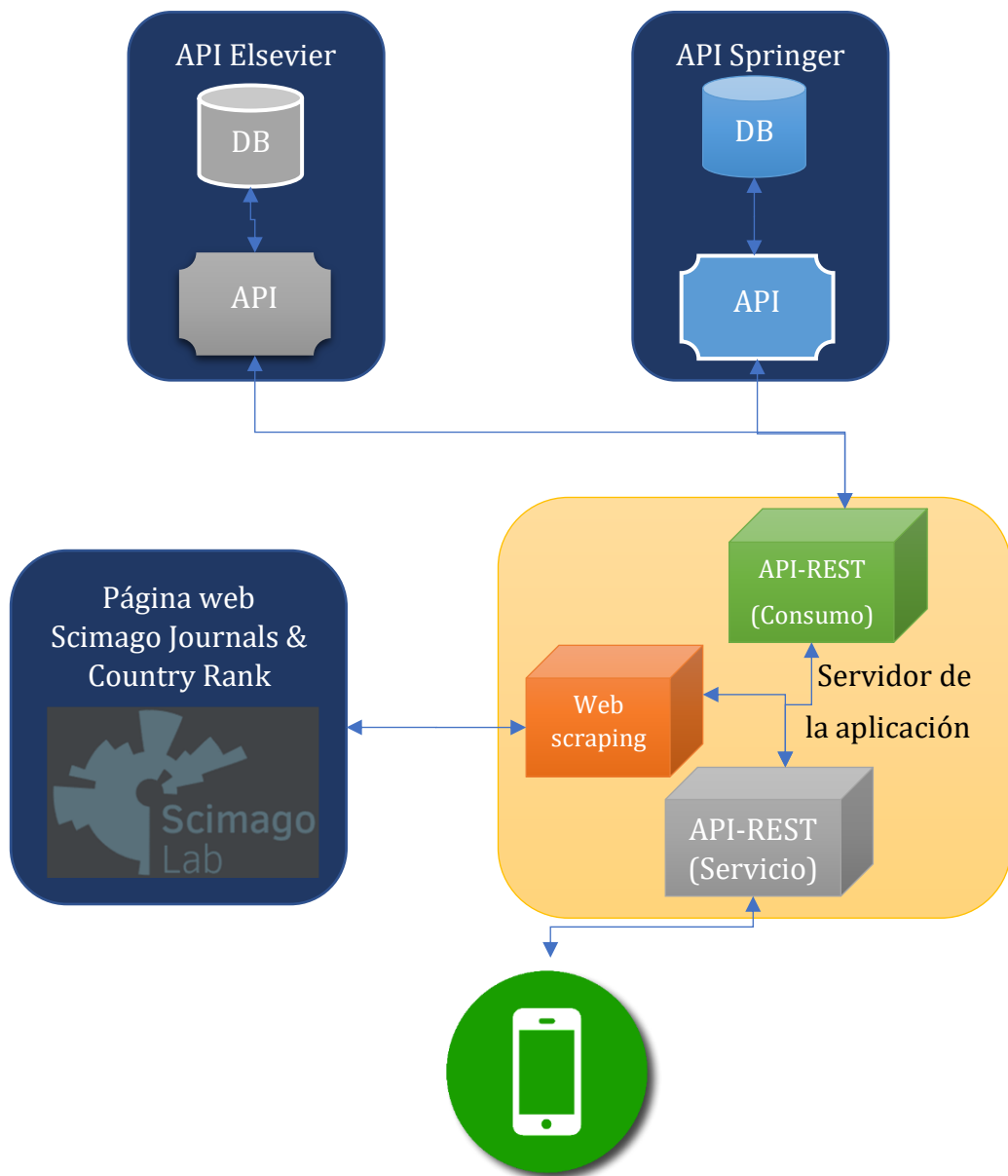


Figura 3.41: Arquitectura del proyecto
Fuente: Elaboración propia

3.1.3.6 Diagramas de actividades del proyecto

En el ingreso a la pestaña de artículos se cargará los parámetros por defecto y se realizará la consulta al servicio por API-REST. Una vez recibida la respuesta se mostrará en widgets dependiendo si tiene o no resultados.

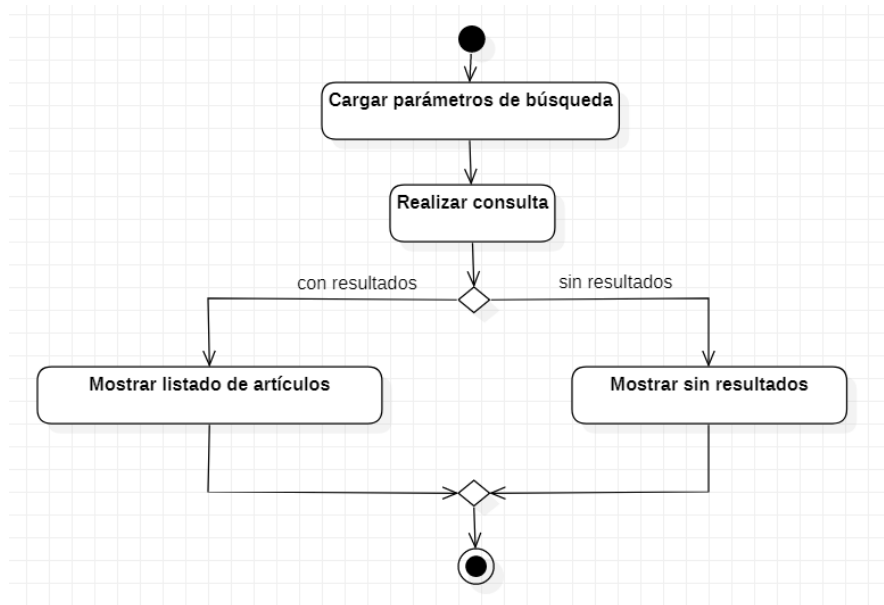


Figura 3.42: Diagrama de actividad.- Buscar artículos en inicio de pestaña
Fuente: Elaboración propia

Cuando el usuario cambie de término y presione el botón de buscar se realizará la consulta de artículos al servicio por API-REST. Una vez recibida la respuesta se mostrará en widgets dependiendo si tiene o no resultados.

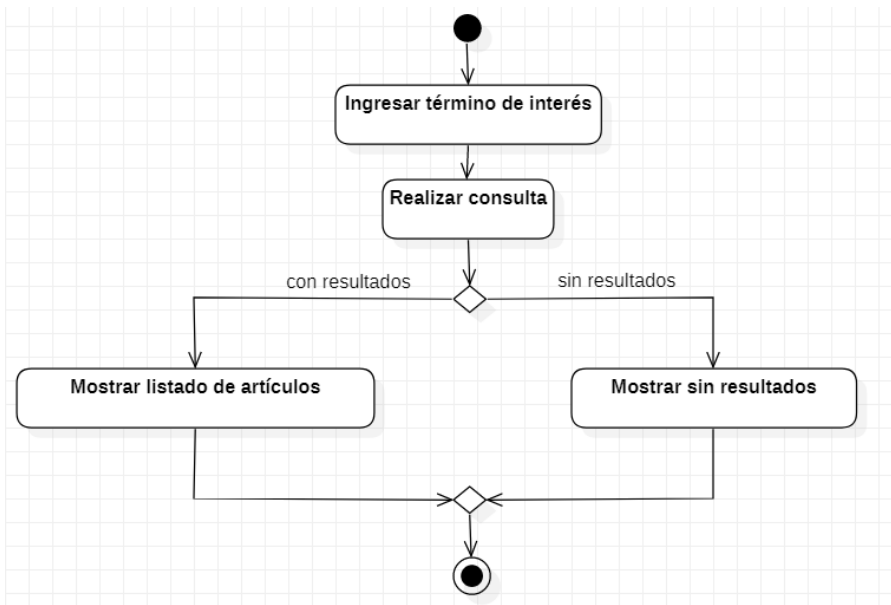


Figura 3.43: Diagrama de actividad.- Buscar artículos al final de la lista
Fuente: Elaboración propia

Una vez el usuario llegue al final de la lista realizando scroll se obtendrá la posición final de la última búsqueda realizada, realizará la consulta al servicio por API-REST con los nuevos parámetros y añadirá el resultado al listado final.



Figura 3.44: Diagrama de actividad.- Buscar artículos al final de la lista
Fuente: Elaboración propia

Si el usuario desea ver el artículo con más detalles deberá presionar en el ítem, abriendo el enlace en donde se encuentra en el navegador.

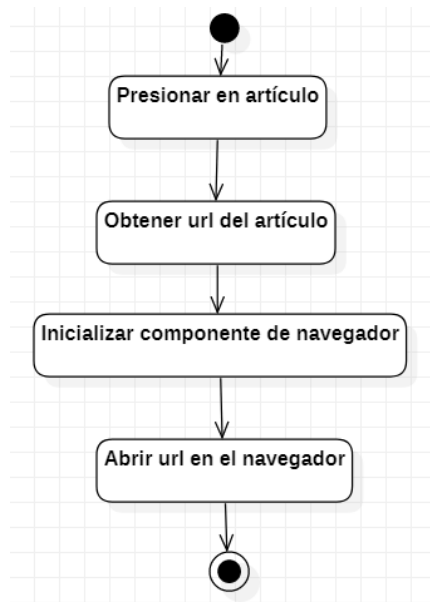


Figura 3.45: Diagrama de actividad.- Redirigir sitio web del artículo
Fuente: Elaboración propia

En el ingreso a la pestaña de revistas se cargará los parámetros por defecto y se realizará la consulta al servicio por API-REST. Una vez recibida la respuesta se mostrará en widgets dependiendo si tiene o no resultados.

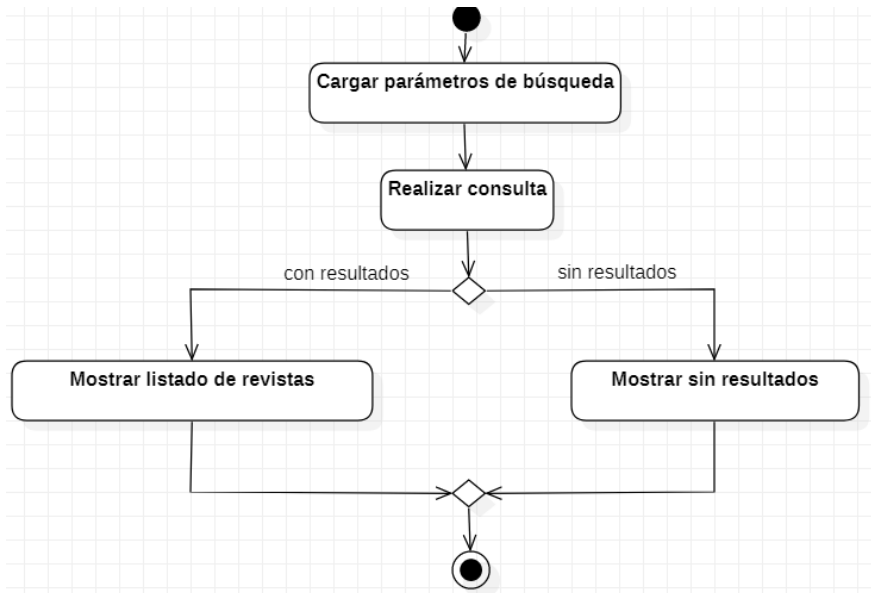


Figura 3.46: Diagrama de actividad.- Buscar revistas en el inicio de pestaña
Fuente: Elaboración propia

Cuando el usuario cambie de término y presione el botón de buscar se realizará la consulta de revistas al servicio por API-REST. Una vez recibida la respuesta se mostrará en widgets dependiendo si tiene o no resultados.

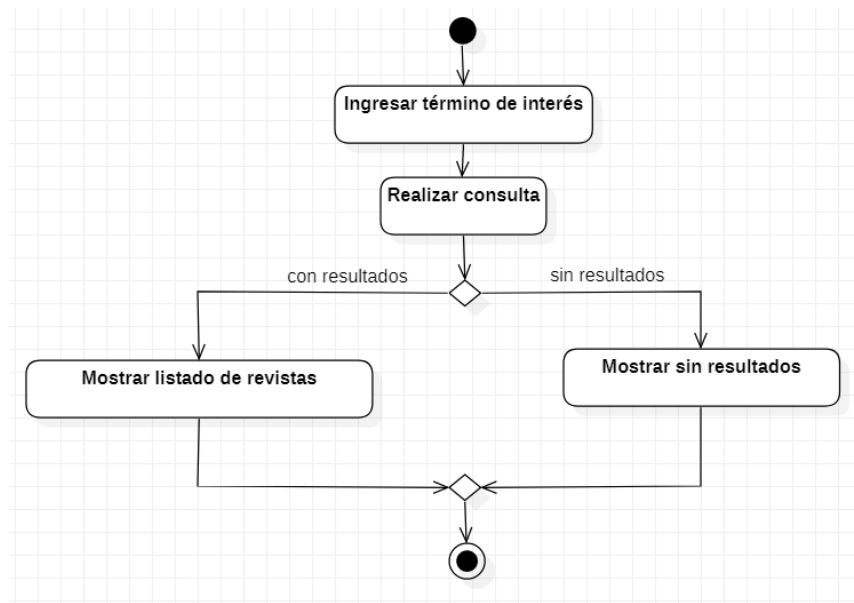


Figura 3.47: Diagrama de actividad.- Buscar revistas con cambio de término por usuario
Fuente: Elaboración propia

Cuando el usuario presione en una revista se abrirá la pestaña de revista enviando los parámetros de búsqueda, se seleccionará el parámetro a buscar y se realizará la consulta al servicio por API-REST. Dependiendo cuál sea el parámetro se realizará otra búsqueda con el parámetro secundario si no se encuentra resultados (en el caso de issn) y se mostrará los resultados.

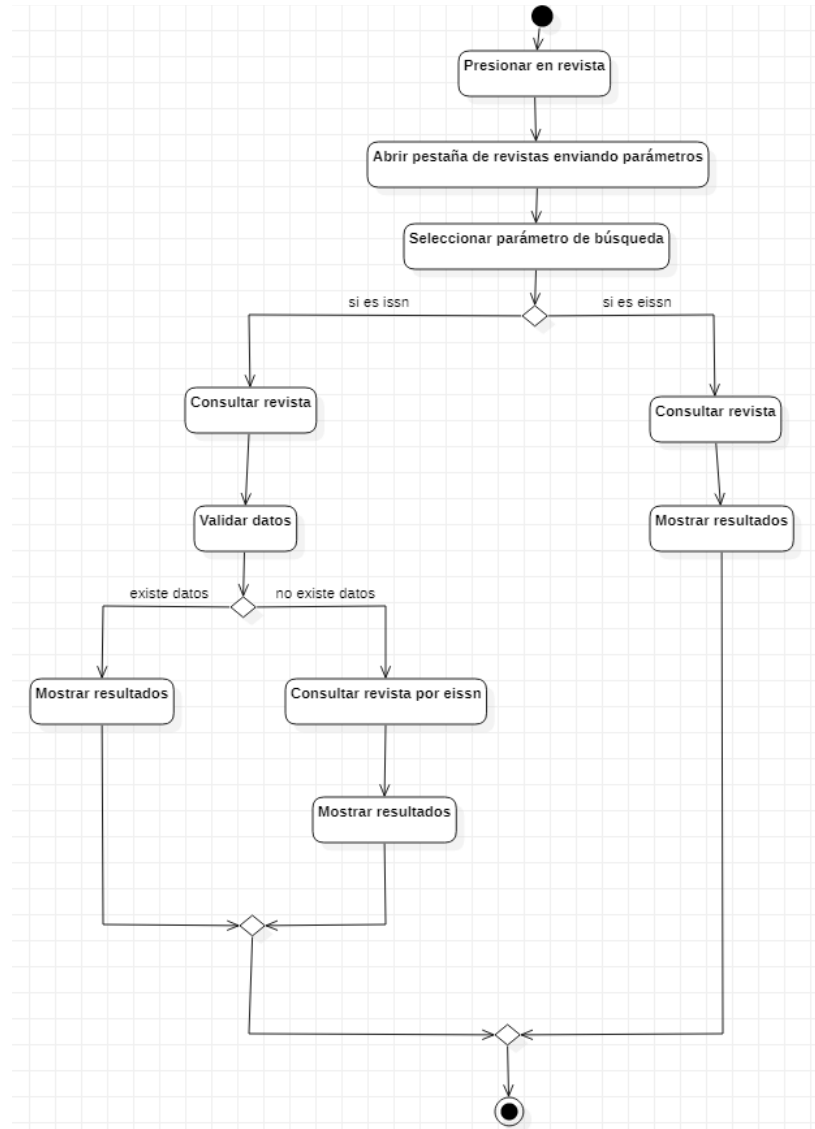


Figura 3.48: Diagrama de actividad.- Consultar datos de revista
Fuente: Elaboración propia

Al presionar el botón Artículos se abrirá una nueva pestaña enviando los parámetros de búsqueda. Se tomará el parámetro principal, la biblioteca a la cual se realizará la consulta y se enviará a consulta al servicio por API-REST. Una vez obtenida la respuesta se mostrará los resultados al usuario.

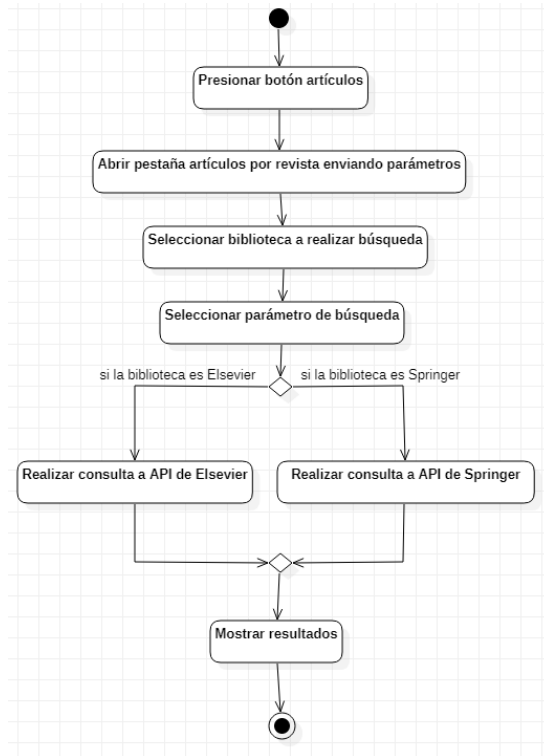


Figura 3.49: Diagrama de actividad.- Buscar artículos por revista
Fuente: Elaboración propia

Al presionar el botón editar se desbloqueará el campo de texto mostrando dos opciones: cancelar el proceso o editar el término por defecto y aceptar los cambios, actualizando la base de datos de parámetros y mostrando un mensaje al usuario para finalmente bloquear nuevamente el campo de texto.

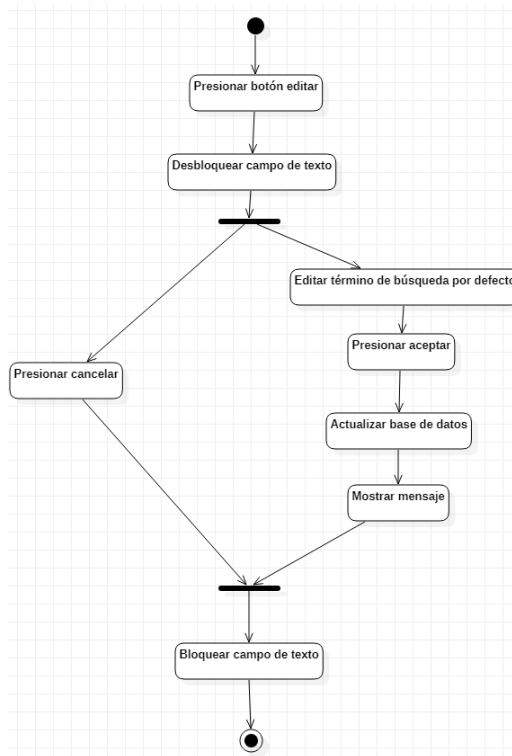


Figura 3.50: Diagrama de actividad.- Cambiar término de búsqueda
Fuente: Elaboración propia

Cuando el usuario presione en el dropdown se desplegará un listado de números que se utilizarán para buscar y agrupar artículos. Permitiéndole seleccionar una opción, que posteriormente actualizará la base de datos de parámetros y se mostrará un mensaje de actualización de datos.

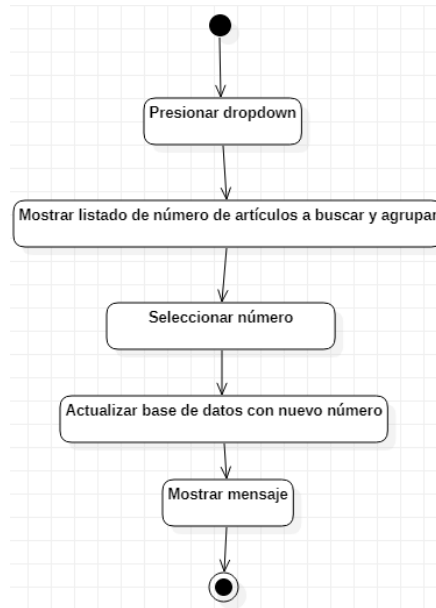


Figura 3.51: Diagrama de actividad.- Cambiar número de artículos a buscar y agrupar
Fuente: Elaboración propia

Cuando el usuario presione en el dropdown se desplegará un listado de áreas de búsqueda que serán utilizados para filtrar las búsquedas. Permitiéndole seleccionar una opción, que posteriormente actualizará la base de datos de parámetros y se mostrará un mensaje de actualización del área de búsqueda.

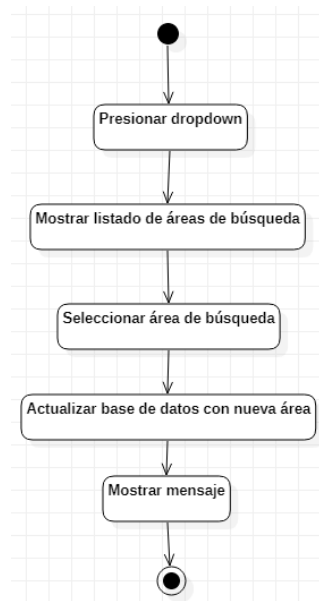


Figura 3.52: Diagrama de actividad.- Cambiar área de búsqueda de biblioteca virtual
Fuente: Elaboración propia

3.1.3.7 Diagramas de secuencias del proyecto

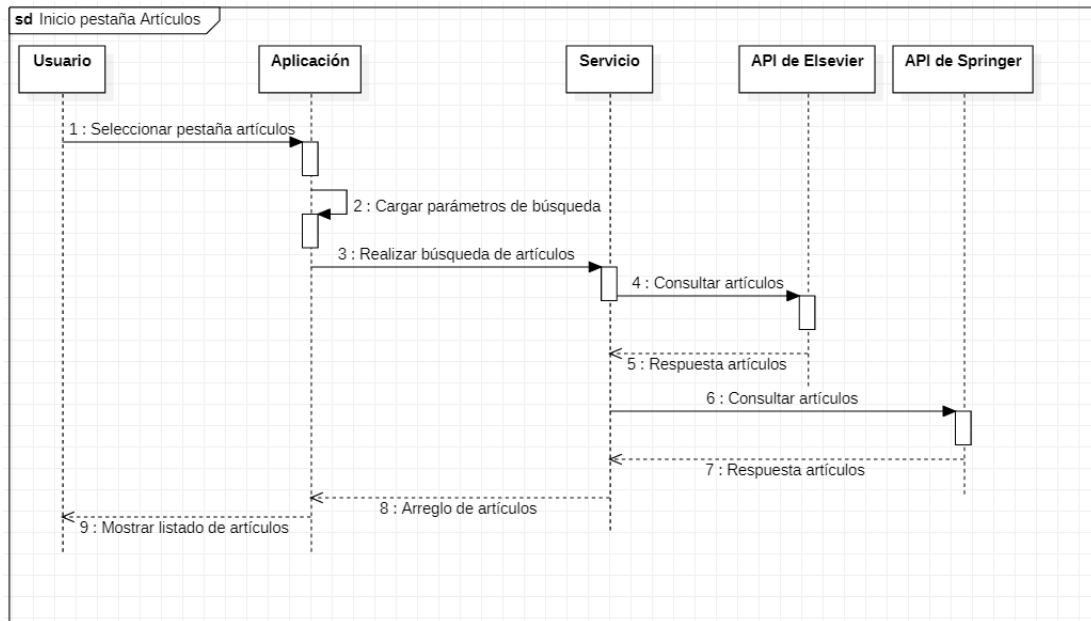


Figura 3.53: Diagrama de secuencia.- Inicio pestaña Artículos
Fuente: Elaboración propia

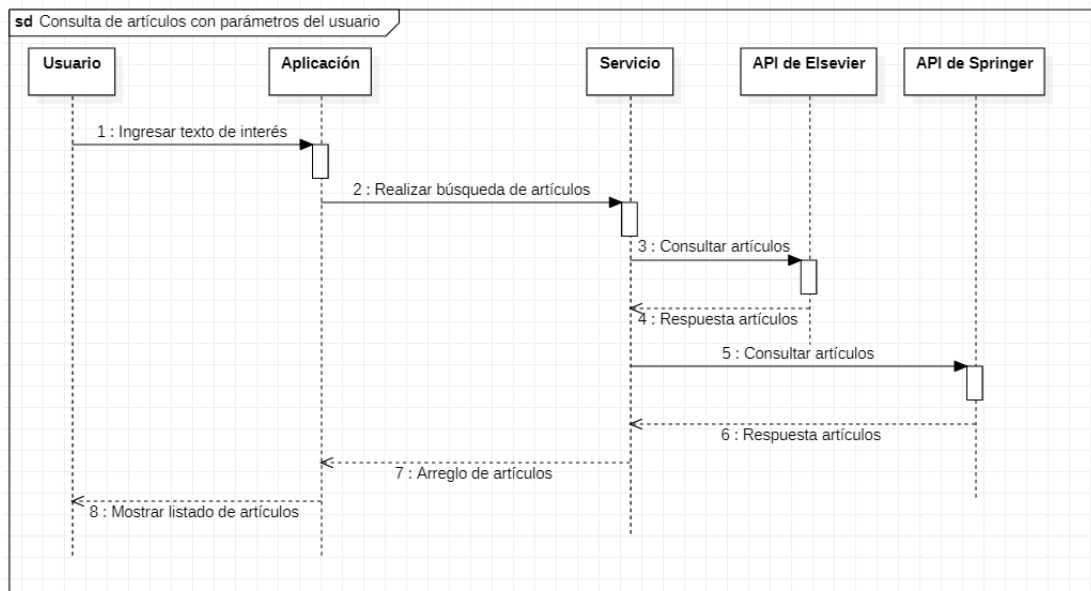


Figura 3.54: Diagrama de secuencia.- Consulta de artículos con parámetros del usuario
Fuente: Elaboración propia

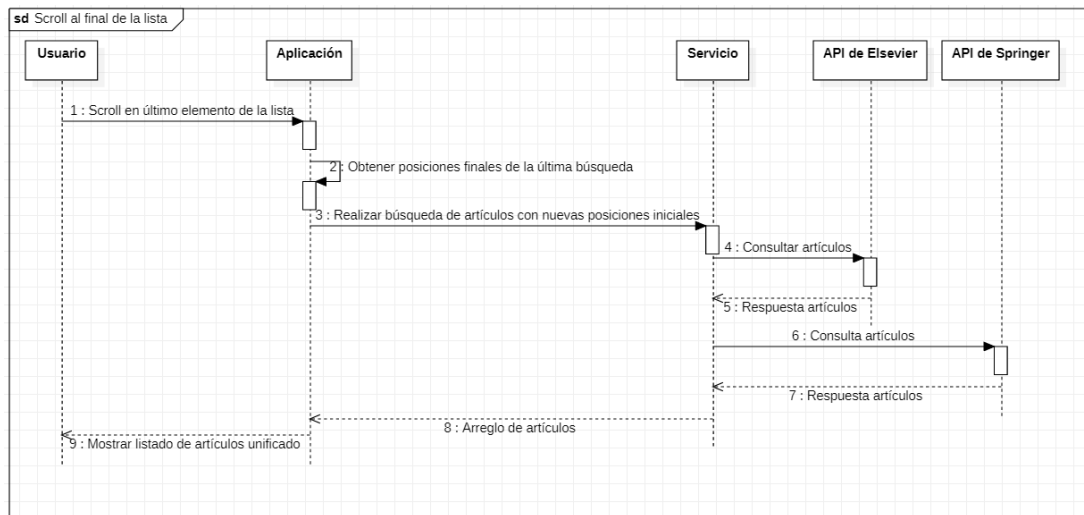


Figura 3.55: Diagrama de secuencia.- Scroll al final de la lista
Fuente: Elaboración propia

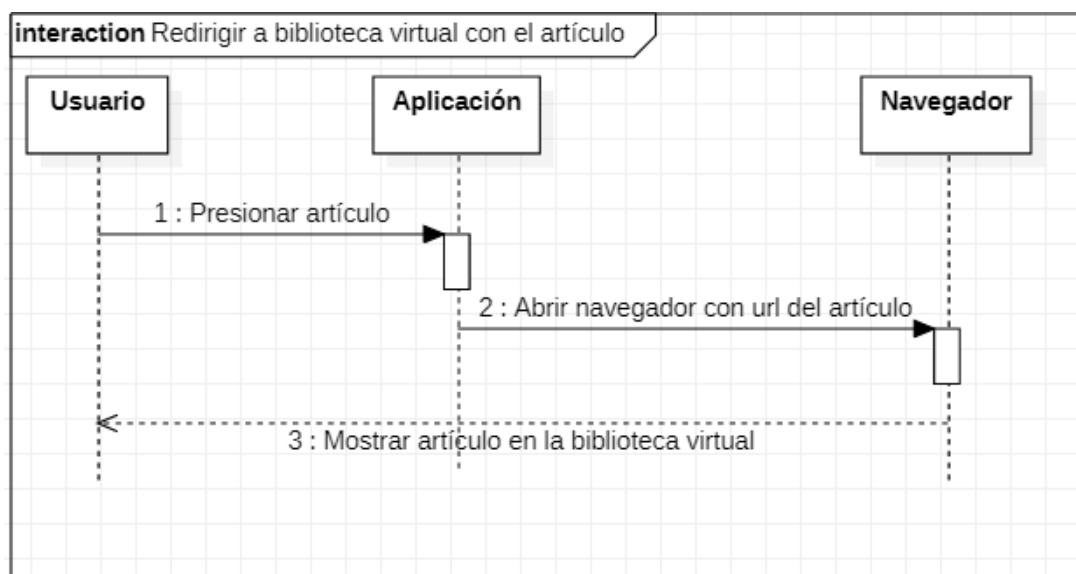


Figura 3.56: Diagrama de secuencia.- Redirigir a biblioteca virtual con el artículo
Fuente: Elaboración propia

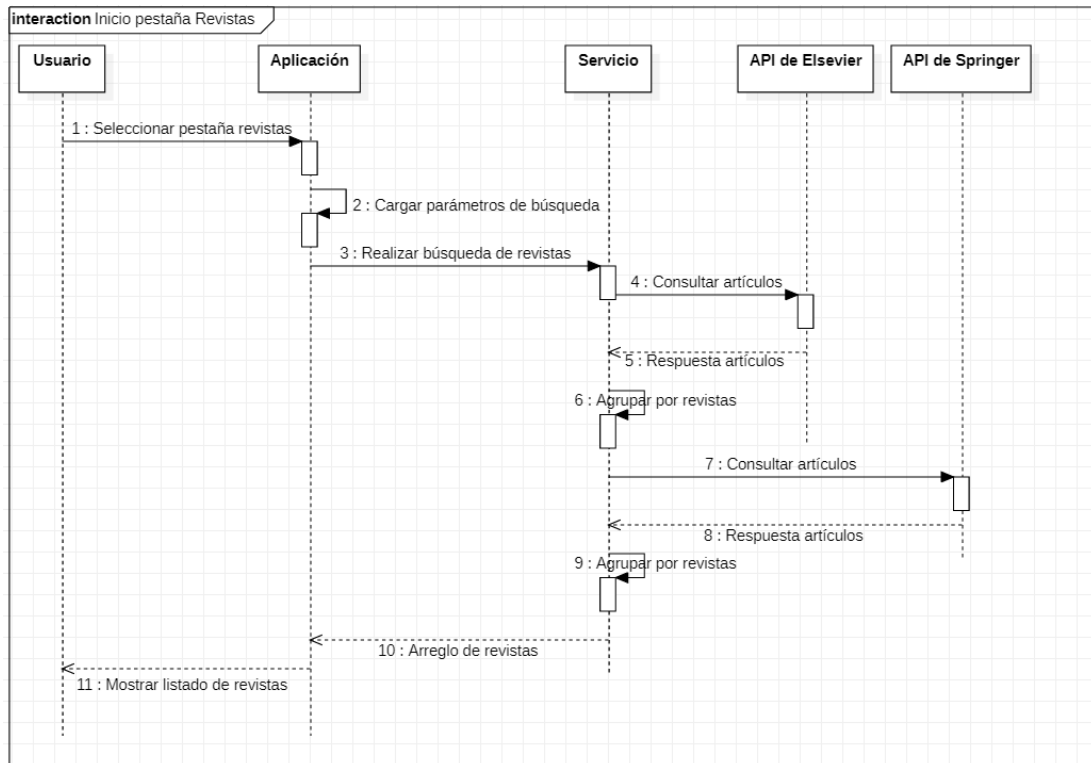


Figura 3.57: Diagrama de secuencia.- Inicio pestaña Revistas
Fuente: Elaboración propia

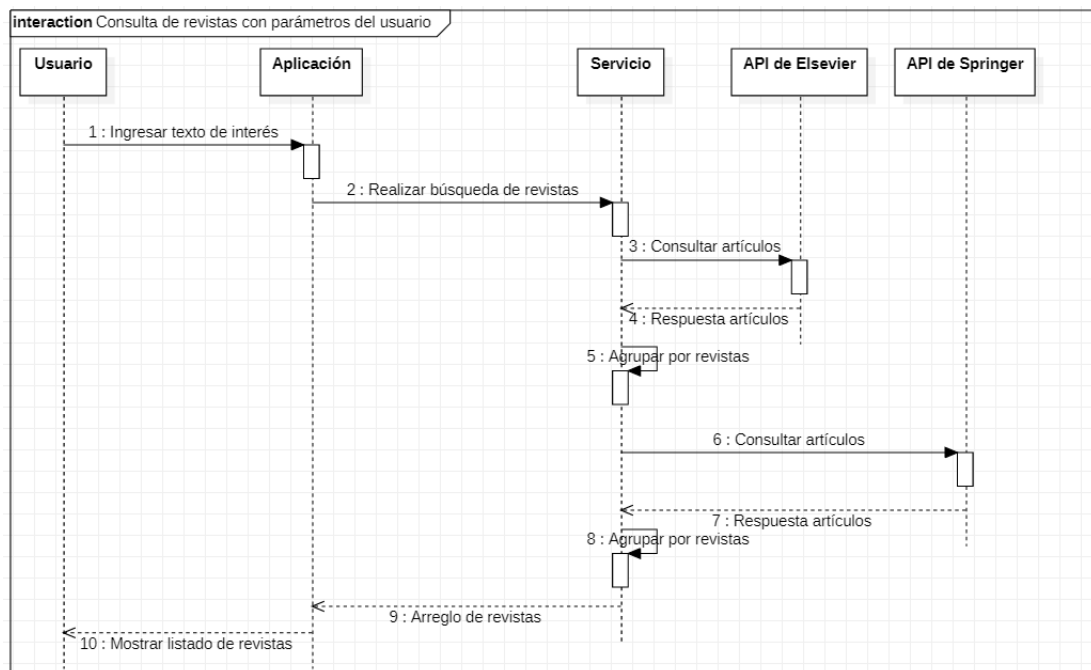


Figura 3.58: Diagrama de secuencia.- Consulta de revistas con parámetros del usuario
Fuente: Elaboración propia

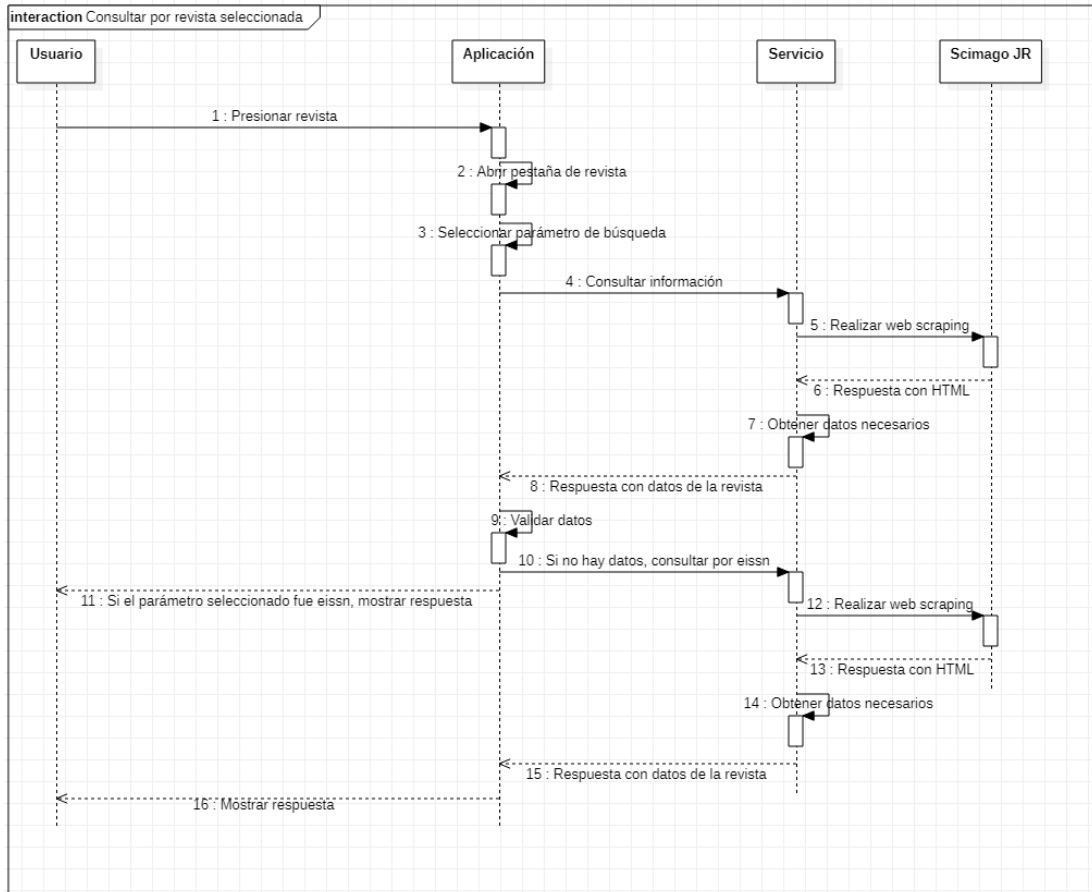


Figura 3.59: Diagrama de secuencia.- Consultar por revista seleccionada
Fuente: Elaboración propia

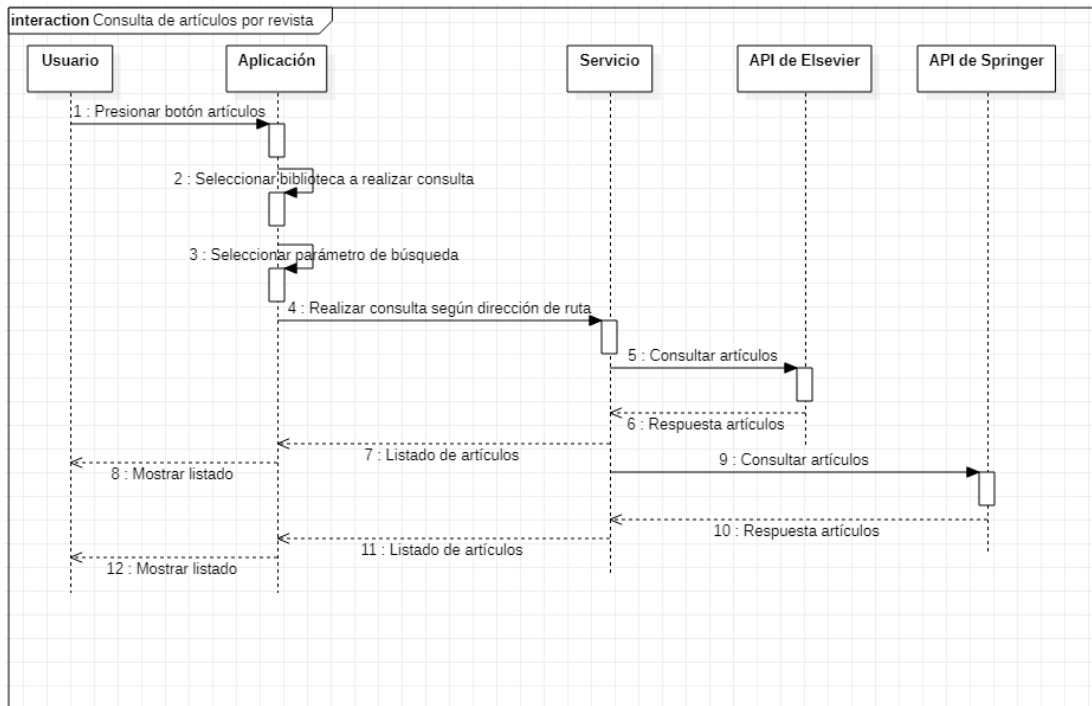


Figura 3.60: Diagrama de secuencia.- Consulta de artículos por revista
Fuente: Elaboración propia

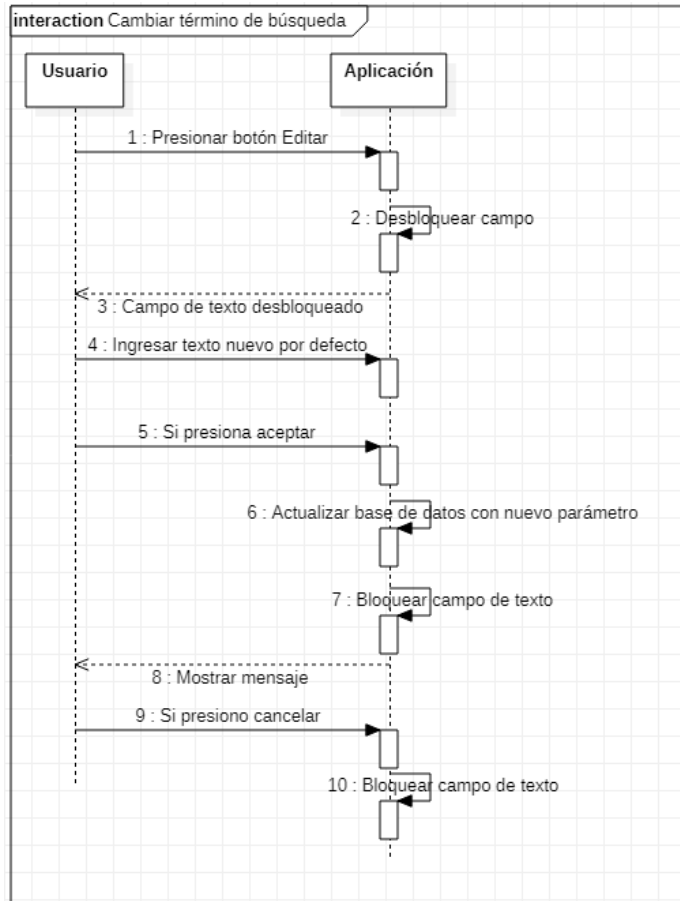


Figura 3.61: Diagrama de secuencia.- Cambiar término de búsqueda
Fuente: Elaboración propia

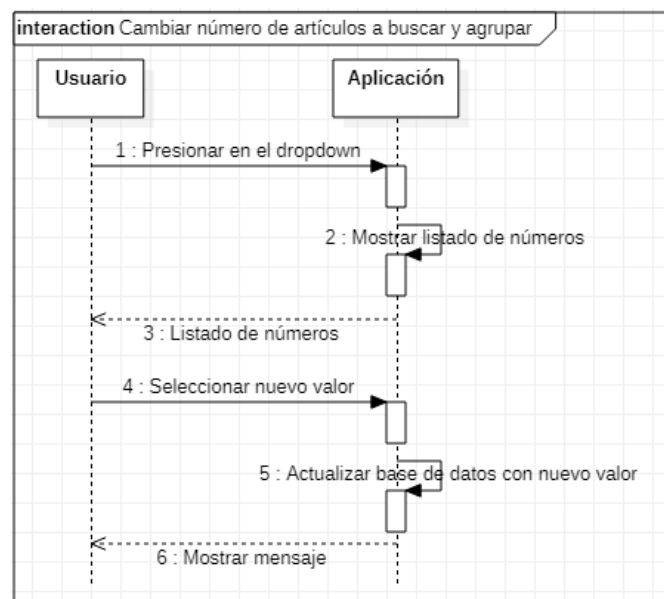


Figura 3.62: Diagrama de secuencia.- Cambiar número de artículos a buscar y agrupar
Fuente: Elaboración propia

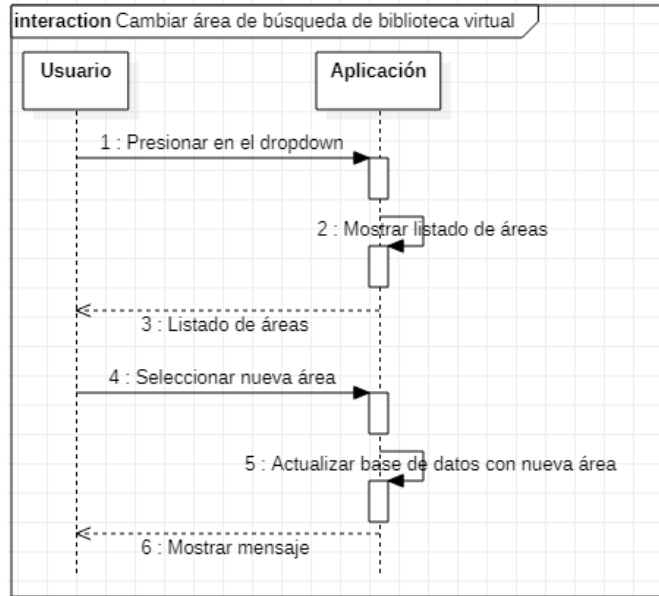


Figura 3.63: Diagrama de secuencia.- Cambiar área de búsqueda de biblioteca virtual
Fuente: Elaboración propia

3.1.3.8 Configuración de proyecto móvil con paquetes necesarios

Instalación de paquetes necesarios para el proyecto.

pubspec.yaml

```

dependencies:
  flutter:
    sdk: flutter
  sqflite: ^1.3.1+1
  path_provider: ^1.6.14
  http: ^0.12.2
  skeleton_text: ^1.0.0
  url_launcher: ^5.5.3
  
```

Figura 3.64: Instalación de paquetes necesarios para el proyecto
Fuente: Elaboración propia

3.1.4 Fase de producción

3.1.4.1 Iteración 1.- Navigation Bar

Día de planificación

- Redirigir a pestañas

Descripción	
Como	Aplicación
Necesito	Redirigir a pestaña seleccionada.
Para	El usuario pueda realizar búsquedas o actualizar parámetros.

Tabla 3.3: Iteración 1- Redirigir a pestañas

Fuente: Elaboración propia

- Colocar título según opción seleccionada

Descripción	
Como	Aplicación
Necesito	Colocar título a la pestaña según su selección.
Para	El usuario conozca la ubicación en la cual se encuentra.
Criterio de Aceptación	
Dado	Deseo movilizarme dentro de la aplicación.
Cuando	Presione una opción en la barra de navegación
Entonces	Se redirigirá a la pestaña y se colocará el título.

Tabla 3.4: Iteración 1- Colocar título según opción seleccionada

Fuente: Elaboración propia

- Establecer opción de búsqueda para ser usada por las otras pestañas

Descripción	
Como	Aplicación
Necesito	Mostrar una opción de búsqueda
Para	Ser usada por el usuario y realizar consultas en artículos y revistas.
Criterio de Aceptación	
Dado	Deseo mostrar al usuario un campo de texto para que pueda realizar búsquedas.
Cuando	Presione el botón de búsqueda
Entonces	Mostrará visible/invisible el campo de texto en la pestaña de artículos y revistas (en el caso de parámetros no se mostrará, pero volverá al estado anterior cuando se acceda nuevamente a las otras pestañas) y mostrará el texto almacenado o personalizado (según sea el caso) del usuario que será colocado en las <i>Iteraciones 2 y 3</i> .

Tabla 3.5: Iteración 1- Establecer opción de búsqueda para ser usada por las otras pestañas

Fuente: Elaboración propia

Día de trabajo

El fragmento de código presentado a continuación hace énfasis en lo más importante realizado en la *Iteración 1*.

home_page

```

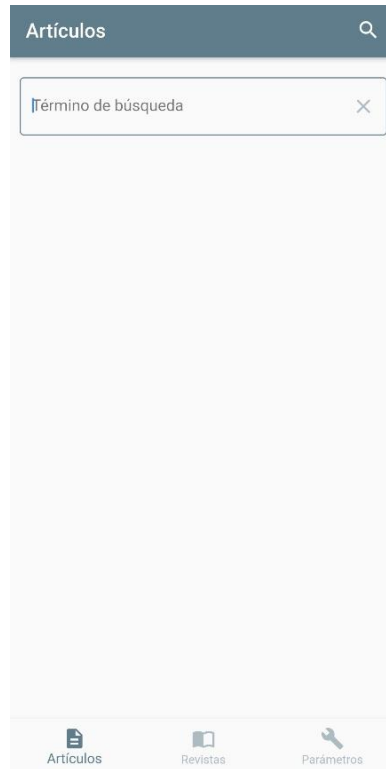
@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomPadding: false,
    appBar: _appBarVisible
      ? AppBar(
          title: Text('$_titulo'),
          actions: <Widget>[
            IconButton(
              padding: EdgeInsets.only(right: 10.0),
              icon: Icon(Icons.search),
              onPressed: () {
                _textFieldVisible = !_textFieldVisible;
                setState(() {});
              }
            )
          ],
        )
      : PreferredSize(
          preferredSize: Size(0.0, 0.0),
          child: Container(),
        ),
    body: StreamBuilder<Parametro>(
      stream: _parametrosBloc.parametrosBusquedaStream,
      builder: (BuildContext context, AsyncSnapshot<Parametro> snapshot) {
        if (snapshot.hasData) {
          return Column(
            children: <Widget>[
              _textFieldBusqueda(snapshot.data),
              _paginaSeleccionada(snapshot.data),
            ],
          );
        }
        return CircularProgressIndicator();
      },
    ),
    bottomNavigationBar: StreamBuilder<Parametro>(
      stream: _parametrosBloc.parametrosBusquedaStream,
      builder: (BuildContext context, AsyncSnapshot<Parametro> snapshot) {
        if (snapshot.hasData) {
          return _navegacionPaginas(context, snapshot.data);
        }
        return CircularProgressIndicator();
      },
    ),
  );
}

```

*Figura 3.65: Iteración 1- Trabajo(home_page)
Fuente: Elaboración propia*

Día de liberación

Demostración del cumplimiento de la iteración según las historias de usuario establecidas.



*Figura 3.66: Iteración 1- Liberación (Home Page)
Fuente: Elaboración propia*

3.1.4.2 Iteración 2.- Búsqueda de artículos

Día de planificación

- **Mostrar artículos según término de interés**

Descripción	
Como	Usuario
Necesito	Obtener un listado de artículos mostrando: título, biblioteca, doi (si posee), tipo de publicación y de contenido; según término de interés.
Para	Visualizar artículos científicos pertenecientes a varias bibliotecas virtuales.

*Tabla 3.6: Iteración 2- Mostrar artículos según término de interés
Fuente: Elaboración propia*

- **Cargar texto por defecto almacenado**

Descripción	
Como	Aplicación
Necesito	Colocar texto por defecto almacenado en la base de datos del dispositivo.
Para	Tener un término de interés preparado para realizar la consulta cuando se ingrese a la pestaña.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo.
Cuando	Inicialice la pestaña de Artículos.
Entonces	Realizará la consulta de texto por defecto a la base de datos del dispositivo y colocará en el campo de texto.

*Tabla 3.7: Iteración 2- Cargar texto por defecto almacenado
Fuente: Elaboración propia*

- **Cargar áreas de búsqueda almacenadas**

Descripción	
Como	Aplicación
Necesito	Cargar áreas de búsqueda almacenados en la base de datos del dispositivo y añadir a la cabecera HTTP (junto con el término de interés) para enviar al servicio.
Para	Tener preparado parámetros que servirán en otros procesos.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo y realizar una consulta al servicio al iniciar la pestaña.
Cuando	Inicialice la pestaña de Artículos.
Entonces	Realizará la consulta de áreas de búsqueda de Elsevier y Springer a la base de datos del dispositivo y enviará una consulta al servicio por API-REST.

*Tabla 3.8: Iteración 2- Cargar áreas de búsqueda almacenadas
Fuente: Elaboración propia*

- **Establecer término de interés mediante campo de texto**

Descripción	
Como	Usuario
Necesito	Establecer un término para realizar búsqueda
Para	Cambiar la cabecera que se enviará por HTTP para realizar la consulta.
Criterio de Aceptación	
Dado	Deseo ingresar un término para realizar la búsqueda.
Cuando	Presione el botón de buscar que se encuentra en el Appbar.
Entonces	Se mostrará visible el campo de texto para ingresar término o invisible para observar de mejor manera los resultados.

*Tabla 3.9: Iteración 2- Establecer término de interés mediante campo de texto
Fuente: Elaboración propia*

- **Realizar consulta por API-REST a servicio integrador de bases de datos**

Descripción	
Como	Usuario
Necesito	Realizar una consulta con el término de interés y áreas de búsqueda (establecidas en la pestaña de parámetros).
Para	Enviar la consulta al servicio por API-REST
Criterio de Aceptación	
Dado	Deseo enviar una búsqueda de artículos científicos una vez establecido el término y filtros.
Cuando	Presione el botón de buscar (en el teclado o integrado en el campo de texto).
Entonces	Enviará una consulta al servicio por API-REST con los nuevos parámetros establecidos.

*Tabla 3.10: Iteración 2- Realizar consulta por API-REST a servicio integrador de bases de datos
Fuente: Elaboración propia*

- **Mostrar skeleton**

Descripción	
Como	Aplicación
Necesito	Mostrar skeleton mientras se espera la respuesta del servicio
Para	Indicar al usuario que la consulta se está realizando.
Criterio de Aceptación	
Dado	Debo esperar la respuesta del servicio.
Cuando	Envíe una consulta al servicio.
Entonces	Mostrará un skeleton con animación de carga.

*Tabla 3.11: Iteración 2- Mostrar skeleton
Fuente: Elaboración propia*

- **El servicio responde y no existe resultados**

Descripción	
Como	Aplicación

Necesito	Mostrar Widget cuando el servicio respondió, pero no se tiene resultados
Para	Indicar al usuario que la consulta se realizó.
Criterio de Aceptación	
Dado	No existe resultados.
Cuando	El servicio envíe una respuesta.
Entonces	Mostrar un widget con el mensaje “No se encontraron resultados”

Tabla 3.12: Iteración 2- El servicio responde y no existe resultados
Fuente: Elaboración propia

○ **El servicio responde y existe resultados**

Descripción	
Como	Aplicación
Necesito	Tener una lista preparada para ser mostrada al usuario.
Para	Tener datos para colocar en los widgets y este sea cómodo a la vista.
Criterio de Aceptación	
Dado	Existe resultados.
Cuando	El servicio envíe una respuesta.
Entonces	Debo buscar la imagen de la biblioteca científica perteneciente a cada artículo y añadir a la lista.

Tabla 3.13: Iteración 2- El servicio responde y existe resultados
Fuente: Elaboración propia

○ **Mostrar listado final al usuario**

Descripción	
Como	Aplicación
Necesito	Colocar listado final en Widgets con imagen de carga de biblioteca científica
Para	El usuario observe los resultados de la consulta realizada al servicio integrador de base de datos
Criterio de Aceptación	
Dado	La lista se encuentre con todos los parámetros necesarios
Cuando	La búsqueda de imágenes concluyó
Entonces	Mostrar el listado resultante con una imagen de carga de cada biblioteca mientras carga de internet. Mientras tanto el usuario puede seguir observando los resultados.

Tabla 3.14: Iteración 2- Mostrar listado final al usuario
Fuente: Elaboración propia

○ **Consultar artículo científico**

Descripción	
Como	Usuario
Necesito	Ingresar al sitio donde se encuentra el artículo científico.
Para	Observar más detalles.
Criterio de Aceptación	

Dado	Deseo ver con detalle el artículo.
Cuando	Presione en el ítem.
Entonces	Debo redirigir al navegador por el url del ítem seleccionado.

Tabla 3.15: Iteración 2- Consultar artículo científico
Fuente: Elaboración propia

- **Cargar los siguientes artículos**

Descripción	
Como	Usuario
Necesito	Cargar y agregar el siguiente listado de revistas al existente.
Para	Observar los siguientes resultados por el término enviado.
Criterio de Aceptación	
Dado	Deseo ver los siguientes resultados.
Cuando	El scroll llegue al final de la lista.
Entonces	Debo cargar los siguientes resultados por término.

Tabla 3.16: Iteración 2- Cargar los siguientes artículos
Fuente: Elaboración propia

Día de trabajo

Los fragmentos de código presentados a continuación hacen énfasis a lo más importante realizado en la *Iteración 2*.

home_page

```
void realizarBusqueda(Parametro parametro) {
    final Busqueda busqueda = new Busqueda(
        busqueda: parametro.texto,
        startElsevier: 1,
        startSpringer: 1,
        subElsevier: parametro.subElsevier,
        subSpringer: parametro.subSpringer,
        numArticulos: 10);
    _articulosBloc.limpiarArticulosConsulta();
    _articulosBloc.obtenerArticulosConsulta(busqueda);
}
```

Figura 3.67: Iteración 2- Trabajo (home_page)
Fuente: Elaboración propia

articulos_page

```
class ArticulosPage extends StatefulWidget {...}

class _ArticulosPageState extends State<ArticulosPage> {
  @override
  void initState() {...}

  @override
  void dispose() {...}

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<ArticuloConsulta>(
      stream: _articulosBloc.articulosConsultaStream,
      builder:
        (BuildContext context, AsyncSnapshot<ArticuloConsulta> snapshot) {
          if (snapshot.hasData && snapshot.data.articulos != null) {
            if (snapshot.data.articulos.length > 0) {
              return Expanded(
                child: GestureDetector(
                  onPanDown: (_) {
                    FocusScope.of(context).requestFocus(new FocusNode());
                  },
                  child: _articulos(snapshot.data)),
              );
            }
            return Padding(
              padding: const EdgeInsets.only(top: 100),
              child: SinResultadosPage(),
            );
          }
          return Expanded(child: obtenerSkeletonArticulos());
        },
    );
  }

  Widget _articulos(ArticuloConsulta articuloConsulta) {...}

  Widget _crearLoading() {...}

  Widget _articuloTile(Articulo item) {...}
}
```

Figura 3.68: Iteración 2- Trabajo (articulos_page)
Fuente: Elaboración propia

Día de liberación

Demostración del cumplimiento de la iteración según las historias de usuario establecidas.



*Figura 3.69: Iteración 2- Liberación (carga skeleton)
Fuente: Elaboración propia*

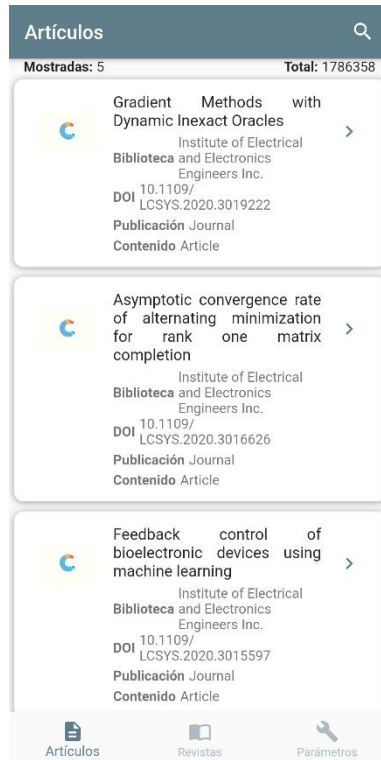


Figura 3.70: Iteración 2- Liberación (carga imágenes bibliotecas virtuales)
Fuente: Elaboración propia

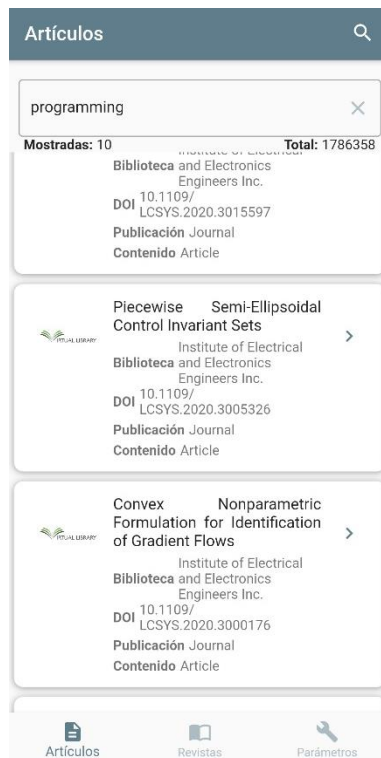


Figura 3.71: Iteración 2- Liberación (búsqueda realizada con campo de texto visible)
Fuente: Elaboración propia

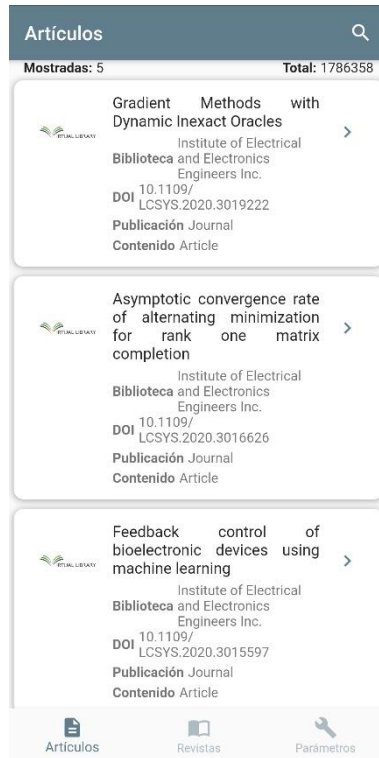


Figura 3.72: Iteración 2- Liberación (búsqueda realizada con campo de texto oculto)
 Fuente: Elaboración propia

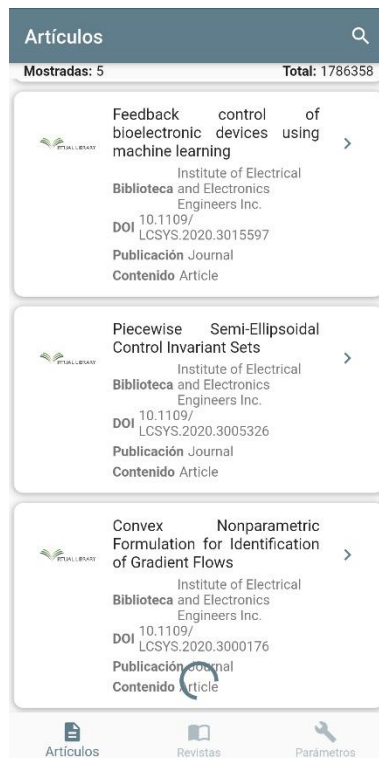


Figura 3.73: Iteración 2- Liberación (cargando siguientes artículos con scroll)
 Fuente: Elaboración propia



Figura 3.74: Iteración 2- Liberación (búsqueda sin resultados)
Fuente: Elaboración propia

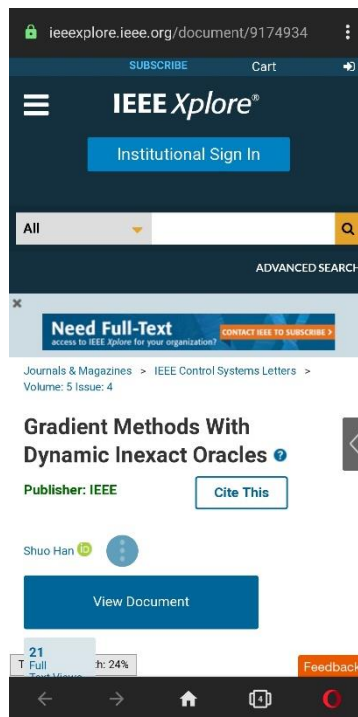


Figura 3.75: Iteración 2- Liberación (redirección al presionar artículo)
Fuente: Elaboración propia

3.1.4.3 Iteración 3.- Búsqueda de revistas

Día de planificación

- **Mostrar revistas según término de interés**

Descripción	
Como	Usuario
Necesito	Obtener un listado de revistas mostrando: título, biblioteca, issn (si posee), eissn (si posee), tipo de publicación y de contenido; según término de interés.
Para	Visualizar revistas científicas pertenecientes a varias bibliotecas virtuales.

*Tabla 3.17: Iteración 3- Mostrar revistas según término de interés
Fuente: Elaboración propia*

- **Cargar texto por defecto almacenado**

Descripción	
Como	Aplicación
Necesito	Colocar texto por defecto almacenado en la base de datos del dispositivo.
Para	Tener un término de interés preparado para realizar la consulta cuando se ingrese a la pestaña.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo.
Cuando	Inicialice la pestaña de Revistas.
Entonces	Realizará la consulta de texto por defecto a la base de datos del dispositivo y colocará en el campo de texto.

*Tabla 3.18: Iteración 3- Cargar texto por defecto almacenado
Fuente: Elaboración propia*

- **Cargar número de artículos almacenado**

Descripción	
Como	Aplicación
Necesito	Cargar número de artículos a buscar y agrupar almacenado en la base de datos del dispositivo.
Para	Tener preparado parámetros que servirán en otros procesos.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo.
Cuando	Inicialice la pestaña de Revistas.
Entonces	Realizará la consulta de número de artículos a la base de datos del dispositivo y guardará en una variable local.

*Tabla 3.19: Iteración 3- Cargar número de artículos almacenado
Fuente: Elaboración propia*

○ **Cargar áreas de búsqueda almacenadas**

Descripción	
Como	Aplicación
Necesito	Cargar áreas de búsqueda almacenados en la base de datos del dispositivo y añadir a la cabecera HTTP (junto con el término de interés y número de artículos) para enviar al servicio.
Para	Tener preparado parámetros que servirán en otros procesos.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo y realizar una consulta al servicio al iniciar la pestaña.
Cuando	Inicialice la pestaña de Artículos.
Entonces	Realizará la consulta de áreas de búsqueda de Elsevier y Springer a la base de datos del dispositivo y enviará una consulta al servicio por API-REST.

*Tabla 3.20: Iteración 3- Cargar áreas de búsqueda almacenadas
Fuente: Elaboración propia*

○ **Establecer término de interés mediante campo de texto**

Descripción	
Como	Usuario
Necesito	Establecer un término para realizar búsqueda
Para	Cambiar la cabecera que se enviará por HTTP para realizar la consulta.
Criterio de Aceptación	
Dado	Deseo ingresar un término para realizar la búsqueda.
Cuando	Presione el botón de buscar que se encuentra en el Appbar.
Entonces	Se mostrará visible el campo de texto para ingresar término o invisible para observar de mejor manera los resultados.

*Tabla 3.21: Iteración 3- Establecer término de interés mediante campo de texto
Fuente: Elaboración propia*

- **Realizar consulta por API-REST a servicio integrador de bases de datos**

Descripción	
Como	Usuario
Necesito	Realizar una consulta con el término de interés, áreas de búsqueda y número de artículos (los 2 últimos establecidos en la pestaña de parámetros).
Para	Enviar la consulta al servicio por API-REST
Criterio de Aceptación	
Dado	Deseo enviar una búsqueda de revistas científicas una vez establecido el término y filtros.
Cuando	Presione el botón de buscar (en el teclado o integrado en el campo de texto).
Entonces	Enviaré una consulta al servicio por API-REST con los nuevos parámetros establecidos.

*Tabla 3.22: Iteración 3- Realizar consulta por API-REST a servicio integrador de bases de datos
Fuente: Elaboración propia*

- **Mostrar skeleton**

Descripción	
Como	Aplicación
Necesito	Mostrar skeleton mientras se espera la respuesta del servicio
Para	Indicar al usuario que la consulta se está realizando.
Criterio de Aceptación	
Dado	Debo esperar la respuesta del servicio.
Cuando	Envíe una consulta al servicio.
Entonces	Mostrará un skeleton con animación de carga.

*Tabla 3.23: Iteración 3- Mostrar skeleton
Fuente: Elaboración propia*

- **El servicio responde y no existe resultados**

Descripción	
Como	Aplicación
Necesito	Mostrar Widget cuando el servicio respondió, pero no se tiene resultados
Para	Indicar al usuario que la consulta se realizó.
Criterio de Aceptación	
Dado	No existe resultados.
Cuando	El servicio envíe una respuesta.
Entonces	Mostrar un widget con el mensaje "No se encontraron resultados"

*Tabla 3.24: Iteración 3- El servicio responde y no existe resultados
Fuente: Elaboración propia*

○ **El servicio responde y existe resultados**

Descripción	
Como	Aplicación
Necesito	Tener una lista preparada para ser mostrada al usuario.
Para	Tener datos para colocar en los widgets y este sea cómodo a la vista.
Criterio de Aceptación	
Dado	Existe resultados.
Cuando	El servicio envíe una respuesta.
Entonces	Debo buscar la imagen de la biblioteca científica perteneciente a cada revista y añadir a la lista.

Tabla 3.25: Iteración 3- El servicio responde y existe resultados

Fuente: Elaboración propia

○ **Mostrar listado final al usuario**

Descripción	
Como	Aplicación
Necesito	Colocar listado final en Widgets con imagen de carga de biblioteca científica
Para	El usuario observe los resultados de la consulta realizada al servicio integrador de base de datos
Criterio de Aceptación	
Dado	La lista se encuentre con todos los parámetros necesarios
Cuando	La búsqueda de imágenes concluyó
Entonces	Mostrar el listado resultante con una imagen de carga de cada biblioteca mientras carga de internet. Mientras tanto el usuario puede seguir observando los resultados.

Tabla 3.26: Iteración 3- Mostrar listado final al usuario

Fuente: Elaboración propia

○ **Observar revista científica**

Descripción	
Como	Usuario
Necesito	Consultar estado de revista en Scimago Journal & Country Rank.
Para	Observar cuartiles, SJR y sus citaciones de la revista.
Criterio de Aceptación	
Dado	Deseo ver el estado de la revista.
Cuando	Presione en el ítem.
Entonces	Debo abrir la pestaña de revista enviado el issn y eissn del ítem seleccionado.

Tabla 3.27: Iteración 3- Observar revista científica

Fuente: Elaboración propia

Día de trabajo

Los fragmentos de código presentados a continuación hacen énfasis a lo más importante realizado en la *Iteración 3*.

home_page

```
void realizarBusqueda(Parametro parametro) {
  if (_selectedIndex == 0) {
    final Busqueda busqueda = new Busqueda(
      busqueda: parametro.texto,
      startElsevier: 1,
      startSpringer: 1,
      subElsevier: parametro.subElsevier,
      subSpringer: parametro.subSpringer,
      numArticulos: 10);
    _articulosBloc.limpiarArticulosConsulta();
    _articulosBloc.obtenerArticulosConsulta(busqueda);
  } else if (_selectedIndex == 1) {
    final Busqueda busqueda = new Busqueda(
      busqueda: parametro.texto,
      startElsevier: 1,
      startSpringer: 1,
      subElsevier: parametro.subElsevier,
      subSpringer: parametro.subSpringer,
      numArticulos: parametro.numeroArticulos);
    _revistasBloc.limpiarRevistaConsulta();
    _revistasBloc.obtenerRevistasConsulta(busqueda);
  }
}
```

Figura 3.76: Iteración 3- Trabajo (home_page)
Fuente: Elaboración propia

revistas_page

```
class RevistasPage extends StatefulWidget {...}

class _RevistasPageState extends State<RevistasPage> {

  @override
  void initState() {...}

  @override
  void dispose() {...}

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<RevistaConsulta>(
      stream: _revistasBloc.revistasConsultaStream,
      builder: (BuildContext context, AsyncSnapshot<RevistaConsulta> snapshot) {
        if (snapshot.hasData && snapshot.data.revistas != null) {
          if (snapshot.data.revistas.length > 0) {
            return Expanded(
              child: GestureDetector(
                onPanDown: (_) {
                  FocusScope.of(context).requestFocus(new FocusNode());
                },
                child: _revistas(snapshot.data),
              );
          }
          return Padding(
            padding: const EdgeInsets.only(top: 100),
            child: SinResultadosPage(),
          );
        }
        return Expanded(child: obtenerSkeletonRevistas());
      },
    );
  }

  Widget _revistas(RevistaConsulta revistaConsulta) {...}

  Widget _revistaTile(Revista item) {...}
}
```

Figura 3.77: Iteración 3- Trabajo (revistas_page)
Fuente: Elaboración propia

Día de liberación

Demostración del cumplimiento de la iteración según las historias de usuario establecidas.



*Figura 3.78: Iteración 3- Liberación (carga skeleton)
Fuente: Elaboración propia*

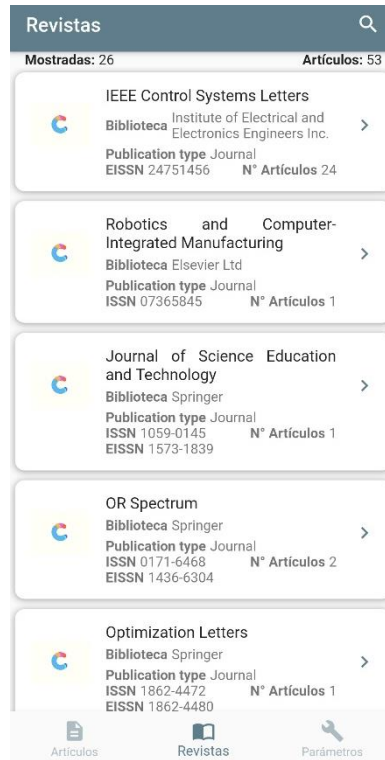


Figura 3.79: Iteración 3- Liberación (carga imágenes bibliotecas virtuales)
Fuente: Elaboración propia



Figura 3.80: Iteración 3- Liberación (búsqueda realizada con campo de texto visible)
Fuente: Elaboración propia

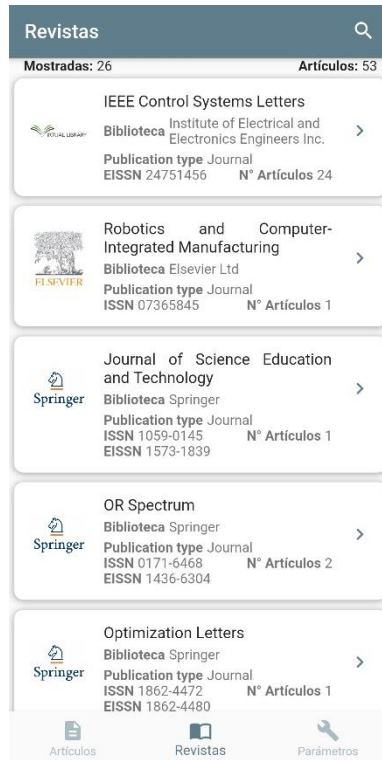


Figura 3.81: Iteración 3- Liberación (búsqueda realizada con campo de texto oculto)
Fuente: Elaboración propia



Figura 3.82: Iteración 3- Liberación (búsqueda sin resultados)
Fuente: Elaboración propia

3.1.4.4 Iteración 4.- Configuración de parámetros

Día de planificación

- **Mostrar artículos según término de interés**

Descripción	
Como	Usuario
Necesito	Ingresar parámetros de búsqueda iniciales
Para	Realizar búsquedas por defecto

Tabla 3.28: Iteración 4- Mostrar artículos según término de interés

Fuente: Elaboración propia

- **Cargar texto por defecto almacenado**

Descripción	
Como	Aplicación
Necesito	Colocar texto por defecto almacenado en la base de datos del dispositivo.
Para	Tener un término de interés y mostrar al usuario
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo.
Cuando	Inicialice la pestaña de Parámetros.
Entonces	Realizará la consulta de texto por defecto a la base de datos del dispositivo y se colocará en el campo de texto.

Tabla 3.29: Iteración 4- Cargar texto por defecto almacenado

Fuente: Elaboración propia

- **Cargar áreas de búsqueda almacenadas**

Descripción	
Como	Aplicación
Necesito	Cargar áreas de búsqueda almacenados en la base de datos del dispositivo
Para	Mostrar al usuario las áreas que se realizarán la búsqueda de artículos y revistas.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo y mostrar al usuario.
Cuando	Inicialice la pestaña de Parámetros.
Entonces	Realizará la consulta de áreas de búsqueda de Elsevier y Springer a la base de datos del dispositivo y se seleccionarán en los dropdowns cargados previamente.

Tabla 3.30: Iteración 4- Cargar áreas de búsqueda almacenadas

Fuente: Elaboración propia

○ **Cargar número de artículos almacenado**

Descripción	
Como	Aplicación
Necesito	Cargar número de artículos a buscar y agrupar almacenado en la base de datos del dispositivo
Para	Mostrar al usuario el número de artículos que se enviará como mínimo para la búsqueda de revistas.
Criterio de Aceptación	
Dado	Deseo obtener información almacenada en la base de datos del dispositivo y mostrar al usuario.
Cuando	Inicialice la pestaña de Parámetros.
Entonces	Realizará la consulta de número de artículos a la base de datos del dispositivo y se seleccionarán en el dropdown cargado previamente.

*Tabla 3.31: Iteración 4- Cargar número de artículos almacenado
Fuente: Elaboración propia*

○ **Cambiar término de interés por defecto**

Descripción	
Como	Usuario
Necesito	Cambiar el término para realizar una búsqueda por defecto
Para	Tener preparado parámetros iniciales en las pestañas de búsqueda.
Criterio de Aceptación	
Dado	Deseo cambiar el término para realizar la búsqueda inicial.
Cuando	Presione el botón de Editar se activará la edición de texto y se deberá aplicar o cancelar los cambios realizados (mediante botones) para observar los cambios.
Entonces	Se actualizará el término de interés por defecto almacenado en la base de datos.

*Tabla 3.32: Iteración 4- Cambiar término de interés por defecto
Fuente: Elaboración propia*

○ **Cambiar número de revistas a buscar y agrupar**

Descripción	
Como	Usuario
Necesito	Cambiar el número de artículos que realizará la búsqueda y agrupación de revistas.
Para	Tener preparado parámetros iniciales en las pestañas de búsqueda.
Criterio de Aceptación	
Dado	Deseo cambiar el número de artículos
Cuando	Presione el dropdown se desplegará una lista y se deberá seleccionar un valor
Entonces	Se actualizará el número de artículos almacenado en la base de datos.

*Tabla 3.33: Iteración 4- Cambiar número de revistas a buscar y agrupar
Fuente: Elaboración propia*

○ **Cambiar áreas de búsqueda**

Descripción	
Como	Usuario
Necesito	Cambiar las áreas que se tomarán para la búsqueda de artículos y revistas.
Para	Tener preparado parámetros iniciales en las pestañas de búsqueda.
Criterio de Aceptación	
Dado	Deseo cambiar las áreas de búsqueda
Cuando	Presione el dropdown de una biblioteca virtual se desplegará una lista y se deberá seleccionar un valor
Entonces	Se actualizará el área de búsqueda de la biblioteca virtual seleccionada almacenada en la base de datos.

*Tabla 3.34: Iteración 4- Cambiar áreas de búsqueda
Fuente: Elaboración propia*

Día de trabajo

El fragmento de código presentado a continuación hace énfasis a lo más importante realizado en la *Iteración 4*.

parametros_page

```

class ParametrosPage extends StatefulWidget {...}

class _ParametrosPageState extends State<ParametrosPage> {
  @override
  void initState() {...}

  @override
  void dispose() {...}

  @override
  Widget build(BuildContext context) {
    _parametrosBloc = widget.parametrosBloc;

    if (countCarga == 0) {
      _parametrosBloc.obtenerParametrosValor();
      countCarga++;
    }

    return SingleChildScrollView(
      child: Column(
        children: <Widget>[
          SizedBox(height: 20.0),
          _busquedaDefecto(context),
          _busquedaArticulos(context),
          _areasBusqueda(context),
          SizedBox(height: 10.0),
        ],
      ),
    );
  }

  Widget _busquedaDefecto(BuildContext context) {...}

  Widget _textfield() {...}

  _botonesEditarTexto(BuildContext context) {...}

  cambiarEstadoTexto() {...}

  void _snackBarCambioTexto() {...}

  _busquedaArticulos(BuildContext context) {...}

  Widget _areasBusqueda(BuildContext context) {...}
  ...

```

Figura 3.83: Iteración 4- Trabajo (parametros_page)
Fuente: Elaboración propia

Día de liberación

Demostración del cumplimiento de la iteración según las historias de usuario establecidas.

The screenshot shows a mobile application interface with three main sections for search parameters:

- Búsqueda por defecto:** A text input field containing the word "programming" and an "Editar" button with a pencil icon.
- Búsqueda de revistas:** A section titled "Búsqueda de revistas:" with a sub-label "Mínimo de artículos a buscar y agrupar". It features a dropdown menu currently set to "50". A red note below reads: "Nota: La velocidad de búsqueda de revistas dependerá del número de artículos seleccionados".
- Áreas de búsqueda:** A section titled "Áreas de búsqueda:" with two dropdown menus. The first is labeled "Elsevier" and has "Todos" selected. The second is labeled "Springer" and also has "Todos" selected.

At the bottom, there is a navigation bar with three icons and labels: "Artículos" (document icon), "Revistas" (book icon), and "Parámetros" (wrench icon).

Figura 3.84: Iteración 4- Liberación (Parámetros)
Fuente: Elaboración propia

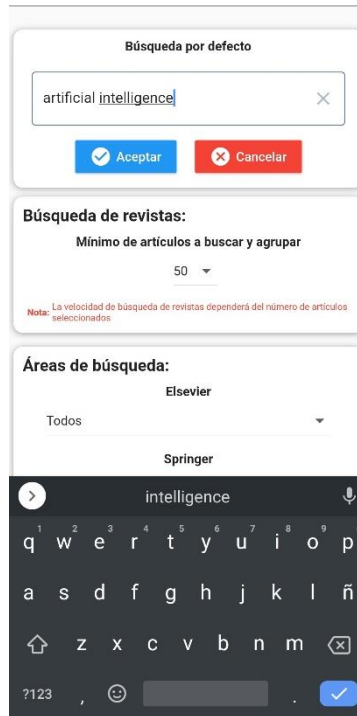


Figura 3.85: Iteración 4- Liberación (Cambio de término por defecto)
Fuente: Elaboración propia



Figura 3.86: Iteración 4- Liberación (Mensaje al actualizar el término de búsqueda)
Fuente: Elaboración propia

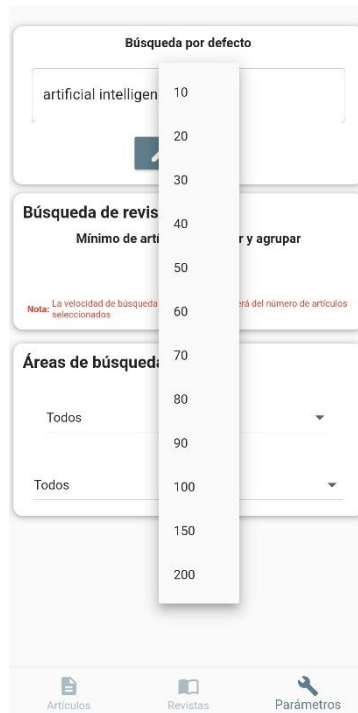


Figura 3.87: Iteración 4- Liberación (Opciones de número de artículos)
 Fuente: Elaboración propia



Figura 3.88: Iteración 4- Liberación (Mensaje de actualización en el número de artículos)
 Fuente: Elaboración propia

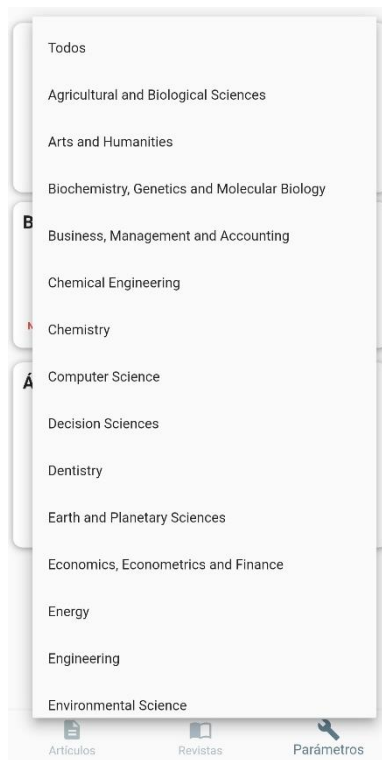


Figura 3.89: Iteración 4- Liberación (Opciones de área de búsqueda en Elsevier)
Fuente: Elaboración propia

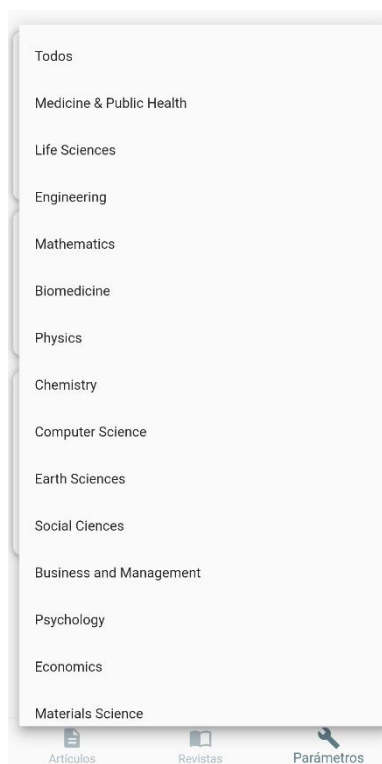


Figura 3.90: Iteración 4- Liberación (Opciones de área de búsqueda en Springer)
Fuente: Elaboración propia

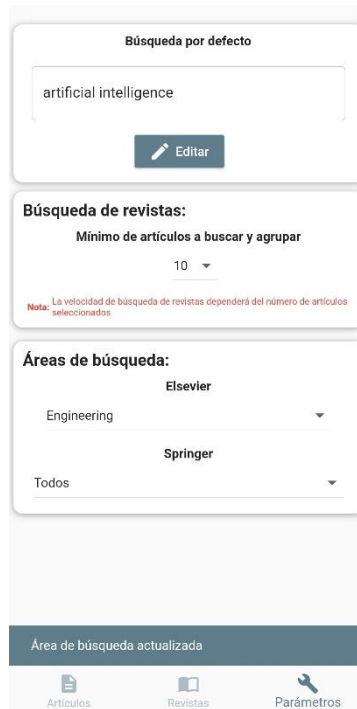


Figura 3.91: Iteración 4- Liberación (Mensaje de actualización en el área de búsqueda)
Fuente: Elaboración propia

3.1.4.5 Iteración 5.- Consulta de revista

Día de planificación

- **Mostrar consulta de revista**

Descripción	
Como	Aplicación
Necesito	Consultar en la página de Scimago Journal & Country Rank
Para	Mostrar los resultados sobre la revista al usuario

Tabla 3.35: Iteración 5- Mostrar consulta de revista
Fuente: Elaboración propia

- **Recibir parámetros enviados**

Descripción	
Como	Aplicación
Necesito	Obtener los parámetros enviados desde la pestaña de Revistas.
Para	Consultar por issn o eissn al servicio integrador.
Criterio de Aceptación	
Dado	Deseo obtener parámetros enviados al presionar una revista.
Cuando	Inicialice la pestaña de Revista.
Entonces	Se tomará los parámetros enviados y se almacenarán en una variable local.

Tabla 3.36: Iteración 5- Recibir parámetros enviados
Fuente: Elaboración propia

○ **Consultar por issn o eissn**

Descripción	
Como	Aplicación
Necesito	Consultar al servicio integrador por API-REST
Para	Verificar si la revista se encuentra en la base de Scimago Journal & Country Rank
Criterio de Aceptación	
Dado	Deseo obtener información respectiva a una revista.
Cuando	Obtenga los parámetros enviados
Entonces	Se consultará al servicio por issn, existirá 3 casos a realizar: 1.- Si la respuesta contiene datos se almacenará para presentar al usuario. 2.- Si no existe respuesta, se consultará por eissn. 3.- Si una vez realizado el caso 2 la respuesta persiste, se colocará como respuesta "No existe resultados".

*Tabla 3.37: Iteración 5- Consultar por issn o eissn
Fuente: Elaboración propia*

○ **Mostrar respuesta**

Descripción	
Como	Aplicación
Necesito	Colocar la respuesta de la consulta al servicio en Widgets.
Para	Mostrar al usuario la respuesta en una interfaz agradable a la vista.
Criterio de Aceptación	
Dado	Deseo mostrar la respuesta al usuario
Cuando	Recibo una respuesta final a la consulta al servicio
Entonces	Se colocará la respuesta en Widgets y según el cuartil se colocará un color (tomando como referencia al establecido en Scimago).

*Tabla 3.38: Iteración 5- Mostrar respuesta
Fuente: Elaboración propia*

○ **Mostrar artículos por revista**

Descripción	
Como	Usuario
Necesito	Enviar la Revista (issn o eissn) a la pestaña de Artículos por Revista
Para	Consultar Artículos según la revista enviada.
Criterio de Aceptación	
Dado	Deseo ver los artículos según la revista consultada.
Cuando	Presione el botón Artículos.
Entonces	Debo abrir la pestaña de Artículos por Revista enviado el issn y eissn del ítem seleccionado.

*Tabla 3.39: Iteración 5- Mostrar artículos por revista
Fuente: Elaboración propia*

- **Mostrar artículos por revista**

Descripción	
Como	Usuario
Necesito	Obtener los parámetros enviados desde la pestaña de Revistas.
Para	Consultar por issn o eissn al servicio integrador.
Criterio de Aceptación	
Dado	Deseo obtener parámetros enviados al presionar una revista.
Cuando	Inicialice la pestaña de Revista.
Entonces	Se tomará los parámetros enviados y se almacenarán en una variable local.

*Tabla 3.40: Iteración 5- Mostrar artículos por revista
Fuente: Elaboración propia*

Día de trabajo

El fragmento de código presentado a continuación hace énfasis a lo más importante realizado en la *Iteración 5*.

revista_page

```

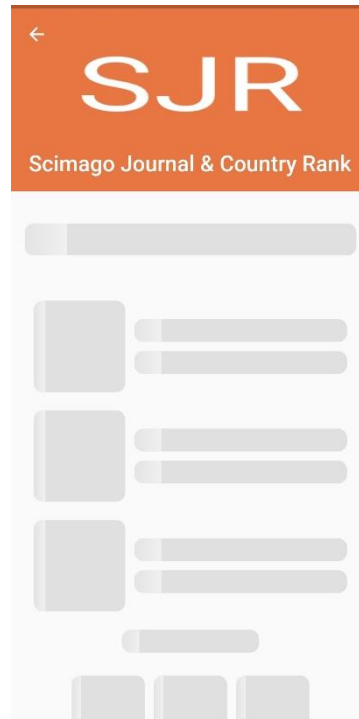
class _RevistaPageState extends State<RevistaPage> {
  @override
  Widget build(BuildContext context) {
    final RevistaEnvio revista = ModalRoute.of(context).settings.arguments;
    if (revista.issn == '' || comprobacion) {
      _valorBusqueda = revista.eissn;
    } else {
      _valorBusqueda = revista.issn;
    }
    if (count == 0) {
      _future = _revistaProvider.getRevista(_valorBusqueda);
      _future.then((Revista value) {
        if (value.quartile.length == 0 && comprobacion == false) {
          setState(() {
            if (revista.eissn != '') {
              _revistaProvider.status = Future.value(0);
              comprobacion = true;
            }
          });
        }
      });
    } else {
      _future = null;
    }
    return Scaffold(
      body: GlowingOverscrollIndicator(
        axisDirection: AxisDirection.down,
        color: Colors.deepOrange,
        child: CustomScrollView(
          slivers: <Widget>[
            _crearAppBar(revista),
            SliverList(
              delegate: SliverChildListDelegate(
                [
                  SizedBox(height: 10.0),
                  _bodyPage(revista, _valorBusqueda),
                  SizedBox(height: 10.0),
                ],
              ),
            ),
          ],
        ),
      ),
    );
  }
  ...
}

```

*Figura 3.92: Iteración 5- Trabajo (revista_page)
Fuente: Elaboración propia*

Día de liberación

Demostración del cumplimiento de la iteración según las historias de usuario establecidas.



*Figura 3.93: Iteración 5- Liberación (Carga de skeleton)
Fuente: Elaboración propia*



Figura 3.94: Iteración 5- Liberación (Respuesta con resultados)
 Fuente: Elaboración propia



Figura 3.95: Iteración 5- Liberación (Respuesta sin resultados)
 Fuente: Elaboración propia

3.1.4.6 Iteración 6.- Búsqueda de artículos por issn o eissn

Día de planificación

- **Mostrar artículos por issn**

Descripción	
Como	Usuario
Necesito	Obtener un listado de artículos mostrando: título, biblioteca, doi (si posee), tipo de publicación y de contenido; según issn enviado por la pestaña de revista.
Para	Visualizar artículos científicos pertenecientes a varias bibliotecas virtuales.

*Tabla 3.41: Iteración 6- Mostrar artículos por issn
Fuente: Elaboración propia*

- **Recibir parámetros enviados**

Descripción	
Como	Aplicación
Necesito	Obtener los parámetros enviados desde la pestaña de Revista.
Para	Consultar por issn o eissn al servicio integrador.
Criterio de Aceptación	
Dado	Deseo obtener parámetros enviados al presionar el botón Artículos en la pestaña de Revista.
Cuando	Inicialice la pestaña de Artículos por Revista.
Entonces	Se tomará como variable de consulta principal a issn (en el caso de no existir será eissn) y será almacenado en una variable local.

*Tabla 3.42: Iteración 6- Recibir parámetros enviados
Fuente: Elaboración propia*

- **Realizar consulta por API-REST a servicio integrador de bases de datos**

Descripción	
Como	Usuario
Necesito	Realizar una consulta con el issn o eissn (enviado desde la pestaña de revista).
Para	Enviar la consulta al servicio por API-REST
Criterio de Aceptación	
Dado	Deseo enviar una búsqueda de artículos científicos una vez establecido el issn o eissn
Cuando	Ingresa a la pestaña de Artículos por issn o eissn
Entonces	Enviará una consulta al servicio por API-REST con los parámetros establecidos.

*Tabla 3.43: Iteración 6- Realizar consulta por API-REST a servicio integrador de bases de datos
Fuente: Elaboración propia*

- **Mostrar skeleton**

Descripción	
Como	Aplicación
Necesito	Mostrar skeleton mientras se espera la respuesta del servicio
Para	Indicar al usuario que la consulta se está realizando.
Criterio de Aceptación	
Dado	Debo esperar la respuesta del servicio.
Cuando	Envíe una consulta al servicio.
Entonces	Mostrará un skeleton con animación de carga.

*Tabla 3.44: Iteración 6- Mostrar skeleton
Fuente: Elaboración propia*

- **El servicio responde**

Descripción	
Como	Aplicación
Necesito	Colocar listado en Widgets.
Para	El usuario observe los resultados de la consulta realizada al servicio integrador de base de datos
Criterio de Aceptación	
Dado	Existe resultados.
Cuando	El servicio envíe una respuesta.
Entonces	Mostrar el listado resultante para que el usuario pueda seguir observando los resultados.

*Tabla 3.45: Iteración 6- El servicio responde
Fuente: Elaboración propia*

- **Observar artículo científico**

Descripción	
Como	Usuario
Necesito	Ingresa al sitio donde se encuentra el artículo científico.
Para	Observar más detalles.
Criterio de Aceptación	
Dado	Deseo ver con detalle el artículo.
Cuando	Presione en el ítem.
Entonces	Debo redirigir al navegador por el url del ítem seleccionado.

*Tabla 3.46: Iteración 6- Observar artículo científico
Fuente: Elaboración propia*

- **Cargar los siguientes artículos**

Descripción	
Como	Usuario
Necesito	Cargar y agregar el siguiente listado de revistas al existente.
Para	Observar los siguientes resultados por el issn o eissn.
Criterio de Aceptación	
Dado	Deseo ver los siguientes resultados.
Cuando	El scroll llegue al final de la lista.
Entonces	Debo cargar los siguientes resultados por issn o eissn.

*Tabla 3.47: Iteración 6- Cargar los siguientes artículos
Fuente: Elaboración propia*

Día de trabajo

El fragmento de código presentado a continuación hace énfasis a lo más importante realizado en la *Iteración 6*.

articulos_issn_page


```

class ArticulosIssnPage extends StatefulWidget {...}

class _ArticulosIssnPageState extends State<ArticulosIssnPage> {
  @override
  Widget build(BuildContext context) {
    final RevistaEnvio revista = ModalRoute.of(context).settings.arguments;
    library = revista.library;
    if (revista.issn == '') {
      _valorBusqueda = revista.eissn;
    } else {
      _valorBusqueda = revista.issn;
    }
    return Scaffold(
      resizeToAvoidBottomPadding: true,
      appBar: AppBar(
        title: Text(
          'Artículos',
          overflow: TextOverflow.ellipsis,
        )),
      body: Stack(
        children: <Widget>[
          _cargarLista(_valorBusqueda),
          _crearLoading(),
        ],
      ),
    );
  }

  Widget _consultaVerificacion() {...}

  Widget _articulosBuscados() {...}

  Widget _elementList(Articulo item) {...}

  dynamic _totalResults() async {...}

  dynamic _totalArticles() async {...}

  void cargarListaAsync(startElsevier, startScopus, library) async {...}

  Widget _crearLoading() {...}

  Widget _cargarLista(String valorBusqueda) {...}
}

```

Figura 3.96: Iteración 6- Trabajo (articulos_issn_page)
Fuente: Elaboración propia

Día de liberación

Demostración del cumplimiento de la iteración según las historias de usuario establecidas.



*Figura 3.97: Iteración 6- Liberación (carga skeleton)
Fuente: Elaboración propia*

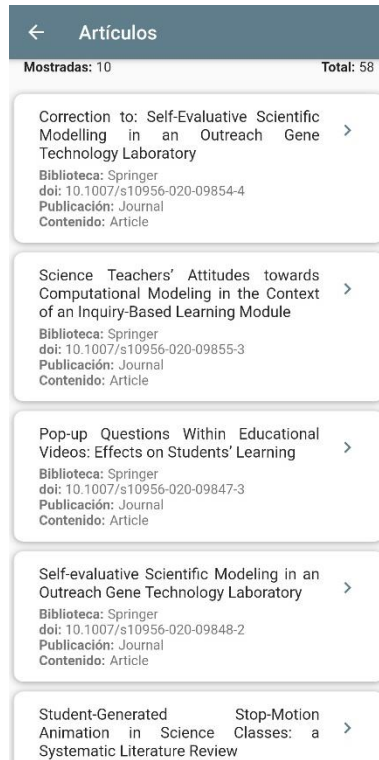


Figura 3.98: Iteración 6- Liberación (búsqueda realizada)
Fuente: Elaboración propia

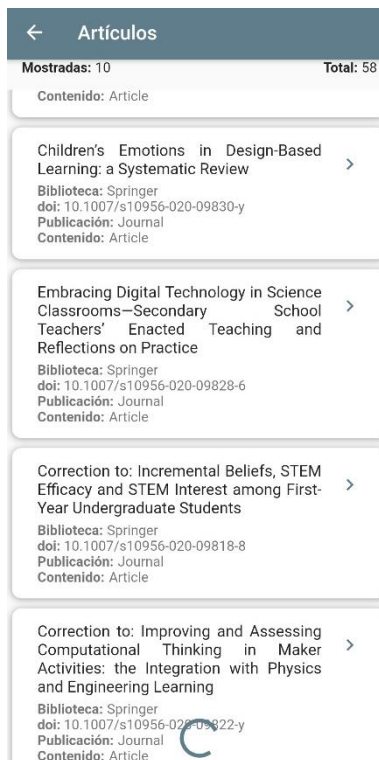


Figura 3.99: Iteración 6- Liberación (cargando siguientes artículos con scroll)
Fuente: Elaboración propia

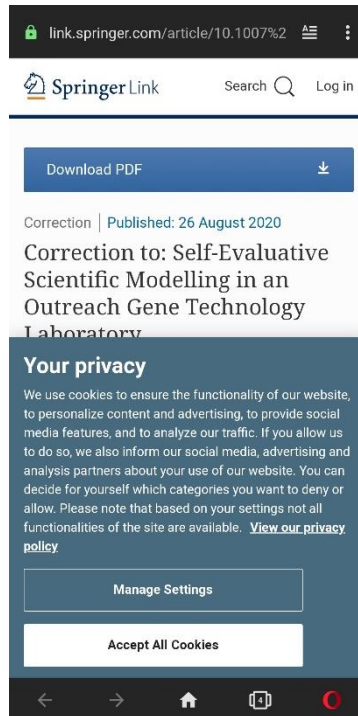


Figura 3.100: Iteración 6- Liberación (redirección al presionar artículo)
Fuente: Elaboración propia

3.1.5 Fase de estabilización

En esta fase se generó un archivo en la aplicación móvil de ruta general para que cada proveedor consulte la dirección del servidor a la que deberá realizar la petición (Figura 3.101).

```
String ruta = 'http://192.168.1.100:4000';
```

Figura 3.101: Aplicación móvil- Ruta general
Fuente: Elaboración propia

En el servidor se estableció un parámetro para que la dirección pueda ser cambiada (si se utiliza flask) (Figura 3.102).

```
if __name__ == '__main__':  
    app.run(host = '192.168.1.100' , debug = True, port = 4000)
```

Figura 3.102: Establecimiento de dirección en flask
Fuente: Elaboración propia

3.1.6 Fase de prueba y reparación

El problema evidenciado en esta fase fue en las consultas por defecto que se realizaban excesiva e innecesariamente al intercambiar pestañas por lo tanto se colocó un tiempo de 2 segundos de permanencia en la pestaña seleccionada para enviar la petición al servidor, mostrado en la *Figura 3.103*.

```
void realizarBusqueda(Parametro parametro) {
    parametro.texto = _textController.text;
    if (_debounce?.isActive ?? false) _debounce.cancel();
    _debounce = Timer(const Duration(milliseconds: 2000), () {
        if (_selectedIndex == 0) {
            final Busqueda busqueda = new Busqueda(
                busqueda: parametro.texto,
                startElsevier: 1,
                startSpringer: 1,
                subElsevier: parametro.subElsevier,
                subSpringer: parametro.subSpringer,
                numArticulos: 10);
            _articulosBloc.limpiarArticulosConsulta();
            _articulosBloc.obtenerArticulosConsulta(busqueda);
        } else if (_selectedIndex == 1) {
            final Busqueda busqueda = new Busqueda(
                busqueda: parametro.texto,
                startElsevier: 1,
                startSpringer: 1,
                subElsevier: parametro.subElsevier,
                subSpringer: parametro.subSpringer,
                numArticulos: parametro.numeroArticulos);
            _revistasBloc.limpiarRevistaConsulta();
            _revistasBloc.obtenerRevistasConsulta(busqueda);
        }
    });
}
```

Figura 3.103: Debounce en intercambio de pestañas
Fuente: Elaboración propia

3.1.7 Recolección de información

Una vez con el software terminado, para evaluar el impacto del sistema de recomendación implementado se realizó una encuesta y se obtuvo como resultado lo siguiente:

1. Creo que me gustaría utilizar este sistema con frecuencia.

10 respuestas

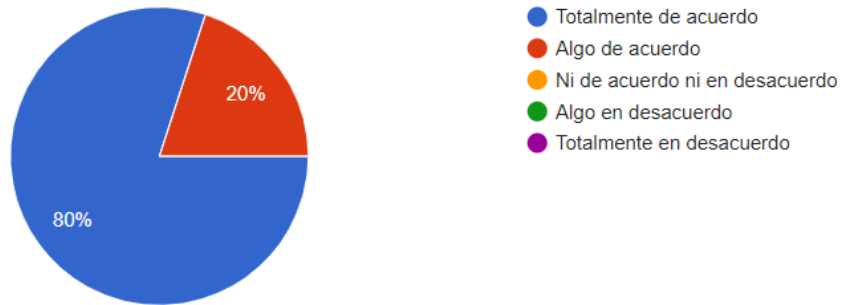


Figura 3.104: Pregunta 1
Fuente: Elaboración propia

2. Encontré el sistema innecesariamente complejo.

10 respuestas

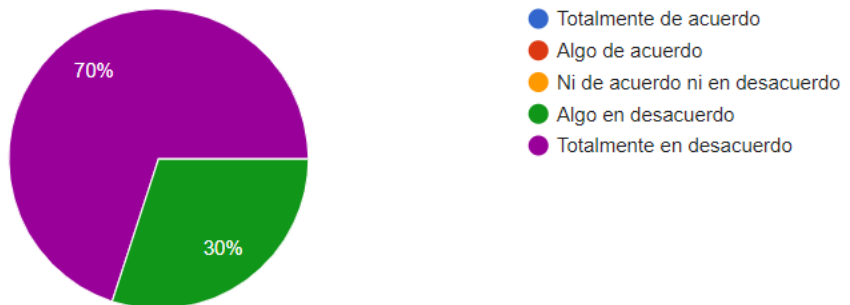


Figura 3.105: Pregunta 2
Fuente: Elaboración propia

3. Pensé que el sistema era fácil de usar.

10 respuestas

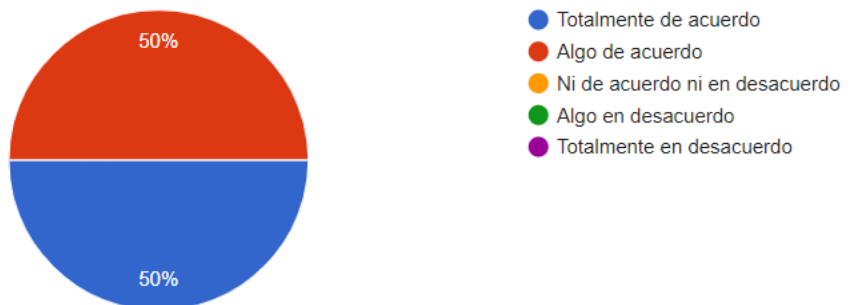


Figura 3.106: Pregunta 3
Fuente: Elaboración propia

4. Creo que necesitaría el apoyo de una persona con conocimientos técnicos para poder utilizar este sistema.

10 respuestas

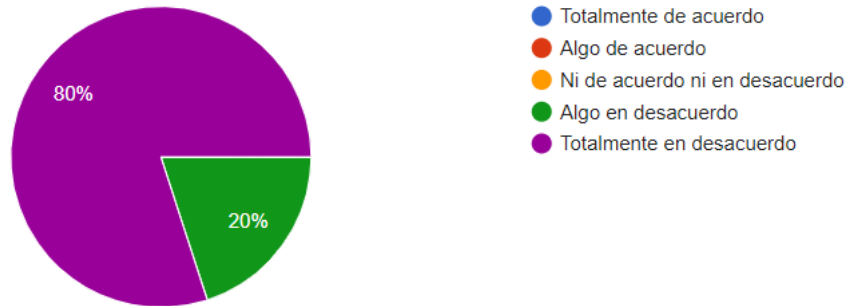


Figura 3.107: Pregunta 4
Fuente: Elaboración propia

5. Encontré que las diversas funciones de este sistema estaban bien integradas.

10 respuestas

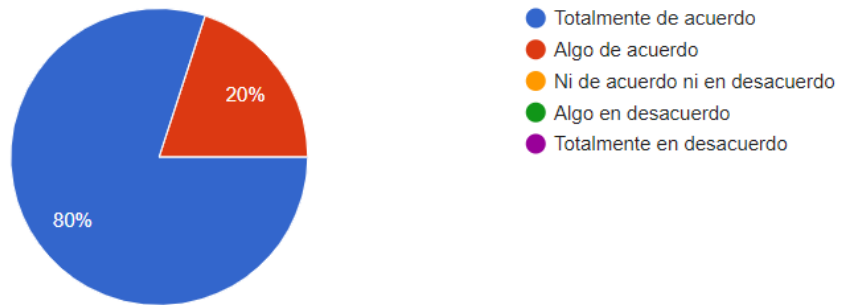


Figura 3.108: Pregunta 5
Fuente: Elaboración propia

6. Pensé que había demasiada incoherencia en este sistema.

10 respuestas

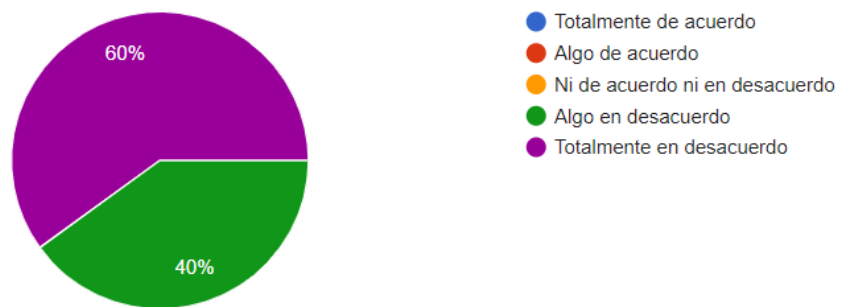


Figura 3.109: Pregunta 6
Fuente: Elaboración propia

7. Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.

10 respuestas

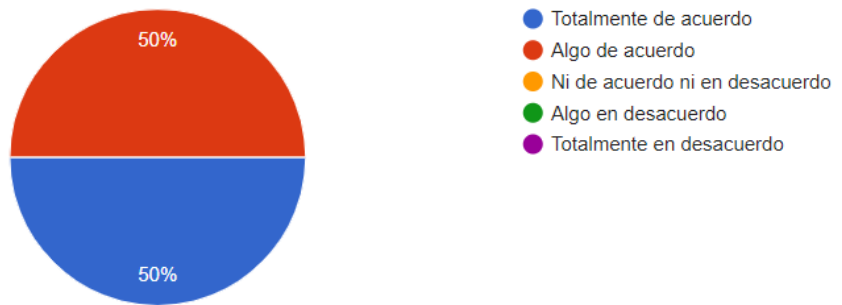


Figura 3.110: Pregunta 7
Fuente: Elaboración propia

8. Me pareció que el sistema es muy difícil de usar.

10 respuestas

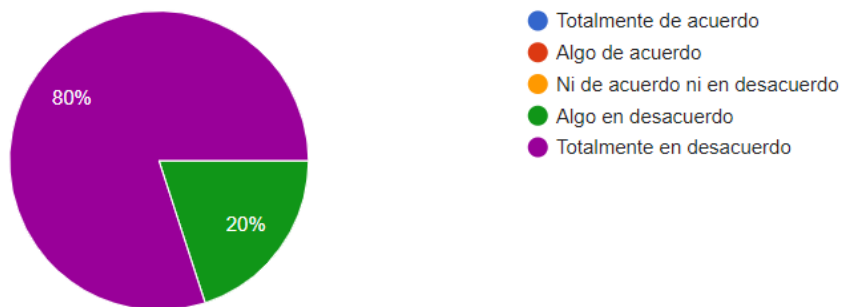


Figura 3.111: Pregunta 8
Fuente: Elaboración propia

9. Me sentí muy seguro de usar el sistema.

10 respuestas

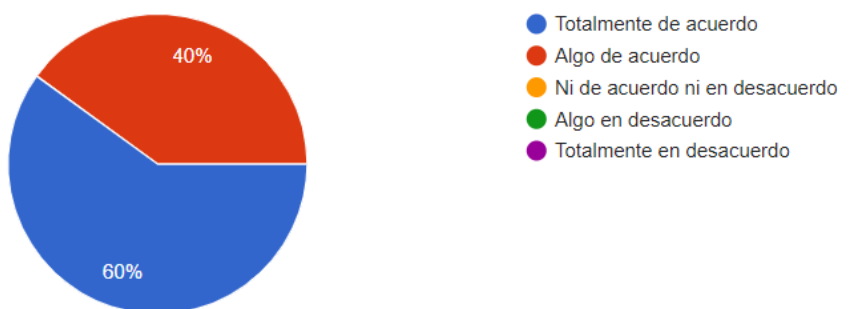


Figura 3.112: Pregunta 9
Fuente: Elaboración propia

10. Necesitaba aprender muchas cosas antes de poder seguir este sistema.

10 respuestas

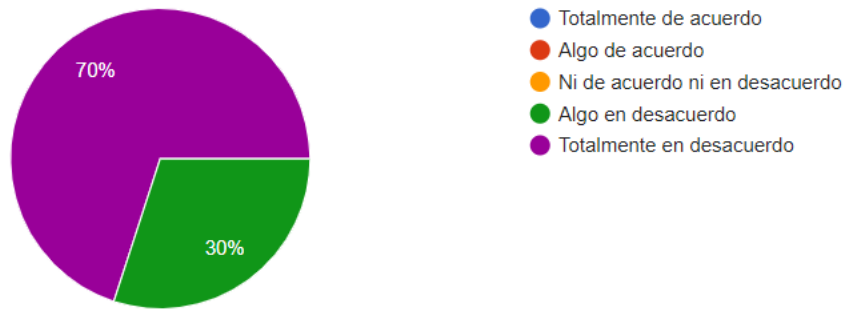


Figura 3.113: Pregunta 10
Fuente: Elaboración propia

3.1.8 Procesamiento y análisis de datos

Utilizando el cálculo por SUS para obtener el nivel de aceptabilidad de la aplicación se obtuvo lo siguiente:

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
		Suma	40	
		Valor(Suma*2,5)	100	

Figura 3.114: Cálculo de SUS en encuesta 1
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
		Suma	30	
	Valor(Suma*2,5)		75	

Figura 3.115: Cálculo de SUS en encuesta 2
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	4	1	3	
	1	5	4	
		Suma	39	
	Valor(Suma*2,5)		97,5	

Figura 3.116: Cálculo de SUS en encuesta 3
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
		Suma	40	
	Valor(Suma*2,5)		100	

Figura 3.117: Cálculo de SUS en encuesta 4
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	2	5	3	
	4	1	3	
	1	5	4	
	5	1	4	
	2	5	3	
	4	1	3	
	1	5	4	
	4	1	3	
	1	5	4	
		Suma	35	
		Valor(Suma*2,5)	87,5	

Figura 3.118: Cálculo de SUS en encuesta 5
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
	4	1	3	
	2	5	3	
		Suma	30	
		Valor(Suma*2,5)	75	

Figura 3.119: Cálculo de SUS en encuesta 6
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	4	1	3	
	1	5	4	
	5	1	4	
	2	5	3	
	4	1	3	
	1	5	4	
	5	1	4	
	2	5	3	
		Suma	36	
		Valor(Suma*2,5)	90	

Figura 3.120: Cálculo de SUS en encuesta 7
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
		Suma	40	
	Valor(Suma*2,5)		100	

Figura 3.121: Cálculo de SUS en encuesta 8
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	4	1	3	
	1	5	4	
	5	1	4	
	1	5	4	
	4	1	3	
	1	5	4	
	5	1	4	
	1	5	4	
		Suma	38	
	Valor(Suma*2,5)		95	

Figura 3.122: Cálculo de SUS en encuesta 9
Fuente: Elaboración propia

	Selección	Valor cálculo	Total	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
	5	1	4	
	1	5	4	
		Suma	40	
	Valor(Suma*2,5)		100	

Figura 3.123: Cálculo de SUS en encuesta 10
Fuente: Elaboración propia

			Puntaje		
		Encuesta 1	100		
		Encuesta 2	75		
		Encuesta 3	97,5		
		Encuesta 4	100		
		Encuesta 5	87,5		
		Encuesta 6	75		
		Encuesta 7	90		
		Encuesta 8	100		
		Encuesta 9	95		
		Encuesta 10	100		
		Promedio	92		

Figura 3.124: Promedio de cálculo de SUS con encuestas recolectadas

Fuente: Elaboración propia

Para establecer si la aplicación está a un nivel aceptable se debe mencionar que SUS considera el valor de 68 como media; por lo tanto, se puede evidenciar:

- Al tener un promedio de 92 se puede decir que el desarrollo de la aplicación fue satisfactorio y cumple con los procesos establecidos.
- En las preguntas 3 y 7 se puede observar que si no se tiene un conocimiento pleno sobre el uso de las bases de datos utilizadas se tendrá dificultad para entender el uso de la aplicación.
- En la pregunta 4 se observó que para el uso de la aplicación no es necesaria la ayuda de una persona con conocimientos técnicos.
- La aplicación estaba correctamente diseñada para realizar el procedimiento que estaba destinado.
- Con práctica y constate uso la aplicación sería de mucha utilidad y se utilizaría con frecuencia.

3.1.9 Muestra del funcionamiento de la aplicación

Para su disponibilidad la aplicación se muestra en el siguiente enlace:

<http://aplicacion.flutterappdrobayo.com/Bosisstant.apk>

Las figuras mostradas a continuación mostrarán el funcionamiento de la aplicación, una vez cumplidas todas las iteraciones de la metodología establecida.

Parámetros de búsqueda

13:00 Término que será utilizado al inicio de la aplicación

Búsqueda por defecto

artificial intelligence

Editar

Búsqueda de revistas:

Mínimo de artículos a buscar y agrupar

Cantidad de artículos que se agruparán en la búsqueda de revistas → 50

Nota: La velocidad de búsqueda de revistas dependerá del número de artículos seleccionados

Áreas de búsqueda:

Elsevier

Computer Science

Singer

Engineering

Áreas de búsqueda que permite cada biblioteca virtual

Área de búsqueda actualizada

Artículos Revistas Parámetros

Figura 3.125: Funcionamiento de la aplicación.- Parámetros de búsqueda
Fuente: Elaboración propia

Búsqueda de artículos

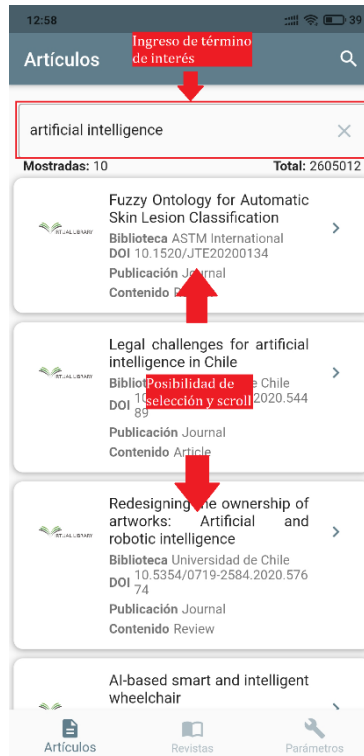


Figura 3.126: Funcionamiento de la aplicación.- Búsqueda de artículos
Fuente: Elaboración propia

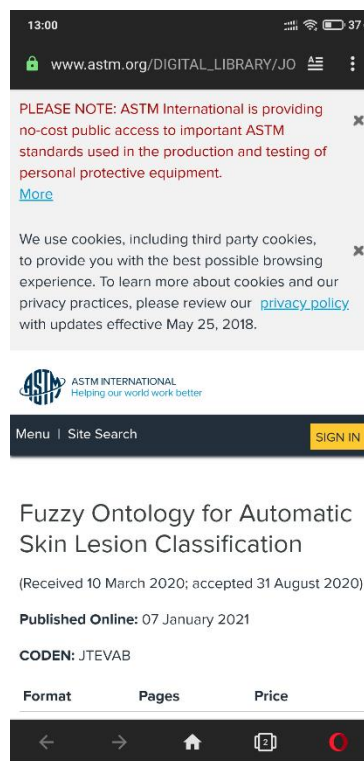


Figura 3.127: Funcionamiento de la aplicación.- Selección de un artículo determinado
Fuente: Elaboración propia

Búsqueda de revistas



Figura 3.128: Funcionamiento de la aplicación.- Búsqueda de revistas
Fuente: Elaboración propia



Figura 3.129: Funcionamiento de la aplicación.- Información relevante según revista seleccionada
Fuente: Elaboración propia



Figura 3.130: Funcionamiento de la aplicación.- Artículos de revista seleccionada
Fuente: Elaboración propia



Figura 3.131: Funcionamiento de la aplicación.- Selección de un artículo determinado
Fuente: Elaboración propia

CAPÍTULO IV.-

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Las tecnologías para el acceso y procesamiento de la información contenidas en bases de datos de publicaciones científicas fueron API-REST y Web Scraping.
- El mecanismo utilizado para realizar la recomendación de publicaciones científicas y espacios de publicación científica fueron la toma de un término de interés y un área específica (para cada biblioteca virtual). Para recomendar espacios de publicación se pedirá adicionalmente el número de publicaciones que se buscarán y agruparán. Para las 2 situaciones los datos deberán ser suministrados por el usuario.
- Se obtuvieron resultados favorables en el procesamiento de la encuesta que se aplicó para determinar el nivel de usabilidad de la aplicación propuesta utilizando SUS.
- Fue posible acceder y procesar la información procedente de los documentos indexados en Elsevier, Springer y Scimago Journal & Country Rank.
- Utilizando los datos recuperados, la aplicación es capaz de sugerir artículos y fuentes de publicación científica, a partir de los términos de interés proporcionados por el usuario.
- De las 8 bibliotecas virtuales investigadas; 4 permiten consultar únicamente desde su sitio web, 3 se necesita tener una suscripción activa para acceder y 2 (Elsevier y Springer) brindaron metadatos de forma privada y de acceso abierto teniendo ciertas restricciones en las respuestas en cada caso de consulta.
- Elsevier ofrece la respuesta a sus metadatos, pero en formato estándar (hasta la finalización de esta investigación no se encontró la forma de recibir la respuesta en formato completo) por lo tanto se optó por redirigir al sitio en el cual se encuentra alojado el artículo para no tener disparidad en la muestra de resultados al usuario.
- El paquete BeautifulSoup facilitó la toma de datos de las páginas web consumidas, pudiendo filtrar por clases, id, y otros aspectos característicos del html del elemento a extraer.
- Para realizar web scraping es necesario tener conocimiento en html para poder inspeccionar la página web y ubicar el elemento a extraer de la página web. La dificultad de programar un web scraping dependerá de cómo se encuentre programado el sitio a tomar los datos.

- Al trabajar con hot reload (suministrado por Flutter), el tiempo en el diseño de la interfaz visual se vio considerablemente reducido al no perder los datos obtenidos previamente y así no tener que esperar respuestas de consultas para observar el diseño con datos reales.

4.2 Recomendaciones

- Se debe realizar un control periódico en el consumo de las APIs de las bibliotecas puesto que no se conocerá el momento en el cual se pueda realizar una actualización de dirección de consumo o la respuesta a una consulta puede cambiar su formato ocasionando que la aplicación y el servicio desarrollados dejen de funcionar.
- Tener un monitoreo constante a los sitios web realizados web scraping ya que si actualizan sus diseños el *bot* no encontrará los elementos tomados anteriormente y el servicio devolverá errores que podrían ser interpretados como errores en el desarrollo.
- Reducir y optimizar las consultas realizadas especialmente porque se trata de consumos a servicios externos. Para el caso del web scraping si son excesivas las peticiones serán consideradas como amenazas para el proveedor del sitio y podrá bloquearlas.

BIBLIOGRAFÍA

- [1] L. Aggarwal, P. Goswami y S. Sachdeva, «ScienceDirect,» 29 12 2020. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494620309947>. [Último acceso: 04 02 2021].
- [2] Z. Zhai, J. F. Martínez, V. Beltran y N. L. Martínez, «ScienceDirect,» 08 02 2020. [En línea]. Available: (<https://www.sciencedirect.com/science/article/pii/S0168169919316497>). [Último acceso: 04 02 2021].
- [3] E. Walling y C. Vaneeckhaute, «ScienceDirect,» 29 04 2020. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S0301479720304473>. [Último acceso: 04 02 2021].
- [4] E. Walid Ghobar, «Universitat Politècnica de València,» 08 01 2018. [En línea]. Available: <https://docs.google.com/viewerng/viewer?url=https://riunet.upv.es:443/bitstream/handle/10251/94049/WALID+-+Un+sis+tema+de+recomendación+basado+en+perfiles+generados+p+or+agrupamiento+y+asociaciones.pdf?sequence%3D1&isAllowed=y>. [Último acceso: 21 11 2019].
- [5] G. M. Cedeño Anchundia, «Universidad de Guayaquil,» 2018. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/27809/1/B-CINT-PTG-N.276%20Cedeño%20Anchundia%20Geovanny%20Michel.pdf>. [Último acceso: 21 11 2019].
- [6] J. C. Hurtado Jerves y B. A. Zabala Bustamante, «ESPOCH,» 04 2018. [En línea]. Available: <http://dspace.esPOCH.edu.ec/bitstream/123456789/9107/1/18T00737.pdf>. [Último acceso: 21 11 2019].
- [7] B. Jennifer, A. Katelynn, B. Jessica, B. Monica, E. Cathleen, G. Josie-Lee, C. Debbie Hewitt, I. Andrea, L. Dee, S. Denise y M. Katherine S, «ScienceDirect,» 26 11 2020. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S1525861020309919>. [Último acceso: 04 02 2021].
- [8] C. G. Porcel Gallego y E. Herrera Viedma, «Revisión de los sistemas de recomendaciones para la recuperación de información,» *Tendencias de investigación en organización del conocimiento*, pp. 507-514, 2003.

- [9] E. Herrera Viedma, C. Porcel y L. Hidalgo, «Universitat Pompeu Fabra,» 02 08 2019. [En línea]. Available: <https://www.upf.edu/hipertextnet/numero-2/recomendacion.html>. [Último acceso: 19 09 2020].
- [10] E. G. Maida y J. Pacienza, «Universidad Católica Argentina,» 2015. [En línea]. Available: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>. [Último acceso: 23 09 2020].
- [11] L. Calabria y P. Píriz, «Universidad ORT de Uruguay,» 2003. [En línea]. Available: https://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia_xp.pdf. [Último acceso: 23 09 2020].
- [12] J. BAUTISTA, «Ingeniería de software,» 2006. [En línea]. Available: <http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACIÓN%20EXTREMA.pdf>. [Último acceso: 23 09 2020].
- [13] J. L. Vila Grau, «Proagilist,» 08 07 2016. [En línea]. Available: <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>. [Último acceso: 08 02 2021].
- [14] P. Blanco, J. Camarero, A. Fumero, A. Warterski y P. Rodríguez, «Adam Westerski,» 2009. [En línea]. Available: http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf. [Último acceso: 23 09 2020].
- [15] J. Nolasco, «DoApps Blog,» 10 10 2018. [En línea]. Available: <https://doapps.me/blog/aplicaciones-nativas-o-hibridas-cual-elegir/>. [Último acceso: 25 09 2020].
- [16] Universidad Interamericana para el Desarrollo, «Universidad Interamericana para el Desarrollo,» [En línea]. Available: https://moodle2.unid.edu.mx/dts_cursos_mdl/lic/TIC/AMWA/S02/AMWA02_Lectura.pdf. [Último acceso: 25 09 2020].
- [17] tedexis, «tedexis,» 06 01 2018. [En línea]. Available: <http://tedexis.com/wp/blog/aplicaciones-nativas-vs-aplicaciones-hibridas/>. [Último acceso: 25 09 2020].
- [18] A. García, «Medium,» 24 11 2016. [En línea]. Available: <https://medium.com/monoku/react-native-está-cambiando-la-forma-en-la-que-desarrollamos-aplicaciones-móviles-de7bbb742555>. [Último acceso: 25 09 2020].

- [19] H. Hernández, «Slashmobility,» 21 04 2020. [En línea]. Available: <https://slashmobility.com/blog/2020/04/todo-lo-que-deberias-saber-sobre-flutter/>. [Último acceso: 25 09 2020].
- [20] J. M. Márquez, «Codigo Base,» 06 05 2019. [En línea]. Available: <https://codigobase.es/flutter-vs-react-native/>. [Último acceso: 08 02 2021].
- [21] Unir, «Unir,» 29 11 2019. [En línea]. Available: <https://www.unir.net/ingenieria/revista/noticias/lenguaje-r-big-data/549204698515/>. [Último acceso: 26 09 2020].
- [22] Python Software Foundation, «Python,» 26 09 2020. [En línea]. Available: <https://docs.python.org/es/3/tutorial/index.html>. [Último acceso: 26 09 2020].
- [23] Equipo de Expertos Universidad Internacional de Valencia, «Universidad Internacional de Valencia,» 25 04 2019. [En línea]. Available: <https://www.universidadviu.com/python-para-big-data-motivos-para-elegirlo/>. [Último acceso: 26 09 2020].
- [24] Unir, «Unir,» 26 03 2020. [En línea]. Available: <https://www.unir.net/ingenieria/revista/noticias/r-vs-python/549204934214/>. [Último acceso: 27 09 2020].
- [25] C. Vega, «Medium,» 23 03 2019. [En línea]. Available: <https://medium.com/comunidad-flutter/implementa-arquitectura-a-tu-proyecto-flutter-usando-el-patrón-bloc-2cb031722166>. [Último acceso: 08 02 2021].
- [26] M. A. Alvarez, «Desarrollo Web,» 28 07 2020. [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-mvc.html>. [Último acceso: 08 02 2021].
- [27] «GitBook,» [En línea]. Available: <https://juanda.gitbooks.io/webapps/content/api/arquitectura-api-rest.html>. [Último acceso: 19 09 2020].
- [28] JetBrains, «Jetbrains,» 2020. [En línea]. Available: <https://www.jetbrains.com/lp/devecosystem-2020/>. [Último acceso: 25 09 2020].
- [29] M. Martí, «Sitelabs,» 08 04 2016. [En línea]. Available: <https://sitelabs.es/web-scraping-introduccion-y-herramientas/>. [Último acceso: 19 09 2020].

- [30] «Hipertextual,» 15 05 2014. [En línea]. Available: <https://hipertextual.com/archivo/2014/05/que-es-api/>. [Último acceso: 19 09 2020].
- [31] Springer, «Springer,» 2020. [En línea]. Available: <https://dev.springernature.com/restfuloperations>. [Último acceso: 29 09 2020].
- [32] Springer, «Springer,» 2020. [En línea]. Available: <https://dev.springernature.com/querystring-parameters>. [Último acceso: 29 09 2020].
- [33] Springer, 2020. [En línea]. Available: <https://dev.springernature.com/adding-constraints>. [Último acceso: 29 09 2020].
- [34] Elsevier B.V., «Elsevier Developers,» 2020. [En línea]. Available: <https://dev.elsevier.com/documentation/ScopusSearchAPI.wadl>. [Último acceso: 29 09 2020].
- [35] Elsevier B.V., «Elsevier Developers,» 2020. [En línea]. Available: https://dev.elsevier.com/sc_search_views.html. [Último acceso: 29 09 2020].
- [36] Elsevier B.V., «Elsevier Developers,» 2020. [En línea]. Available: <https://dev.elsevier.com/documentation/AuthorSearchAPI.wadl>. [Último acceso: 29 09 2020].
- [37] Elsevier B.V., «Elsevier Developers,» 2020. [En línea]. Available: https://dev.elsevier.com/sc_author_search_views.html. [Último acceso: 29 09 2020].
- [38] usability.gov, «usability.gov,» 16 10 2020. [En línea]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Último acceso: 16 10 2020].
- [39] F. Devin, «UXpañol,» 25 02 2017. [En línea]. Available: <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/#:~:text=Un%20Sistema%20de%20Escalas%20de,un%20objeto%20C%20dispositivo%20o%20aplicación..> [Último acceso: 05 02 2021].
- [40] J. Nielsen y R. Budiu, «Google Books,» 28 02 2013. [En línea]. Available: https://books.google.com.ec/books/about/Usabilidad_en_dispositivos_móviles.html?id=eVXLmQEACAAJ&redir_esc=y. [Último acceso: 23 11 2020].

ANEXOS

Anexo 1.- Código HTML Cuartiles

```
<div class="cellslide"> == $0
  <p>...</p>
  <table>
    <thead>
      <tr>
        <th>Category</th>
        <th>Year</th>
        <th>Quartile</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Arts and Humanities (miscellaneous)</td>
        <td>2019</td>
        <td>Q3</td>
      </tr>
    </tbody>
  </table>
</div>
```

Figura 6.1: Código HTML Cuartiles
Fuente: Elaboración propia

Anexo 2.- Código HTML SJR

```
<div class="cellslide"> == $0
  <p>...</p>
  <table>
    <thead>
      <tr>
        <th>Year</th>
        <th>SJR</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>2019</td>
        <td>0.121</td>
      </tr>
    </tbody>
  </table>
</div>
```

Figura 6.2: Código HTML SJR
Fuente: Elaboración propia

Anexo 3.- Código HTML Citaciones por documento


```

<div class="cellslide">
  <p>...</p>
  <table>
    <thead>
      <tr>
        <th>Cites per document</th>
        <th>Year</th>
        <th>Value</th>
      </tr>
    </thead>
    <tbody> == $0
      <tr>
        <td>Cites / Doc. (4 years)</td>
        <td>2017</td>
        <td>0.000</td>
      </tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
    </tbody>
  </table>
</div>

```

Figura 6.3: Código HTML Citaciones por documento
Fuente: Elaboración propia

Anexo 4.- Código HTML Total citas

```

<div class="cellslide">
  <p>...</p>
  <table>
    <thead>
      <tr>
        <th>Cites</th>
        <th>Year</th>
        <th>Value</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Self Cites</td>
        <td>2017</td>
        <td>0</td>
      </tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
    </tbody>
  </table>
</div>

```

Figura 6.4: Código HTML Total citas
Fuente: Elaboración propia

Anexo 5.- Código HTML Citas externas por documento

```
<div class="cellslide">
  <p>...</p>
  <table>
    <thead>
      <tr>
        <th>Cites</th>
        <th>Year</th>
        <th>Value</th>
      </tr>
    </thead>
    <tbody>
      <tr == $0
        <td>External Cites per document</td>
        <td>2017</td>
        <td>0</td>
      </tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
    </tbody>
  </table>
</div>
```

Figura 6.5: Código HTML Citas externas por documento
Fuente: Elaboración propia

Anexo 6.- Código HTML Documentos citables

```
<div class="cellslide">
  <p>...</p>
  <table>
    <thead>
      <tr>
        <th>Documents</th>
        <th>Year</th>
        <th>Value</th>
      </tr>
    </thead>
    <tbody == $0
      <tr>
        <td>Non-citable documents</td>
        <td>2017</td>
        <td>0</td>
      </tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
      <tr>...</tr>
    </tbody>
  </table>
```

Figura 6.6: Código HTML Documentos citables
Fuente: Elaboración propia

Anexo 7.- Código HTML Documentos citados

```
▼<div class="cellslide"> == $0
  ▶<p>...</p>
  ▼<table>
    ▼<thead>
      ▼<tr>
        <th>Documents</th>
        <th>Year</th>
        <th>Value</th>
      </tr>
    </thead>
    ▼<tbody>
      ▼<tr>
        <td>Uncited documents</td>
        <td>2017</td>
        <td>0</td>
      </tr>
      ▶<tr>...</tr>
      ▶<tr>...</tr>
      ▶<tr>...</tr>
      ▶<tr>...</tr>
    </tbody>
  </table>
</div>
```

Figura 6.7: Código HTML Documentos citados
Fuente: Elaboración propia